



ENTERPRISE ARCHITECT

用户指南系列

XML Schema (XSD)

Author: Sparx Systems

Date: 13/11/2024

Version: 17.0

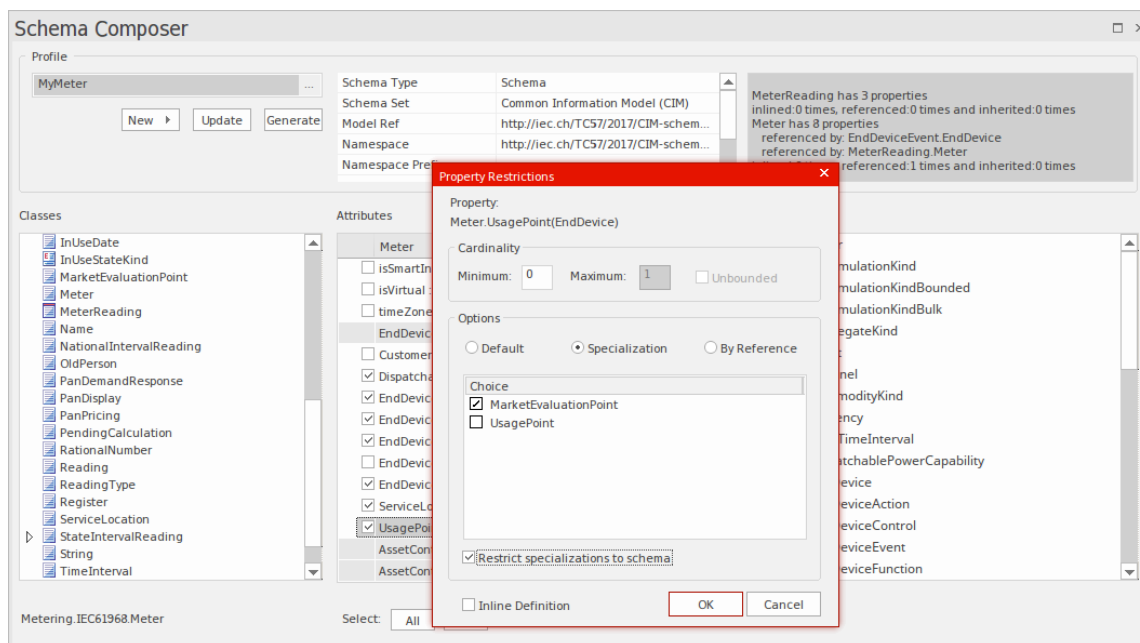
创建于  **ENTERPRISE
ARCHITECT**

目录

XML Schema (XSD)	4
Schema 编辑器	5
Schema 编辑器Profiles	7
创建架构配置文件	9
架构组成	11
类图表	16
架构分析	18
生成架构	19
选择架构配置文件	20
生成架构文件	22
CIM架构指南	24
NIEM架构指南	26
UPCC架构指南	28
模型组成	29
生成模型子集 (变换)	31
核心组件 (UPCC) 的UML配置文件	33
可用的框架	36
安装核心框架	39
架构Importer	42
Schema 编辑器自动化集成	44
Schema 编辑器Addin集成	45
Schema 编辑器脚本集成	46
MDG 技术- UML配置文件Extensions	51
XSD 模型	53
建模XSD	54
XSD图表	56
架构包	57
全局元素	59
当地元素	61
全局属性	63
本地属性	65
属性组	67
复杂类型	69
简单类型	71
团体	73
任何	75
任何属性	77
联盟	79
集团模型	81
枚举	83
来自抽象类模型的 XML	85
默认UML到 XSD 映射	87
生成XSD	88
生成全局元素	90
导入XSD	91
全局元素和复杂类型	93
XSL 转换	94

XSL变换模型	96
执行 XSL变换	98
调试一个 XSL变换	99
XML 验证	100
面向 XML 服务架构	103
WSDL	104
WSDL 1.1 模型结构	105
WSDL模型	107
WSDL命名空间	110
WSDL信息	111
WSDL信息部件	113
WSDL端口类型	115
WSDL端口类型操作	117
WSDL捆绑	119
WSDL捆绑操作	121
WSDL 服务	123
WSDL 文档	125
生成WSDL	127
导入WSDL	129
SoaML	130
SoaML工具箱Pages	132
SOMF 2.1	135
全国信息交换NIEM建模2.1	136
全国信息交换NIEM建模	143
NIEM的UML配置文件	144
下载NIEM参考模型	152
创建NIEM IEPD	153
自定义您的 IEPD模型	156
NIEM IEPD 生成	160
创建NIEM数据模型	161
使用Schema 编辑器对NIEM进行子集化	163
浏览示例	166
示例NIEM架构	170
导入NIEM XML Schema	178

XML Schema (XSD)



Enterprise Architect中的结构模型，尤其是类模型，经常用于定义某些感兴趣领域的元模型。例如，可以使用类模型来定义元模型，以严格定义构成地理空间信息领域的对象、数据、关系和类型。同样，可以（并且正在）构建模型来描述诸如水管理、健康、零售、保险、汽车注册、娱乐等领域。

这些模型非常有价值，并且经常代表商业或基于标准的组织在时间和金钱上的重大投资。实现这些模型的好处的一个重要部分，特别是在必须在多方之间交换信息的情况下，是在模式的定义中（通常基于 XSD），它规定了如何形成消息以符合底层元模型。传统上，这种消息模式是基于元模型手工编写的。这通常是一项费力且容易出错的练习。

Enterprise Architect长期以来一直与基于商业和标准的元模型的开发相关联，并且在Enterprise Architect模型文件中定义了许多模型示例，用于指定感兴趣的信息域的确切构造。

Enterprise Architect中的Schema 编辑器旨在最大限度地利用存储在Enterprise Architect模型文件或存储库（或基于云的服务器）中的模型，将模型信息转换为符合命名标准和格式的模式。各种流行的行业元模型。这种方法大大减少了形成有效模式所需的时间，并消除了将模型信息转录为模式文本的人为错误。

当前版本的Schema 编辑器支持多种技术的 XSD 生成，此外还支持通过与自动化接口和插件

紧密集成来自定义输出插件

框架。以这种方式，可以使用“原样”提供的模式生成器之一或使用JavaScript编写自定义生成器，或者通过编写合适的插件

来进一步完全自定义过程插件

在选择的语言。

除了新的Schema 编辑器，Enterprise Architect还支持使用支持相关类型显式建模的UML Profiles对 XSD 和 WSDL 定义进行建模。当从头开始构建复杂的 XSD 或 WSDL 并且需要有一个完整的最终模式的可视化模型时，这有时是必要的。注记由于Enterprise Architect还支持 XSD 文档的导入，因此可以使用生成模式Schema 编辑器，然后出于文档和可视化目的（甚至进一步自定义），将该模式导入回当前或不同的模型中。

架构工程部分中包含的其他主题专门用于元物件功能（MOF）、本体定义元模型（ODM）和国家信息交换模型（NIEM）。关于NIEM的部分非常广泛，因为Enterprise Architect包括许多对模型和使用NIEM域和模式所必需的特征。与其他一些技术一样，还有一个可下载的NIEM核心版本作为Enterprise Architect模型。

Schema 编辑器

无缝模型架构兼容信息在一个简单而高效的工具中定义

Schema 编辑器是一个多功能工具，用于从模型中快速轻松地定义各种正式模式。由于Schema 编辑器的独特性，在构建 XSD (或其他) 文档的定义时，没有必要使用配置文件或原型元素。这极大地增强了底层模型的可重用性，并有助于减轻直接处理 XSD 或其他元素类型和限制时出现的复杂性。

在过去十年中，许多行业一直在努力定义特定于其行业的共享元模型，而这些模型现在构成了跨组织和跨地理边界共享合同信息的基础。Schema 编辑器的典型使用场景是创建消息定义 (schema) 以在组织之间交换信息，确保此类消息符合相关各方已采用的底层元模型。

当组织之间共享信息时，通常只需要完整元模型的一个子集，但共享的内容必须精确地符合商定的元模型。在这种情况下，Schema 编辑器是基于子集和受限数据集推导契约模式的完美工具，这些数据集在整个元模型中进行“切片”。

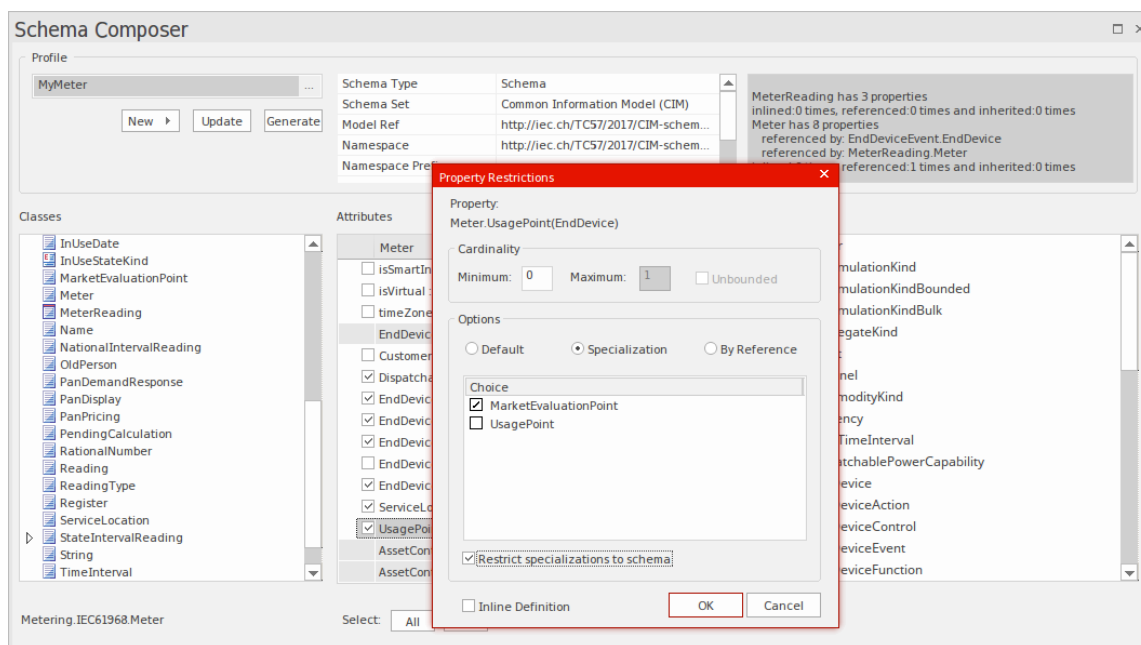
Schema 编辑器避免了直接使用 XSD 和其他模式语言的常见“痛点”：

- 除了您的“正常”业务和数据模型之外，无需创建由特定 XSD 元素组成的相对复杂的 XSD 模型来定义所需的数据、其关联和引用以及任何限制或条件
- 您无需了解如何使用 XSD 元素并应用 XSD 命名规则和约定来正确构建此类模型；支持的标准指定的格式和命名规则会自动得到处理

Schema 编辑器以可重用和可访问的方式极大地简化了创建符合标准的模式的过程。在此图中，您可以看到如何使用简单的类图作为Schema 编辑器的源来生成XML Schema。

Schema 编辑器在Enterprise Architect的企业版、统一版和终极版中

Schema 编辑器



此图显示了示例模型中进程顺序域的架构组合。

访问

--	--

功能区	开发 > 架构建模 > Schema 编辑器 > 打开Schema 编辑器
-----	---------------------------------------

好处

Schema 编辑器：

- 在类模型而不是 XML 模式配置文件上运行
- 让您摆脱特定于 XSD 的设计和模式生成决策，同时仍确保整个配置文件的一致性
- 可以对通用类模型进行操作以提供通用的 XSD 文档
- 在具有特定领域意义的行业标准类模型上操作时最有用
- 在大多数情况下，在一个完整模型上运行，从中抽取选定类的属性子集以构建特定消息，仅传达信息发送或请求所必需的内容
- 对于NIEM等标准，将生成一个新的子模型，作为更广泛的NIEM兼容模式定义的一部分

Schema 编辑器目前支持的标准包括：

- 公共信息模型(CIM)
- 全国信息交换NIEM建模
- 联合国贸易便利化和电子业务中心 (UN/CEFACT)建模方法 (UMM)，特别是命名和设计规则 (NDR) 2. 1和 3.0
- 通用业务语言 (UBL) ，特别是命名和设计规则 (NDR) 3.0

Schema 编辑器还可以帮助您使用不同格式构建同一消息的定义，例如：

- XSD
- RDFS
- JSON

另外Schema 编辑器：

- 支持使用自定义插件实现的格式插件
它利用了Schema 编辑器自动化接口
- 内置支持不同行业模型使用的各种序列化格式和样式

Schema 编辑器Profiles

Schema 编辑器配置文件是描述构成特定模式或子模型的元素和限制的配置文件。Profiles通常与特定技术相关联，例如公共信息模型(CIM) 或核心组件的UML 配置文件(UPCC)，以及对配置文件中材料的解释以及发布的模式或子模型的性质将取决于所使用的特定技术生成器。虽然Enterprise Architect支持许多“开箱即用”的技术（并且正在计划更多），但也可以通过利用Enterprise Architect中广泛的自动化接口来利用Schema 编辑器Profiles的丰富内容来自定义流程根据您的条件，在插件

中插件

或脚本。

架构Profiles

Schema 编辑器配置文件A两种形式。每种形式都满足特定的系统要求——架构生成（xsd、rdfs、json）和子模型创建。当您在Schema 编辑器中创建配置文件时，您可以根据需要选择要使用的表单。Schema 编辑器中A单个配置文件可用于以常用形式组成模式，或从核心模型创建UML子模型。

配置文件类型

类型	描述
模型变换	这种类型A配置文件用于从核心模型。
架构	这种类型A配置文件用于生成模式；通常是表示消息的 XSD 模式，还有其他格式，例如 JSON object表示法和资源描述符格式。

架构组合方法论

国家信息交换模型 (NIEM)

Enterprise Architect提供了一个NIEM框架和Schema 编辑器，用于生成子模型和 XML 模式。

公共信息模型(CIM)

Enterprise Architect Schema 编辑器支持提供开箱即用的 CIM 标准，用于组合符合 CIM 的模式。

通用业务语言 (UBL)

Enterprise Architect提供了一个通用业务语言框架，以及为模式生成提供 UBL 标准的Schema 编辑器。

核心部件技术规范 (CCTS) UN/CEFACT

Enterprise Architect为 Core Components 框架和Schema 编辑器提供了UML配置文件。 Composer 可以从核心组件库生成业务组件库，并简化从消息组件/业务信息实体的模式的组合/发布。

通用的

在标准不满足您的要求的情况下，通用选项为您的UML模型中的快速模式组合提供了更简单的选择。通常，您将使用具有属性、关联、聚合和继承的UML类来模型您自己的数据库。然后，您可以将此模型用作Schema 编辑器的输入。

EA脚本引擎

Enterprise Architect提供了一个脚本引擎，支持JavaScript、VBScript 和 JScript 语言。脚本引擎也与Schema 编辑器集成。在为特定标准或通用方案生成模式时，可以使用脚本自行执行操作或作为标准提供的选项的补充。

EA插件

Enterprise Architect提供插件

与Schema 编辑器集成。一个插件

可以通过向Enterprise Architect注册其兴趣来参与子模型或模式的生成。插件

可以提供要在“架构生成”对话框中列出的选项和备选方案，并且在选择其选项时将被调用。插件

可以使用Schema 编辑器自动化接口访问配置文件的内容。

创建架构配置文件

模式配置文件A模式的名称、技术和内容标识为定义如何生成模式的前兆。您可以根据需要创建和编辑任意数量的架构配置文件。架构配置文件绑定到单一技术，并将映射到生成的架构或子设置转换。

访问

功能区	开发 > 架构建模 > Schema 编辑器 > 打开Schema 编辑器
-----	---------------------------------------

新建配置文件

如果您要为特定技术创建架构，请先打开已加载所需元模型的模型。使用模型生成器和/或 Sparx RAS/云服务时，Sparx Systems会提供许多元模型。请按照以下步骤构建新配置文件。显示Schema 编辑器后，单击“新建”按钮并选择配置文件类型，架构或转换。

新配置文件屏幕

选项	行动
架构集	选择要使用的标准，或选择“通用”选项。
命名空间	根据您选择的标准，此字段可能采用自动值或保持空白。如果为空，请提供相关的命名空间。请参阅下一节，了解如何在Schema 编辑器中管理命名空间。
将个人资料保存到：	Profiles可以存储在文件系统或模型中。存储在模型中的Profiles可以与他人共享，而文件系统配置文件是私有的。

确定	单击此按钮可在 Composer 中编辑新模式。
----	--------------------------

命名空间

创建新配置文件时，您指定目标命名空间和命名空间前缀。模式通常涉及多个名称空间，并且Schema 编辑器在单个模型中提供了对此的支持。标识名称空间的方案是在一个包上存在两个特定属性。属性是指定命名空间的“URI”和提供命名空间前缀的“别名”。该属性可以出现在直接包或父包上。当存在时，类的元素将采用该命名空间。在不存在命名空间的情况下，类将采用创建配置文件时指定的目标命名空间。

保存配置文件

单击更新按钮以保存您刚刚创建的配置文件。

注记

- 为NIEM创建和生成模式的过程在NIEM帮助主题中有额外的注记
- Schema 编辑器在Enterprise Architect的企业版、统一版和终极版中

架构组成

模式组合是指从模型中提取的一组受限制的元素，它们共同描述了模型中没有等效的唯一实体。通常，模式组合用于生成模式文件，例如 XSD 文件。相反，模型组合用于将材料配置为子集“转换”的基础 - 例如在创建 NIEM模型子集时。

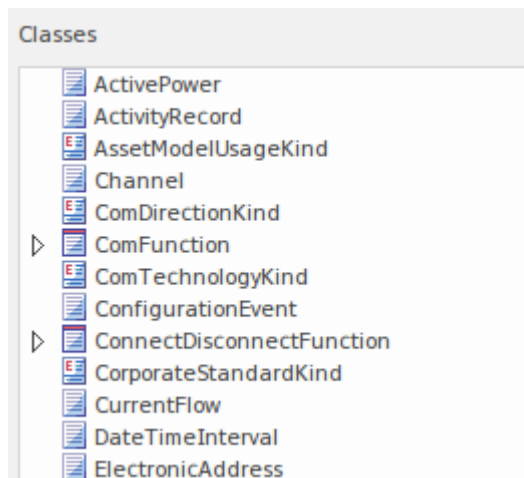
定义架构内容

这些步骤将引导您完成在架构配置文件中编写类型的基本过程，并展示如何限制元素的内容以满足消息要求。

添加类

将所需的类元素从浏览器窗口拖到“类”面板中。当你添加一个类时：

- 它的祖先列在中间面板的“继承”部分
- 其属性列在“继承”部分下，每个属性对应一个空白复选框；关联和聚合条目根据连接器上的角色名称命名
- 它的模型结构路径显示在“类”面板下方



选择属性

Attributes	
	ComFunction
	Inheritance
<input type="checkbox"/>	EndDeviceFunction
<input type="checkbox"/>	AssetFunction
<input type="checkbox"/>	IdentifiedObject
	ComFunction.Attributes
<input checked="" type="checkbox"/>	amrAddress : String
<input checked="" type="checkbox"/>	amrRouter : String
<input checked="" type="checkbox"/>	direction : ComDirectionKind
<input checked="" type="checkbox"/>	technology : ComTechnologyKind
	ComFunction.Associations
<input type="checkbox"/>	ComModule : ComModule
	EndDeviceFunction.Attributes
<input checked="" type="checkbox"/>	enabled : Boolean
	EndDeviceFunction.Associations
<input type="checkbox"/>	EndDevice : EndDevice
<input type="checkbox"/>	Registers : Register

每当您在“类”列表中选择一个类时，其属性和模型祖先都会列在“属性”列表中。选中每个属性对应的复选框以定义此类型的元素。选择后，属性的类型会自动添加到架构中，出现在“类”列表和右侧的“架构”面板中。

取消选中某个属性时，不会自动删除该类型。可以使用类上下文菜单删除类型。值得注意的是，每次选择一个类时，所有对该类的引用都会显示在状态面板中，让您可以快速审阅任何类的使用情况。

```
referenced by: ConfigurationEvent.Names
inlined by: ConnectDisconnectFunction.Names
referenced by: Manufacturer.Names
referenced by: Meter.Names
referenced by: MeterMultiplier.Names
referenced by: ReadingType.Names
referenced by: Register.Names
```

继承

如果您喜欢或预见到在您准备的模式中需要继承，那么首先从祖先开始组合是有意义的，然后在添加子类时重新使用它们。该方法不是一成不变的。您可以随时从继承模型切换到聚合组合，反之亦然。这里简单介绍一下 Schema 编辑器中提供的继承。

Schema 编辑器在处理继承方面提供了灵活性。例如，您可以选择从类及其父级中聚合选定的属性，同时选择继承祖父级。但是，当您选择使用继承时，您也选择继承该类型的受限形式。在此列表中选择祖先时，生成的 XML 模式将显示标识此祖先的扩展元素。只能选择一个祖先。

单击更新按钮以验证并保存您的架构配置文件。

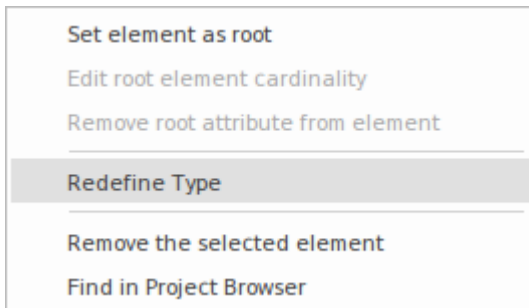
如果配置文件有任何问题，它们会在屏幕右上角的状态面板中进行标识。

重新定义的类型

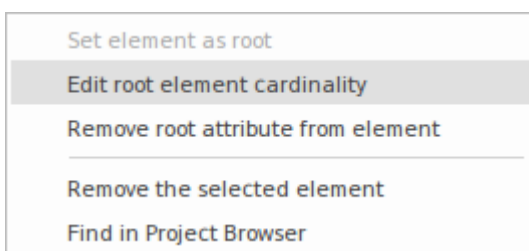
模式组合的常见问题之一是需要能够改变类型描述以满足模式描述的实例的各种需求。例如，车辆可以按其品牌、模型和价格通过卡车类型的元素来描述，但通过年份、模型和颜色可以通过轿车类型的元素来描述。问题是我们可能只有一个实际的 *Vehicle* 类可供我们使用。为了解决这个问题，Schema 编辑器允许您克隆 *Vehicle* 类并为其命名。然后，您可以将此版本的 *Vehicle* 分配属性 *Vehicle* 作为其类型的任何属性。创建的类型仅在模式域内可用 - 模型未受影响。

要创建类型的新定义，首先在“类”列表中选择类，然后右键单击它并选择“重新定义类型”选项。输入此类型的

唯一名称，然后按 Enter 键。然后，您可以像对任何类一样独立定义或限制此类型。



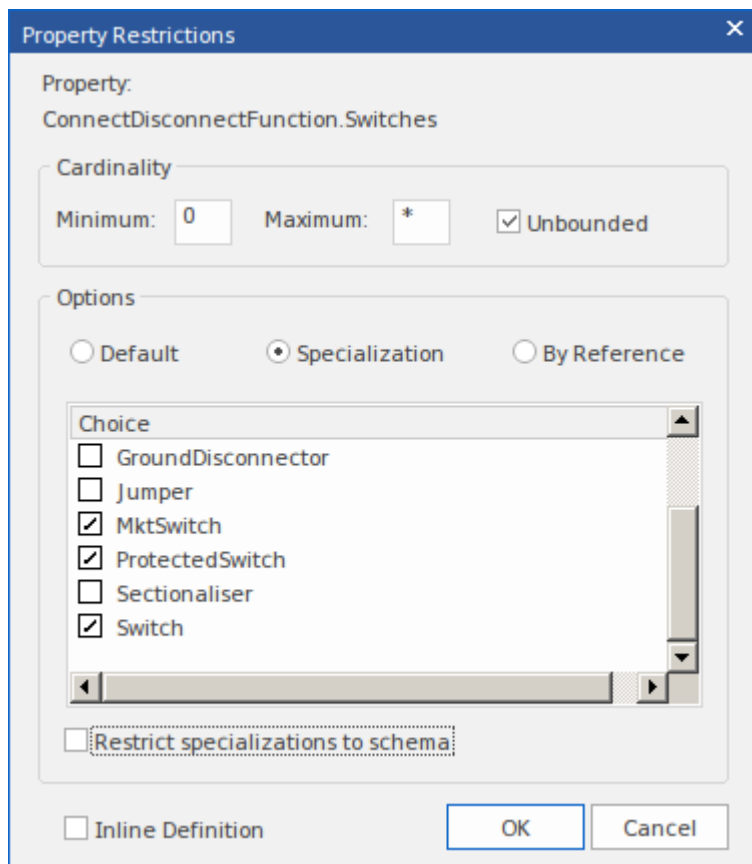
根元素



生成模式时，会生成表示消息的单个顶级元素。这个顶层元素的主体或元素是标记为根元素的类。这些根元素的基数可以调整。要将类标记为根元素或限制其基数，请右键单击列表中的类并使用以下上下文菜单选项：

- 将元素设置为根-根元素构成表示消息/配置文件的顶级元素的主体
- 编辑元素根- 设置最小和最大实例数
- 从元素中删除根属性-从类中删除根标记
- 删除选定的元素- 从架构中删除选定的元素
- 在项目中查找浏览器- 在浏览器窗口中定位并突出显示元素

属性限制



在“属性”列表中，右键单击选定的属性并使用上下文菜单添加、编辑或属性限制。使用特征用于：

- 修改属性基数
- 重新定义属性的类型
- 启用和限制可用于此属性的选项
- 标记要作为内联元素定义发出的属性
- 将要发出的属性标记为“通过参考”

基数

属性的基数可以从其对应的模型进一步限制，但不能减少限制。任何根元素类和任何类属性都可以更改基数。

类型重新定义

当类在Schema编辑器中重新定义时，它会创建一个新类型。新类型是原始类型的克隆，但具有模式唯一的名称。A，Payment枚举类型可能被重新定义为CardPayment以更好地适应模式目的。新类型是对原始类型的限制，因为不能向其添加新属性。其它属性共享此类型并通过在其限制对话框中指定新类型来类似地受到限制。在其他属性的限制下，可以提供重新定义的类型（例如子类型）作为附加的选择元素。

专长

如果存在属性类型的特化，则这些子类型将在“限制”对话框中可用。When more than one specialization is selected, these will appear as choice elements in the schema.当仅选择一个时，该属性将在模式中显示此子类型。

内联元素

选中A框时，属性类型将作为内联定义发出。

通过参考

属性在模式中A时将采用 按引用"形式。'By reference' 表单发出一个内联 complexType , 它定义了一个名为 'ref' 的类型为 'string' 的单个属性。

属性约束- Facets

Schema 编辑器通用配置文件支持构面。分面的来源是属性上的标记值。如果从XML Schema规范中命名了一个约束分面，则该标记值被识别为分面；JSON 验证关键字也被识别。



Tagged Values	
Attribute (name)	
minLength	3
whitespace	preserve
maxLength	64

```

<!--
-->
<xs:complexType name="Manufacturer">
  <xs:sequence>
    <xs:element name="name" minOccurs="1" maxOccurs="1">
      <xs:simpleType>
        <xs:restriction base="xs:string">
          <xs:maxLength value="64"/>
          <xs:minLength value="3"/>
          <xs:whiteSpace value="preserve"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:element>
  </xs:sequence>
</xs:complexType>

```

从XML Schema约束方面：

- 长度
- 最小长度
- 最长长度
- 图案
- 枚举
- 空白
- 最大包容性
- 最大独家
- 分钟独家
- 最少包含
- 总数
- 分数位数

JSON中的验证关键字：

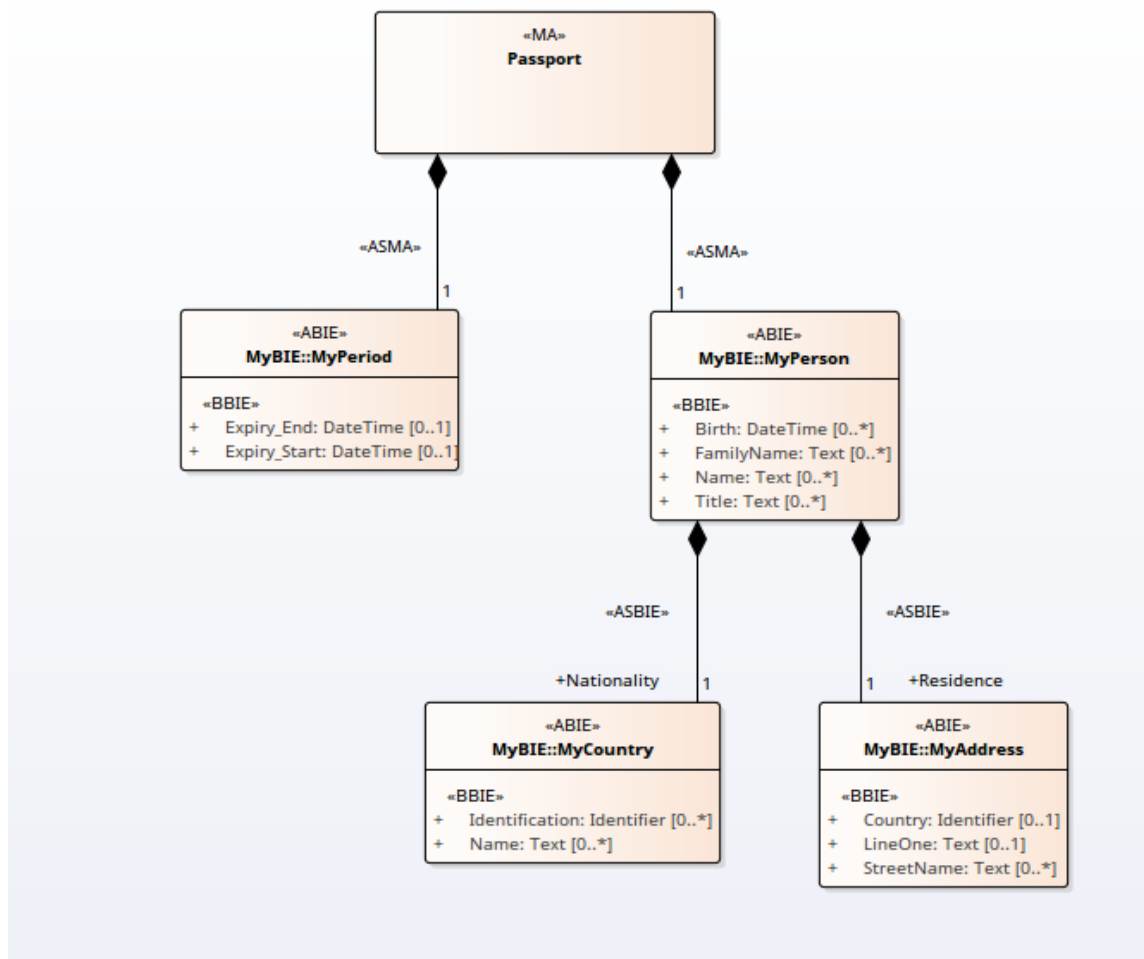
- 数字和整数
 - 倍数
 - 最低限度
 - 最大
 - 独家最低
 - 独家最大
- 字符串
 - 最小长度
 - 最长长度
 - 图案
- 数组
 - minItems
 - 最大物品
 - 独特的物品

类图表

Schema 编辑器还支持从通用UML类创建简单的 XSD 和其他格式。例如，当需要以通用方式导出类定义以供脚本或基于 Web 的工具使用时，这尤其有用。

从类图生成模式

喜欢在合成中使用建模方法的用户也可以使用Schema 编辑器来生成他们选择的格式。任何类图都可以加载到 Schema 编辑器中。此图像说明了使用核心组件的UML配置文件组成的消息，但没有必要根据特定的UML配置文件对消息进行建模。



将消息加载到 Composer

通过在表示消息的图表上选择一个类并使用其上下文菜单在Schema 编辑器中将图表呈现为模式，将消息加载到 Composer 中。选择的类将成为消息的根元素，其关系将塑造加载的模式。

这是加载到Schema 编辑器中的类图

The screenshot displays a software interface for managing XML Schemas (XSD). The interface is divided into several sections:

- Profile:** Shows the selected profile as "Passport". It includes buttons for "New", "Update", and "Generate".
- Schema Information:** A table-like view showing:
 - Schema Type: Schema
 - Schema Set: Core Components (UN/CEFACT) - NDR 3.0
 - Model Ref: My Model
 - Namespace: http://myauthority.org/passports
 - Namespace Pr...: rsm:
 - Unified Schema: true
- Classes:** A list of classes including Address, Country, Decimal, MyCode, MyDateTime, MyIdentifier, MyMeasure, MyText, Period, Person, and String.
- Attributes:** A list of attributes for the selected class (Address), including BuildingNumber (MyText), CityName (MyText), CountryName (MyText), Postcode (MyCode), and StreetName (MyText). Each attribute has a checked checkbox.
- Schema:** A tree view showing the "Passport" schema and its components: Address, Country, MyCode, MyDateTime, MyIdentifier, MyMeasure, MyText, Period, and Person.
- Summary:** A text box on the right stating: "Address has 5 properties referenced by: Person.Residence inlined:0 times, referenced:1 times and inherited:0 times".

架构分析

随时随地分析

Schema 编辑器在将每种类型添加到模式时以及在选择类时执行分析。系统输出窗口将显示该类型存在多少（如果有）引用、它被继承的次数和其他有用的信息。此图显示了一条消息，详细说明了引用选定类的元素。

```
referenced by: Period.DateOfIssue_Start
referenced by: Person.Birth
inlined:0 times, referenced:3 times and inherited:0 times
MyCode has 10 properties
referenced by: Address.Postcode
referenced by: Person.Gender
inlined:0 times, referenced:2 times and inherited:0 times
```

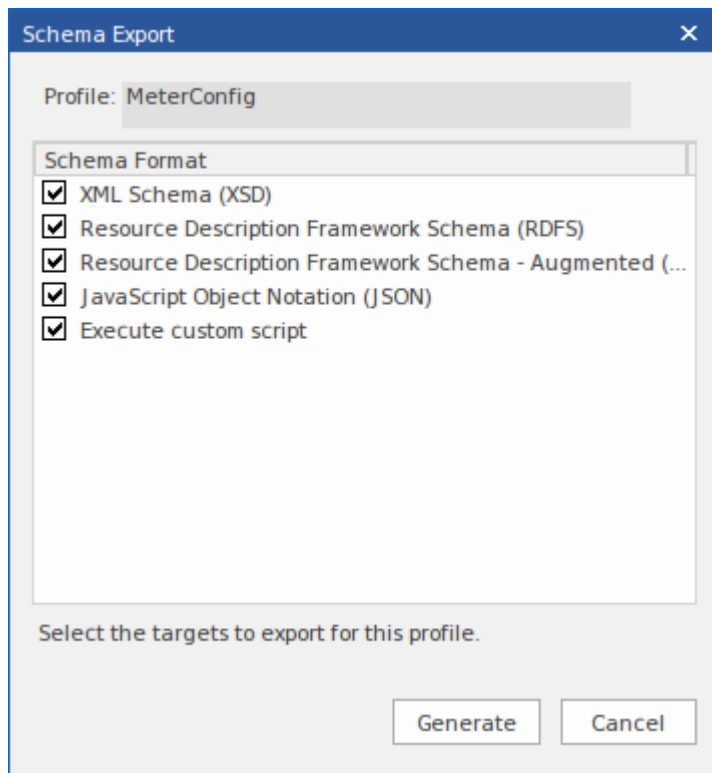
随时随地验证

Schema 编辑器对一项技术执行特定的验证。此图像显示有关基于 UN/CEFACT 核心组件标准构建的架构中的类缺少标记值的警告。

```
Warning: dataTypeQualifierTermName facet missing for class MyIdentifier
Warning: dataTypeQualifierTermName facet missing for class MyMeasure
Warning: dataTypeQualifierTermName facet missing for class MyText
Warning: dataTypeQualifierTermName facet missing for class MyCode
Warning: dataTypeQualifierTermName facet missing for class MyIdentifier
Warning: dataTypeQualifierTermName facet missing for class MyDateTime
Warning: dataTypeQualifierTermName facet missing for class MyMeasure
```

生成架构

在任何阶段都设计了配置文件，并且具有最少的定义和自定义，您可以快速轻松地生成模式或子模型。根据选择的技术和配置文件类型（模式或转换），呈现给您的格式会有所不同。注记通常可以一次生成多种格式。而且，当然，您可以轻松地重复该过程，随着组合的发展或模型的设计更改。



访问

功能区	开发 > 架构建模 > Schema 编辑器 > 打开Schema 编辑器：生成
-----	--

注记

- Schema 编辑器在Enterprise Architect的企业版、统一版和终极版中

选择架构配置文件

访问

功能区	开发 > 架构建模 > Schema 编辑器 > 查找Schema 编辑器Profiles 开发 > 架构建模 > Schema 编辑器(icon)
-----	---

在模型中查找架构Profiles

架构配置文件可以从功能区的Schema 编辑器下拉菜单中快速定位。该菜单提供对模型中现有配置文件的快速访问。

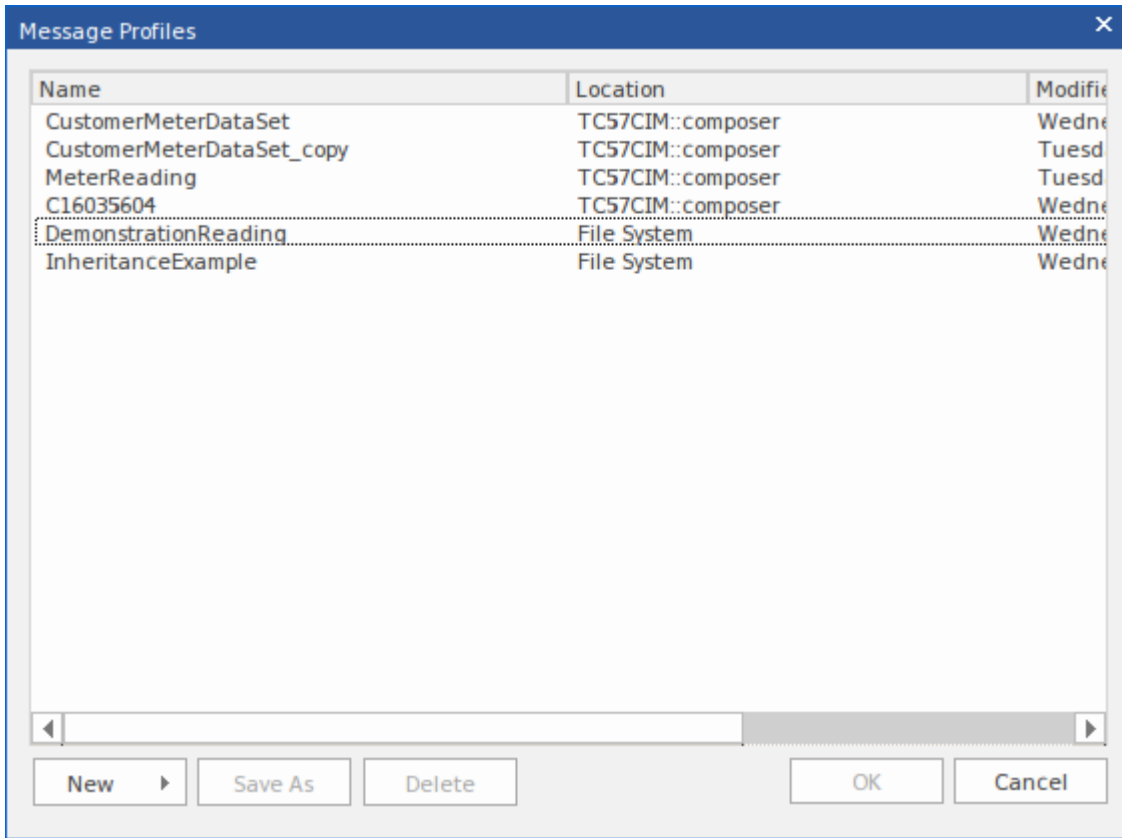
The screenshot shows the 'Find in Project' dialog box with the following content:

- Find in Project dropdown menu:
 - Open Schema Composer
 - Import for Schema Composer
 - Find Schema Composer Profiles** (selected)
 - Help...
- Search results table:

Element	Type	Aut...	Stereotype	Scope	Status	Phase	ModifiedDate
CustomerMeterDataSet	Artifact	sme...		Public	Proposed	1.0	9/03/2016 9:4...
CustomerMeterDataSet_copy	Artifact	sparx...		Public	Proposed	1.0	15/03/2016 4...
MeterReading	Artifact	sparx...		Public	Proposed	1.0	15/03/2016 5...
C16035604	Artifact	sparx...		Public	Proposed	1.0	16/03/2016 12...

在Schema 编辑器中定位架构Profiles

架构配置文件可以存储在模型和文件系统中。您可以通过打开Schema 编辑器并单击选择配置文件按钮（带有省略号“...”的按钮）在您的模型中创建的每个配置文件轻松找到该模型的所有配置文件列表并指示位置他们住;模型或文件系统。您可能有许多您使用的Enterprise Architect模型，但只会列出与开放模型相关的那些文件系统配置文件。



The screenshot shows a 'Message Profiles' dialog box with a table listing several profiles. The table has three columns: Name, Location, and Modified. The 'DemonstrationReading' profile is highlighted with a dotted border.

Name	Location	Modified
CustomerMeterDataSet	TC57CIM::composer	Wedne
CustomerMeterDataSet_copy	TC57CIM::composer	Tuesd
MeterReading	TC57CIM::composer	Tuesd
C16035604	TC57CIM::composer	Wedne
DemonstrationReading	File System	Wedne
InheritanceExample	File System	Wedne

At the bottom of the dialog box, there are several buttons: 'New', 'Save As', 'Delete', 'OK', and 'Cancel'. A scrollbar is visible below the table.

生成架构文件

定义了模式配置文件并添加了必要的元素和限制后，您可以快速轻松地生成模式。XML 模式生成适用于所有技术，但每种技术都可能支持其他格式。

访问

功能区	开发 > 架构建模 > Schema 编辑器 > 打开Schema 编辑器：生成
-----	--

架构格式

选中要导出的每种模式格式的复选框。

架构格式	细节
CIM	<ul style="list-style-type: none"> XML Schema (XSD) 资源描述框架架构(RDFS) 资源描述框架架构- 增强型 (RDFS) JavaScript物件表示法 (JSON) 执行自定义脚本
UN/CEFACT NDR 3.0	<ul style="list-style-type: none"> XML Schema (XSD) 执行自定义脚本
UN/CEFACT NDR 2.1	<ul style="list-style-type: none"> XML Schema (XSD) 执行自定义脚本
通用的	<ul style="list-style-type: none"> XML Schema (XSD) 资源描述框架架构(RDFS) JavaScript物件表示法 (JSON) 执行自定义脚本
UBL 2.1	<ul style="list-style-type: none"> XML Schema (XSD) 执行自定义脚本
执行自定义脚本	<p>虽然Schema 编辑器可以为许多公认的标准生成模式，但它也为那些想要控制模式的格式和特征的用户提供了一个脚本解决方案。当您为生成器指定脚本时，它指的是您的模型中存在的语言脚本，例如JavaScript。脚本产生的方式和内容几乎取决于您。脚本如何访问Schema 编辑器中的 Schema 记录在脚本 Schema Composer Scripting Integration 中。</p>

生成

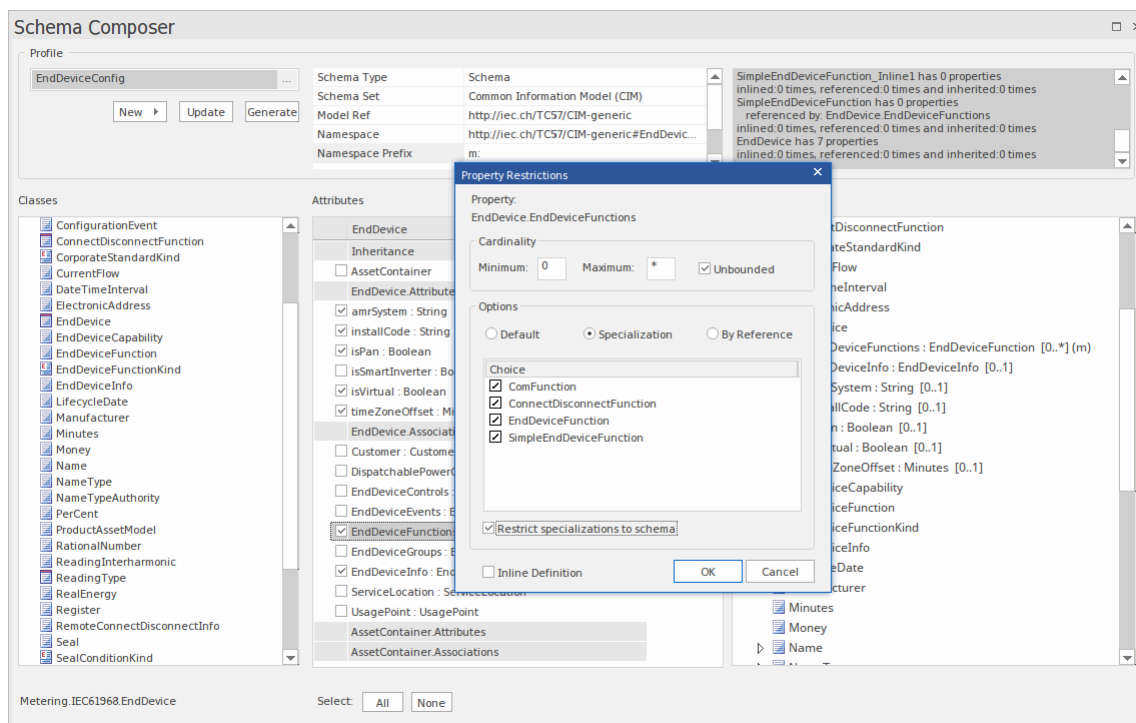
单击此按钮以生成模式。

使用文件浏览器来定位和打开模式文件。

注记

- 您可以使用Enterprise Architect编辑和验证 XML 文档，包括 XSD 架构
- 您可以将Enterprise Architect设置为 XML 文档的默认文档处理程序

CIM架构指南



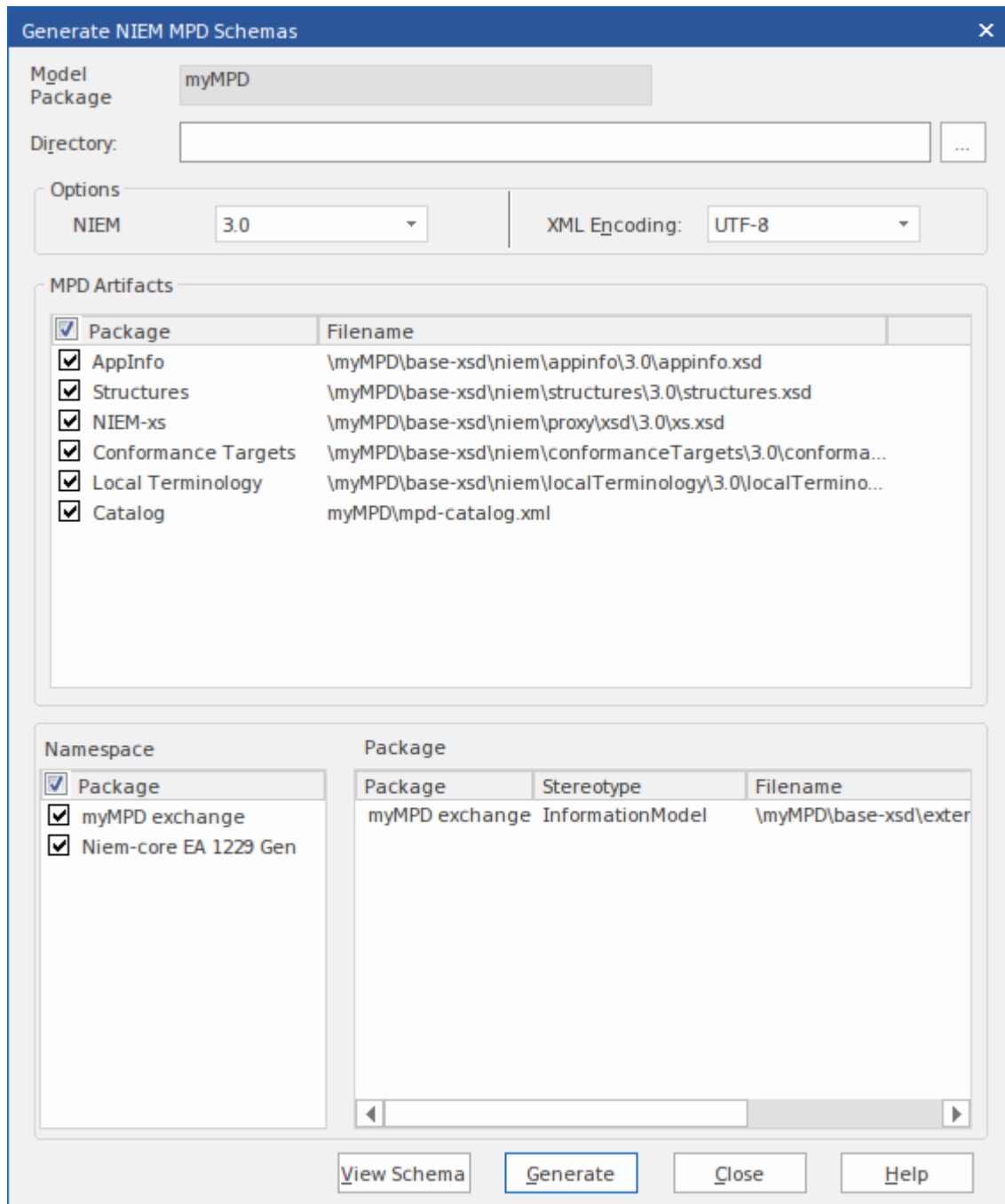
本指南描述了 CIM 兼容XML Schema的创建和生成。

创建 CIM 消息

节	行动
1	显示Schema 编辑器。
2	单击“新建 架构”。
3	输入此 CIM 模式 (消息) 的唯一名称。
4	选择公共信息模型。
5	将初始 CIM类拖到最能代表消息的类窗口中。使用上下文菜单适当地设置根元素。
6	如果您想使用继承来组合此类型，请从继承列表中选择一个祖先。
7	使用每个类的属性上的复选框来定义将描述此消息或模式的属性集。
8	使用属性上的上下文菜单对元素应用限制。
9	单击更新以保存消息。
10	单击生成按钮并选择要导出的模式格式。

NIEM架构指南

NIEM模式的生成是在 ModelPackageDescription类 (NIEM 3.0 及更高版本) 或 «ModelPackageDescription» 原型组件 (NIEM 2.1) 的实例上完成的。无论哪种情况，都会出现一个对话框，允许您配置生成的模式。



生成NIEM Schemas (NIEM 2.1)

单击具有 «ModelPackageDescription» 构造型的组件并选择以下选项之一：

功能区	特定>技术> NIEM 2.1 >生成NIEM 2.1架构或

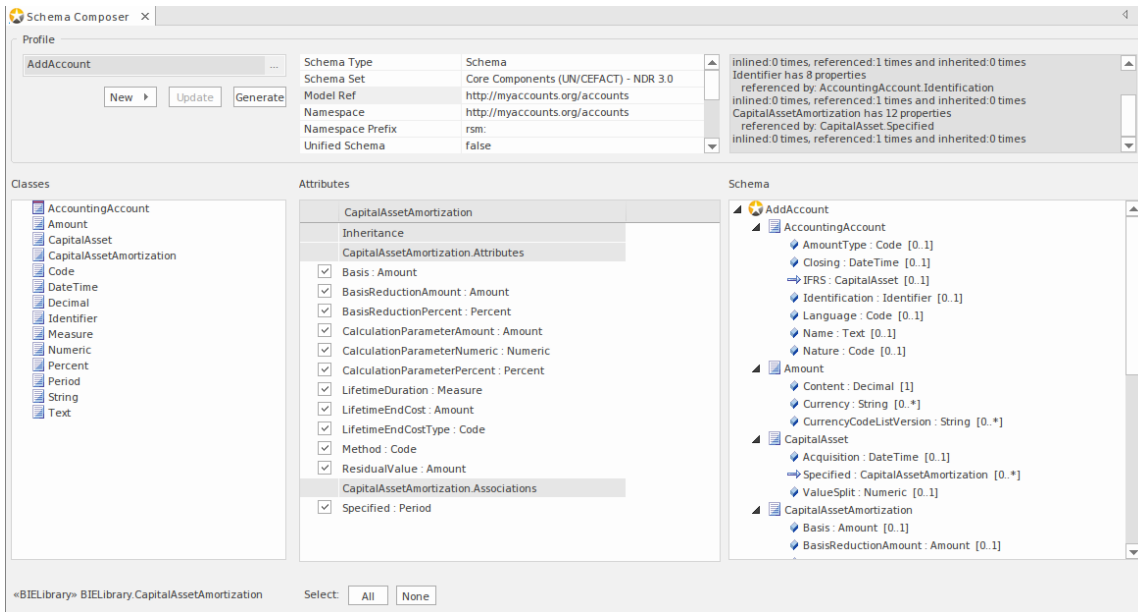
上下文菜单	右键单击元素 特定 NIEM 1 生成NIEM 1架构
-------	-------------------------------

生成NIEM Schemas (NIEM 3.0 及以上)

单击 ModelPackageDescription类的任何object实例并选择以下选项之一：

功能区	特定>技术> NIEM >生成NIEM架构
上下文菜单	右键单击元素 特定 NIEM 生成NIEM架构

UPCC架构指南



本指南描述了符合 UPCC 的 XML 模式的组成和生成。

创建UPCC架构

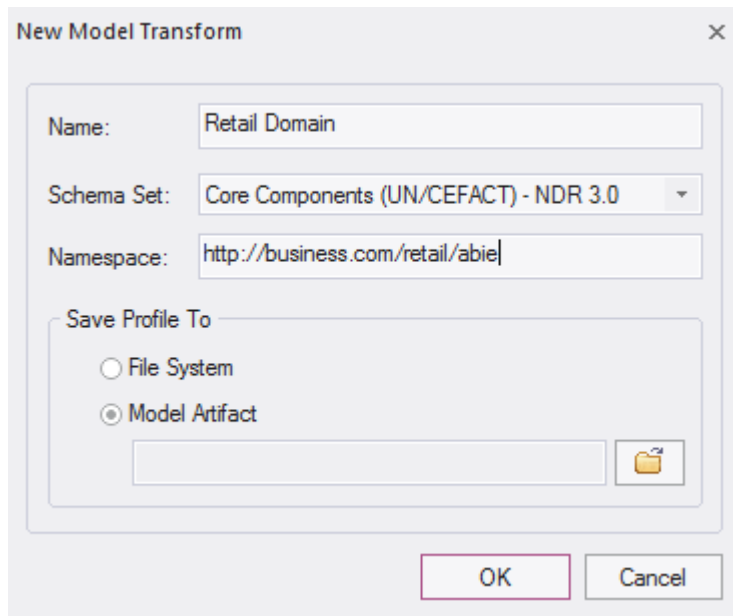
节	行动
1	显示Schema 编辑器。
2	单击 “新建 架构”。
3	输入架构的唯一名称。
4	从标准列表中选择要使用的 UPCC 命名和设计规则。
5	将一个或多个 <ABIE> 组件从 <BIE library> 拖到类列表中。
6	使用上下文菜单将类设置为根元素。
7	选择所需的属性（将引用类型添加到架构中）。
8	单击更新按钮以保存更改。
9	单击生成按钮并选择 “XML Schema ”。点击确定按钮。

模型组成

Schema 编辑器的组成特征对于从核心模型创建子模型很有用。这可以像从核心包生成单个业务包（*UN/CEFACT* 核心组件标准中的 *CDT* 库到 *BDT* 库转换）或从大型核心模型创建完成子模型一样简单。

这种任务的艰巨性可能令人生畏并且容易出错。例如，确保子模型引用的每个类型都包含在子模型中。Schema 编辑器通过自动计算依赖关系并在必要时将它们添加到模式中来解决这个问题。

创建变换

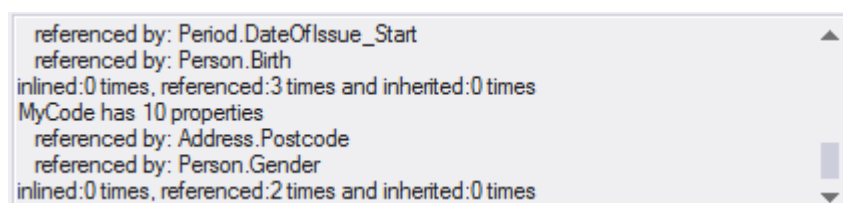


定义内容模型

通过将类从模型中拖放到Schema 编辑器类窗口并选择要包含的属性，可以将内容添加到模型中。结果类型可以反映核心类型或提供更简单的分类。当包含一个属性时，Schema 编辑器将检查属性类型，如果缺少类型，将自动将其添加到模式中。

参考检查

当属性先前包含在架构中且不再引用的属性时，不会自动删除该属性类型。但是，如果您在类窗口中选择该类型，Schema 编辑器将始终显示该类型的引用数。完全没有引用的类型可以很容易地删除。



概括

这里总结了组成子模型的过程：

1. 创建一个 Schemer Composer 转换配置文件
2. 通过将模型中的类拖放到模式中来创建元素。
3. 包括/排除所需的属性。
4. 生成子模型。

访问

功能区	开发>架构建模> Schema 编辑器>打开Schema 编辑器：新建>转换
-----	--

生成模型子集 (变换)

定义了子模型或库的内容并应用了任何限制后，您现在可以生成模型。可以执行的模型转换取决于与配置文件关联的技术。此处列出了每种技术及其支持的转换：

访问

功能区	开发 > 架构建模 > Schema 编辑器 > 打开Schema 编辑器：生成
-----	--

模型变换

选择要运行的模型。

变换选项	描述
NIEM	<p>NIEM模型子集</p> <p>此选项将生成一个NIEM模型子集，其中包含配置文件描述的模式。</p> <p>当您单击确定按钮时，系统将提示您选择目标包以创建子集。构成子集的 <<Namespace>>包随后将在此位置创建。如果任何子集包已经存在于该位置，它们的内容将被添加到。所有的subset包都会有'defaultPurpose'标记值设置为'subset'。</p> <p>执行自定义脚本</p> <p>A执行用户定义的语言脚本，例如JavaScript。该脚本可以使用Schema 编辑器自动化接口访问配置文件。</p>
通用的	<p>通用模型子集</p> <p>系统将提示您输入目标包。这将使用模式中的类型填充。如果输入限定符，这将应用于生成的类。架构中的任何限制也将被应用。目标包中存在的类型将被覆盖。将添加新属性。此过程不会删除目标中存在但配置文件中不再存在的类型或属性。</p> <p>执行自定义脚本</p> <p>A执行用户定义的语言脚本，例如JavaScript。该脚本可以使用Schema 编辑器自动化接口访问配置文件。</p>
UN/CEFACT NDR 3.0	<p>BDT 图书馆</p> <p>业务Datatype库将从配置文件中列出A核心数据类型填充。构造型将根据 CCTS 规范进行转换。这些类型可能比它们的核心对应物更受限制。存在的数据类型的属性将被覆盖。新的属性和类型将添加到库中。类型由名称和构造型匹配。</p> <p>此过程不会删除目标中存在但配置文件中不再存在的类型或属性。</p> <p>国际展览局图书馆</p> <p>业务信息实体库将从聚合A核心组件中填充。配置文件中列出的构造型将根据 CCTS 规范进行转换。存在的数据类型的属性将被覆盖。新的属性和类型将添加到库中。类型由名称和构造型匹配。</p> <p>此过程不会删除目标中存在但配置文件中不再存在的类型或属性。</p>

	<p>执行自定义脚本</p> <p>A 执行用户定义的语言脚本，例如JavaScript。该脚本可以使用Schema 编辑器自动化接口访问配置文件。</p>
UN/CEFACT NDR 2. 1	<p>UDT 库</p> <p>将选定核心数据类型的非限定副本执行到 UDT 库。</p> <p>QDT 库</p> <p>合格A业务Datatype库将从配置文件中列出的核心数据类型填充。结果类型的名称将由配置文件中的命名限定符限定。构造型将根据属性规范进行转换。存在的数据类型将被覆盖。新的属性和类型将添加到库中。类型由名称和构造型匹配。</p> <p>国际展览局图书馆</p> <p>业务信息实体库将从配置文件中列出A聚合核心组件填充。构造型将根据 CCTS 规范进行转换。现有数据类型的属性将被覆盖。新的属性和类型将添加到库中。类型由名称和构造型匹配。</p> <p>执行自定义脚本</p> <p>A 执行用户定义的语言脚本，例如JavaScript。该脚本可以使用Schema 编辑器自动化接口访问配置文件。</p>

生成

单击确定按钮以生成模式。当生成完成时，将显示消息导出配置文件 <名称> 已完成。然后您可以在浏览器窗口中展开该包以查看生成的UML模型。

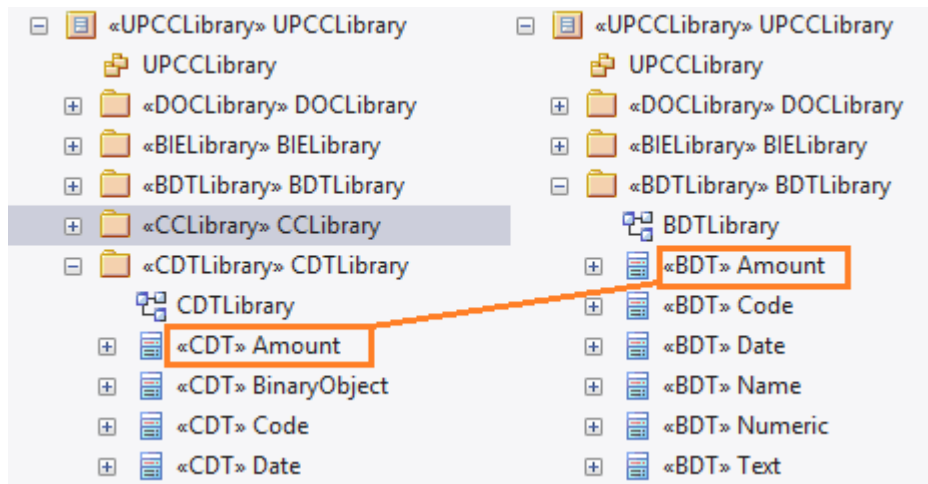
注记

- Schema 编辑器在Enterprise Architect的企业版、统一版和终极版中

核心组件 (UPCC) 的UML配置文件

UPCC 框架提供核心组件和核心数据类型库， Enterprise Architect用户可通过模型生成器使用。无论您是根据 UMM 规范模型，还是想利用此标准带来的优势，或有合规性要求，使用此技术进行模型，您至少需要一个业务数据类型库和一个业务信息实体库。Schema Schema 编辑器可以为您生成这些库。

该图显示了从 UPCC 核心 CDT 库创建的 BDT 库



公共库

核心组件的UML配置文件的两个版本共享的库。

图书馆	描述
CC图书馆	CCTS 核心组件库。
CDT图书馆	CCTS 核心数据类型库。它包含基本的数据类型，如金额、代码、文本和图形。
BIE图书馆	包含基于 CCLibrary 组件的 ABIE 实体A业务库。可以使用Schema 编辑器来组合实体。这些也可以使用该技术可用的UML建模工具进行建模。
文档库	A包通常用于信息程序集的建模。您可以通过将信息组件加载到Schema 编辑器中来生成信息组件的架构组装

UPCC 库

核心组件的UML配置文件有两个版本，NDR 3.0 和 NDR 2.1。两个配置文件都描述了一组通用的库，但有一些区别，如下所述：

NDR 3.0

图书馆	描述
-----	----

BDT图书馆	包含基于 CDTLibrary 类型的 BDT 类型A业务库。Schema 编辑器可用于从核心库中的选定类型轻松生成 BDTLibrary 的内容。
--------	--

NDR 2.1

图书馆	描述
UDT库	不合格的数据类型库。基本上是由于业务时间的上下文的镜像。Schema 编辑器可用于从核心库中的选定类型轻松生成 UDTLibrary 的内容。
QDT库	A合格的数据类型库。该库包含基于具有限定类型名称的 CDTLibrary 的受限类型。Schema 编辑器可以轻松地从中生成 QDTLibrary 的内容。


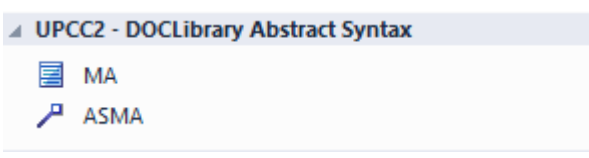
图表

核心组件的UML配置文件使用UML类图来组合组件。然而，该技术为其每个库提供了特定的工具箱页面。

UPCC工具箱Pages

公共

在此表示法中，UPCC_x 表示 UPCC 配置文件，x 是 NDR 的版本

公共	描述
UPCC _x - CCLibrary 抽象语法	
UPCC _x - DOCLibrary 抽象语法	
UPCC _x - CDTLibrary 抽象语法	

	<ul style="list-style-type: none"> ▲ UPCC2 - CDLibrary Abstract Syntax <ul style="list-style-type: none"> CDT CON SUP
UPCCx - BIELibrary 抽象语法	<ul style="list-style-type: none"> ▲ UPCC2 - BIELibrary Abstract Syntax <ul style="list-style-type: none"> ABIE BBIE ASBIE

NDR 3.0

库语法	描述
UPCC - BDLibrary 抽象语法	<ul style="list-style-type: none"> ▲ UPCC3 - BDLibrary Abstract Syntax <ul style="list-style-type: none"> BDT CON SUP

NDR 2.1

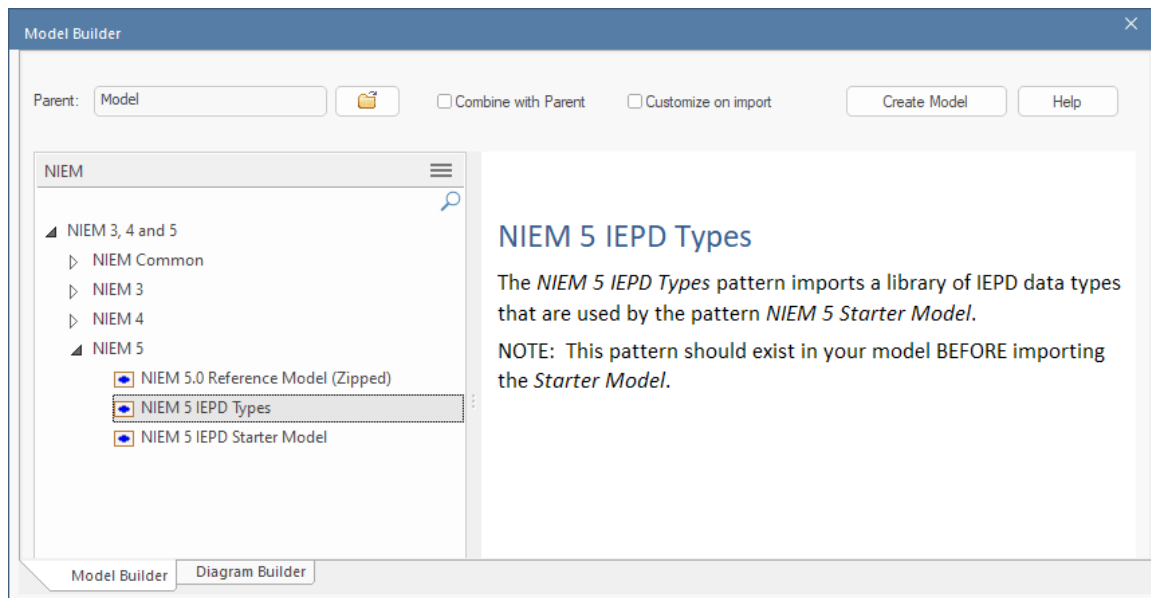
库语法	描述
UPCC - UDLibrary 抽象语法	<ul style="list-style-type: none"> ▲ UPCC2 - UDLibrary Abstract Syntax <ul style="list-style-type: none"> UDT CON SUP
UPCC - QDLibrary 抽象语法	<ul style="list-style-type: none"> ▲ UPCC2 - QDLibrary Abstract Syntax <ul style="list-style-type: none"> QDT CON SUP

可用的框架

使用Enterprise Architect模型Builder，您可以在几分钟内将Schema 编辑器支持的任何框架（例如NIEM，CIM 和 CCTS）部署到您的模型中，为使用这些技术进行建模提供多功能的UML媒介。

这些框架也可以直接从Sparx Systems可重用资产服务获得（通过云服务器“云连接”对话框，然后通过模型构建器上的“框架”）。

注记：除了 CIM 和NIEM等自定义框架之外，还可以使用标准类模型来快速构建通用模式，因此如果您不针对特定的元模型，最简单的方法可能是在UML中模型您的数据，并使用生成的模型作为Schema 编辑器的输入。



国家信息交换模型 (NIEM)

这是NIEM项目管理办公室（PMO）发布的[National Information Exchange模型](#)。

Enterprise Architect为NIEM中的建模提供了以下资源：

- 提供的框架包括NIEM核心、NIEM域、代码列表和外部模式适配器：
 - 使用 NIEM-UML 1建模的NIEM 3.0。 1
 - NIEM 1使用 NIEM-UML 1建模。 1
 - 使用 NIEM-UML 1建模的NIEM 4.0。 1
 - 使用 NIEM-UML 1建模的NIEM 5.0。 1
- NIEM子集创建：
 - Schema 编辑器可帮助您创建符合NIEM命名空间的子集
- NIEM模式生成：
 - 根据NIEM 3 中的模型包描述生成完成NIEM IEPD，NIEM 4 或NIEM 5 格式

公共信息模型(CIM)

这是国际电工委员会 (IEC) 技术委员会 57 发布的[CIM 规范](#)。

Enterprise Architect为在 CIM 中建模提供了以下资源：

- 架构组合
 - XML 模式 (XSD)
 - 资源描述符格式 (RDFS)

- 资源描述符增强格式
- JavaScript物件表示法 (JSON)
- 插件
- 一体化
- 脚本集成

通用业务语言 (UBL)

UBL 是由 [OASIS](#) 发布的 CCTS 实施，事实证明它在欧洲政府中很受欢迎，用于整合机构之间的信息交换。

Enterprise Architect为使用 UBL 组合业务文档提供了以下资源：

- UML框架
 - UBL 2.1主要文档库
 - UBL 2.1公共部件库
- 业务文档组合
 - 用于组件组合的Schema 编辑器
 - 用于文档合成的Schema 编辑器
 - 用于模式生成的Schema 编辑器
 - 插件
 - 一体化
 - 脚本集成

核心部件技术规范 (CCTS)

这是 UN/CEFACT 发布的 [CCTS 规范](#)。

Enterprise Architect为在 CCTS 中建模提供了以下资源：

- UML框架：
 - 1 2.1 核心组件库
 - UPCC 3.0 核心组件库
 - UMM 2.0 业务需求、编排和信息视图。
- 业务部件库创建/管理
 - Schema 编辑器用于 ABIE 和 BDT 组合
 - 插件
 - 一体化
 - 脚本集成
- 业务部件架构组合
 - XSD、JSON 的Schema 编辑器
 - 插件
 - 一体化
 - 脚本集成

插件

框架 (自定义)

除了这些方法之外，Schema 编辑器还与Enterprise Architect自动化接口集成，以支持任何个人或团体实施他们自己的方法。一个插件

向Enterprise Architect注册提供架构生成功能的兴趣的公司将有机会在Schema 编辑器工具中提供其任何产品。

脚本Framework (自定义)

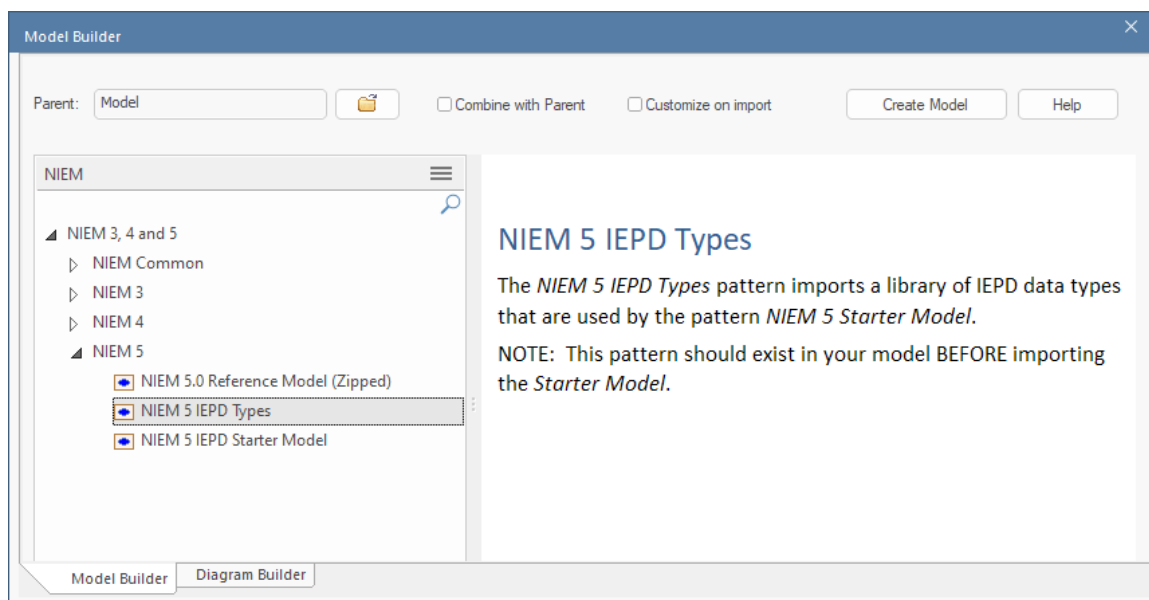
Schema 编辑器还为使用它创建的任何配置文件提供对模式生成的无条件控制。通过编写自己的脚本，作者可以访问任何模式的定义，并最终以他们选择的格式生成他们想要的任何文档。

安装核心框架

Enterprise Architect提供丰富多样的建模技术，包括Schema 编辑器中列出的所有标准。这些框架可用作UML模型和/或MDG技术，使用Enterprise Architect的模型生成器。模型本身也可直接从Enterprise Architect的可重用资产服务访问。

注记：如果您要建模通用解决方案，而不是直接使用核心框架（如 CIM 或 UBL ），则无需安装核心框架/模型。在这种情况下，最好使用具有属性的简单UML类来创建数据模型。

模型生成器



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	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 '汉堡包' 图标 >模型生成器（模式库）

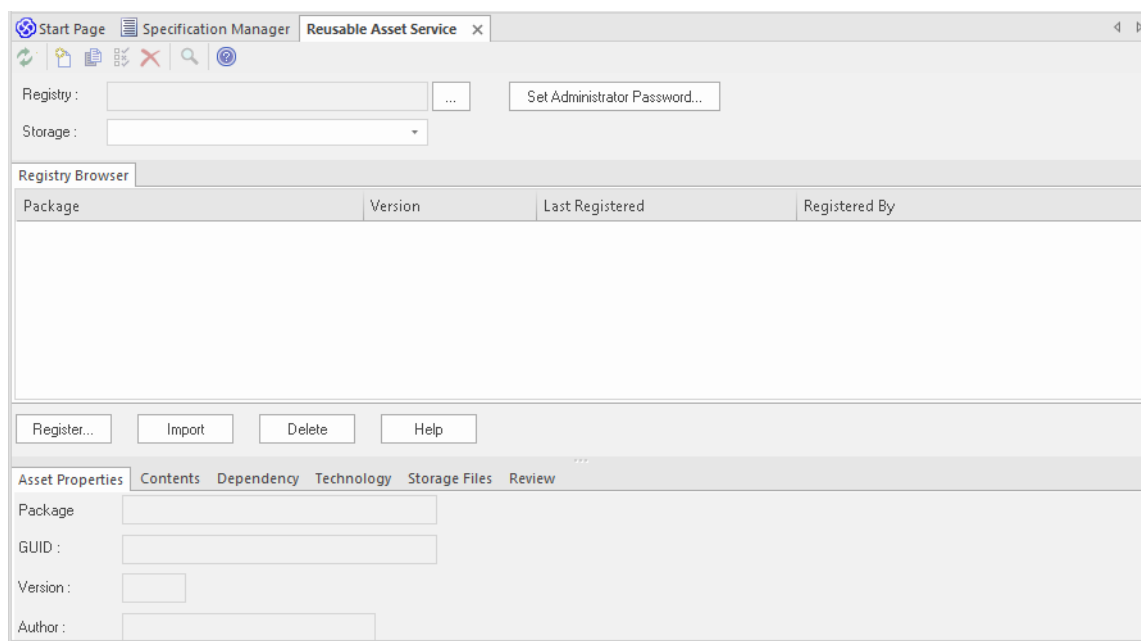
注记

您可以通过选择功能区选项来限制要使用的MDG 技术：“特定>技术> 管理技术”。在这里您可以看到当前启用了哪些技术。

导入模型

节	行动
1	显示模型生成器对话框。
2	单击 <perspective name> 按钮并选择所需的技术。
3	突出技术。
4	选择要导入的技术标准。
5	单击 创建模型按钮将框架导入到您的模型中。

可重用资产服务



访问

功能区	发布 > 模型交换 > Reusable Assets
-----	-----------------------------

导入模型

节	行动
1	连接到可重用资产服务。
2	从可用的存储库列表中选择
3	选择UML模型包
4	单击确定将选定的包导入您的模型。

架构Importer

您可以使用架构导入器将与Schema 编辑器兼容的架构导入Enterprise Architect 。架构导入器对架构进行验证，验证成功后创建一个架构类型的Schema 编辑器配置文件，可以直接在Schema 编辑器中查看。

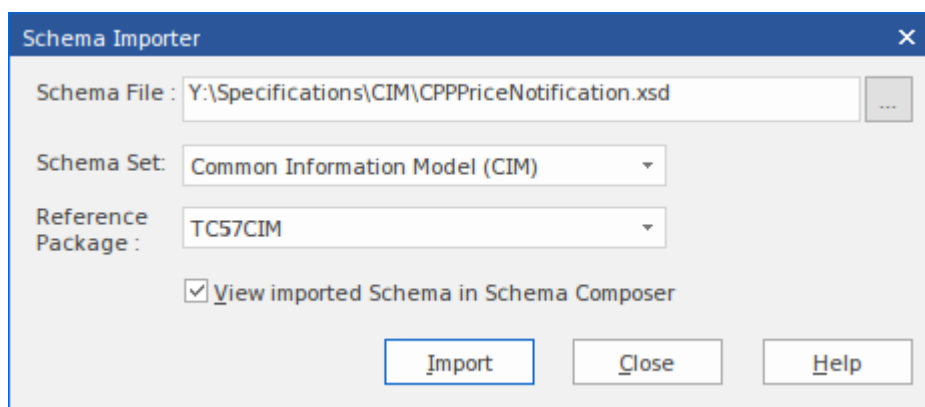
目前，您可以使用架构导入器来导入这些架构：

- 公共信息模型 (CIM) 特定的XML Schema
- 公共信息模型(CIM) 特定 RDFS XML

访问

功能区	开发>架构建模> Schema 编辑器>导入Schema 编辑器
-----	----------------------------------

使用架构导入器导入架构



选项	行动
架构文件	类型从中导入架构文件的目录路径和文件名。
架构集	选择要导入的架构类型。 目前架构Importer 支持导入特定的CIM： <ul style="list-style-type: none"> • XML Schema和 • RDFS XML
参考包	选择包含特定于模式集的公共元素的包。 架构导入器将根据参考包中的元素验证正在导入的架构中的元素。
在Schema 编辑器中视图导入的架构	选择此选项可在Schema 编辑器中打开导入的配置文件。
导入	单击此按钮开始导入过程。
关	单击此按钮可关闭 架构导入器“对话框”。

注记

- 系统输出窗口会显示导入进度
- Schema 编辑器会在导入架构之前根据参考包中的元素验证架构；如果验证失败，系统输出窗口将显示验证失败的架构元素，并停止导入过程
- 双击系统输出窗口中的验证错误条目以在Enterprise Architect的内部文件编辑器中打开架构并源错误的来源
- 如果验证成功，弹出“架构定义”对话框，您可以将导入的配置文件保存在文件系统中，也可以作为工件文件保存在当前模型中

Schema 编辑器自动化集成

Schema 编辑器可以从Enterprise Architect Automation接口访问。客户端A脚本或插件

) 可以使用 存储库“ object的 属性”属性获得对接口的访问。当Schema 编辑器加载了配置文件时，此接口可用。

Schema 编辑器Addin集成

Enterprise Architect插件可以与Schema 编辑器集成，通过提供替代方案为用户提供模式和子模型的生成。

Schema 编辑器脚本集成

尽管Schema 编辑器提供了基于各种流行技术的开箱即用的模式组合，但它的脚本集成为您提供了一些灵活性，让您可以实现自己的需求。您可以通过三种方式在Schema 编辑器中利用脚本：

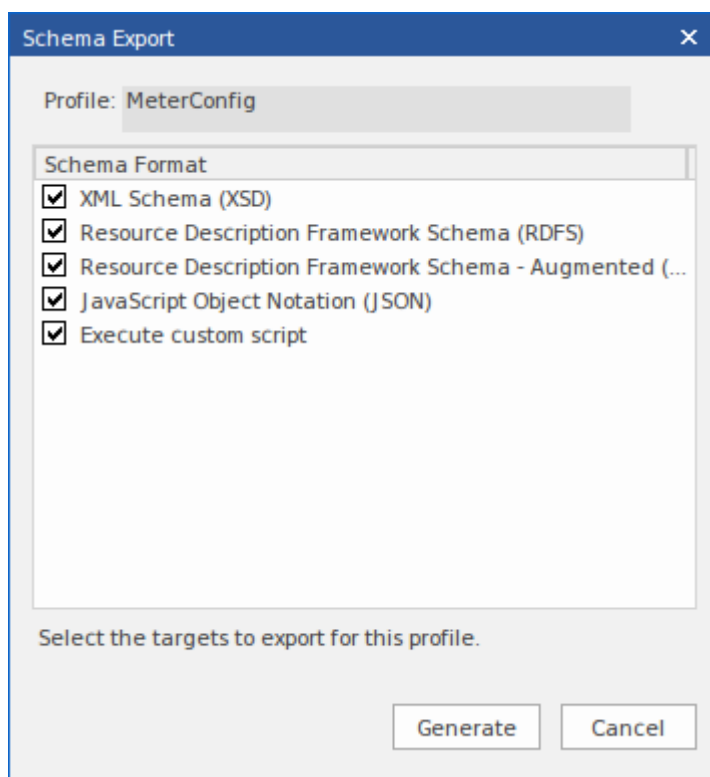
- 使用脚本语言提供自定义模式生成
- 使用脚本语言提供自定义模型转换
- 提供自定义原型映射到任何标准模型转换（例如 UPCC）

脚本模型变换

虽然Schema 编辑器为各种框架提供内置转换，但您始终可以编写自己的转换，使用 Composer 的组合工具设计模式，然后使用手工制作的脚本执行自定义转换。

架构脚本生成

当您在Schema 编辑器中选择一条消息并单击生成时，您会看到多种导出格式。这些选择之一是“执行自定义脚本”



架构迭代脚本示例

这个例子演示了在一个用JavaScript编写的Enterprise Architect脚本中访问Schema 编辑器。该脚本首先获取到Schema 编辑器的接口，然后遍历模式，打印出类型及其每个属性。

```
/*
*示例名称:脚本Schema 编辑器脚本
*作者 : Sparx Systems
* 目的 : 演示使用自动化和JavaScript访问Schema 编辑器
* 语言 : JavaScript
* 日期 : 2020
*/
函数 打印类型( xml 类型 , xmlns , 乌里 )
{
    变量 xmlProp 作为 艺电。架构属性 ;
    变量 xmlPropEnum 作为 艺电。 SchemaPropEnum ;
    变量 xmlChoiceEnum1 作为 艺电。模式类型枚举 ;
    变量 xmlChoiceEnum2 作为 艺电。模式类型枚举 ;

    会话.输出( "类型: " + xml 类型。类型名称 + 在命名空间中 : " + xmlns + " : " + uri );
    xmlPropEnum = xmlType .属性 ;
    如果( xmlPropEnum )
    {
        xmlProp = xmlPropEnum 。获取第一 ( ) ;
        而( xmlProp )
        {
            if ( xmlType . IsEnumeration () )
            {
                会话.输出( " " + xmlProp .名称 ) ;
            }
            else
            {
                变量 sPropDesc = xmlProp .名称 ;
                sPropDesc += " : " ;
                if ( xmlProp . IsPrimitive () )
                    sPropDesc += xml 属性。原始类型 ;
                else
                    sPropDesc += xml 属性。类型名称 ;

                if ( xmlProp . IsByReference () )
                {
                    sPropDesc += " ( 通过引用 ) " ;
                }
                if ( xmlProp . IsInline () )
                {
                    sPropDesc += " ( 内联 ) " ;
                }
            }
        }
    }
}
```

```
    会话.输出( " " + sPropDesc + " , 基数 : " + xml 属性。基数 ) ;

    xmlChoiceEnum1 = xml 属性。选择 ;
    xmlChoiceEnum2 = xml 属性。模式选择 ;
    变量 数数 = xmlChoiceEnum1 。获取计数() + xmlChoiceEnum2 。获取计数 (
) ;

    如果(计数 > 1 )
    {
        会话.输出 ( 选择 : " ) ;
        xml选择 = xmlChoiceEnum1 。获取第一 ( ) ;
        而( xmlChoice )
        {
            会话.输出( " " + xml 选择。类型名称 ) ;
            xml选择 = xmlChoiceEnum1 。获取下一个 ( ) ;
        }
        xml选择 = xmlChoiceEnum2 。获取第一 ( ) ;
        而( xmlChoice )
        {
            会话.输出( " " + xml 选择。类型名称 ) ;
            xml选择 = xmlChoiceEnum2 。获取下一个 ( ) ;
        }
    }
    xmlProp = xmlPropEnum 。获取下一个 ( ) ;
}
}
```

函数 主要 ()

```
{
    变量 图式 作为 艺电。架构作曲家 ;
    变量 xml类型 作为 艺电。模式类型 ;
    变量 xmlTypeEnum 作为 艺电。模式类型枚举 ;
    变量 xml命名空间枚举 作为 艺电。模式名称空间枚举 ;
    变量 xmlNS 作为 艺电。模式命名空间 ;

    // 获取 SchemaComposer
    图式 = 存储库。架构作曲家 ;

    // 打印命名空间引用
    xml命名空间枚举 = 架构。命名空间 ;
    如果( xmlNamespaceEnum )
```



```

{
    xmlNS = xmlNamespaceEnum 。 获取第一 ( ) ;
    而( xmlNS )
    {
        会话.输出( “xmlns : ” + xmlNS .名称 + “URI=” + xmlNS 。 URI );
        xmlNS = xmlNamespaceEnum 。 获取下一个 ( ) ;
    }
}
// 获取架构类型枚举器
xmlTypeEnum = 架构。 模式类型 ;
xml类型 = xmlTypeEnum 。 获取第一 ( ) ;
而( xmlType )
{
    变量 xmlns = 架构。 GetNamespacePrefixForType ( xml 类型。 类型ID );
    乌里 = 架构。 GetNamespaceForPrefix ( xmlns );
    打印类型( xmlType , xmlns , uri );
    xml类型 = xmlTypeEnum 。 获取下一个 ( ) ;
}
}

```

主要 () ;

智能感知帮助脚本

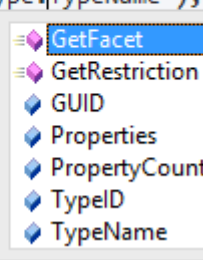
编辑Enterprise Architect脚本帮助您使用Schema 编辑器属性编写脚本，方法是在智能感知脚本中提供编辑脚本或其自动化接口的交互和方法。

```

// Enumerate Types
xmlType = umlModelTypeEnum.GetFirst();
while(xmlType)
{
    print( "Type: " + xmlType.TypeName );

    umlPropEnum = xmlType.
    if(umlPropEnum)
    {
        umlProp = umlPropE
        while(umlProp)
        {
            if(umlProp.IsP

```



模型变换中的构造型映射

构造型构成了MDG 技术方法的很大一部分。MDG 技术的单独UML配置文件定义了构造型以为其元素提供有用

的分类。从核心框架转到业务模型或子域时，重新分配构造型是一个常见的要求。当您使用 CCTS 框架时，您生成的业务组件的原型由Enterprise Architect根据 CCTS 规范定义的映射（例如，ACC 到 ABIE）自动生成。

当您在Schema 编辑器中打开或创建模型转换配置文件时，您可以指定一个脚本来为您执行此映射。可以从属性窗口中选择脚本。

Schema Type	Transform
Schema Set	Core Components (UN/CEFACT) - NDR 3.0
Transform Rule Script	Schema Composer.BDTTransformRule ...
Qualifier	
BDTLibrary	BDTLibrary
BIELibrary	BIELibrary

该脚本可以用JavaScript、JScript 或 VBScript 编写，并且只需要实现这个函数（这里用JavaScript表示法描述）：

函数TranslateStereotype(srcStereo)

```
{  
var destStereo = srcStereo  
if (srcStereo == "BDT")  
{  
destStereo = "My_BDT"  
}  
返回目标立体声；  
}
```

MDG 技术- UML配置文件Extensions

Schema 编辑器使用MDG技术。它用于模式生成的标准（除 *Generic* 外）仅对遵循该框架的模型有意义。然而，扩展现有的MDG 技术非常容易。确保在您的业务特定域或子域中创作的元素提供一致命名的元数据或 标记值”。

Schema 编辑器通过其脚本集成支持对UML 配置文件/框架的扩展。当在Schema 编辑器中分配一个脚本时，转换过程将调用该脚本并要求它翻译关键字。这些关键字通常是UML原型。如果特定技术与配置文件相关联，Schema 编辑器将调用此函数，将MDG 技术的名称传递给它。

该脚本可以返回输入名称，并且不会发生映射，或者它可以返回另一个MDG 技术的名称。发生这种情况时，Schema 编辑器将再次要求该函数选择性地映射任何UML 配置文件。最后，它会要求脚本翻译来自核心技术的刻板印象。

模型转换的结果将是子模型的任何UML元素将显示除了任何核心标记值之外的扩展标记值。

映射MDG 技术的示例脚本

函数TranslateStereotype (stereo)

```
{  
var newStereo = 立体声 ;  
如果 ( 立体声 == “UPCC3” )  
{  
newStereo = “XXX UPCC3”  
}  
返回新立体声 ;  
}
```

映射UML配置文件的示例脚本

函数TranslateStereotype (stereo)

```
{  
var newStereo = 立体声 ;  
if (stereo == “UPCC3 - BIE 库抽象语法”)  
{  
newStereo = “UPCC3 - BIE 库 XXX 语法”  
}  
返回新立体声 ;  
}
```

映射UML构造型的示例脚本

函数TranslateStereotype (stereo)

```
{  
var newStereo = 立体声 ;
```

```
如果 ( 立体声="ABIE" )  
{  
newStereo = "XXX ABIE";  
}  
返回新立体声 ;  
}
```

XSD 模型

XML Schema Definition (XSD), 也称为XML Schema, 是一种 World Wide网络Consortium (W3C) XML 技术, 用于指定 XML 文档必须遵守的规则。XSD 支持对于开发完整的完成服务的架构(SOA) 至关重要, UML 2.5 和 XML 的耦合提供了在组织内指定、构建和部署基于 XML 的 SOA 工件的自然机制。

XSD 的UML 配置文件指定了一组可以应用于UML 模型的构造型、标记值和约束, 以更改结果模式的特定方面。Enterprise Architect通过工具箱的XML Schema页面为图表配置文件提供原生支持。Enterprise Architect支持的 XSD配置文件是对出版物建模Applications with UML中定义的配置文件的改编。

通过Enterprise Architect使用 XSD配置文件, 您可以快速模型、正向工程和反向工程XML Schema。

您还可以使用Enterprise Architect Schema 编辑器快速定义和生成 XSD 和其他模式。

建模XSD

您可以使用以下UML类图在两个级别上模型XML 模式：

- 没有特定于 XML 模式的实现细节，由Enterprise Architect的架构生成器直接生成；生成器应用一组默认映射将抽象模型包转换为 W3C XML Schema (XSD) 文件
- 使用图表工具箱的“XML Schema”页面，使用 XML 模式特定的定义进行图表，该工具箱提供 XSD 的 UML配置文件的结构

对XML Schema进行模型

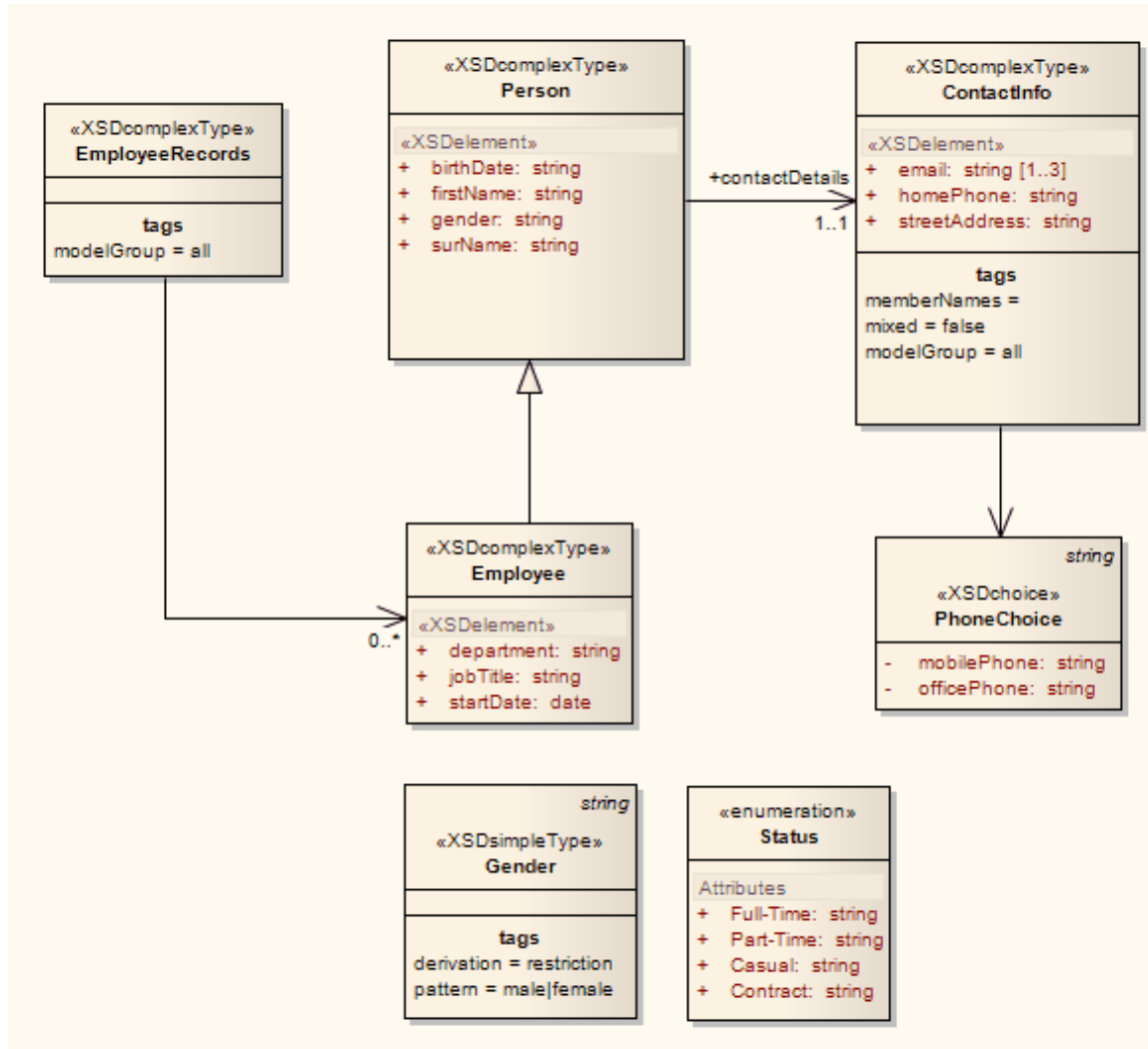
节	行动
1	在浏览器窗口中，创建您需要的顶层项目结构（模型和视图），然后单击相应的视图。
2	单击浏览器窗口标题下拉菜单中的“新包”选项。将显示“新模型包”对话框。
3	在“名称”字段中输入新包的名称，然后选择“创建图表”单选按钮。 单击确定按钮。将显示“新图表”对话框。
4	在“名称”字段中输入新图表的名称。 在“Select From”面板中选择“UML Structural”，在“图表”面板中选择“类”。
5	单击确定按钮。在浏览器窗口中，双击新图表名称旁边的图标；图表在“图表视图”中打开，类“页面显示在“图表工具箱”中。 此时，您可以： <ul style="list-style-type: none"> • 使用类工具箱图标创建类图，或 • 使用工具箱的“XML Schema”页面创建定制的图表XML Schema图（继续第6步）
6	单击  以显示“查找工具箱项”对话框并指定“XML Schema”以显示“XML Schema”工具箱页面。
7	单击工具箱中的“架构”图标并将其拖到类图中。 将显示“XSD 模式属性”对话框。 完全此对话框，然后单击确定按钮。 将显示“新图表”对话框。
8	同样，在“名称”字段中输入新图表的名称。 在“Select From”面板中选择“UML Structural”，在“图表”面板中选择“类”。 单击确定按钮。
9	在浏览器窗口和图表上创建了一个XSDschema原型包，带有一个子类图表。 双击图表上的包打开子类图表，并使用XML Schema工具箱中的构造来模型XML Schema。

注记

- 类的UML属性直接映射到 XML 元素或属性
- 如果您已将 XSD架构建模为直类图，则可以使用Schema 编辑器定义和生成架构
- XML Schema模型中的类没有方法，因为类方法和 XSD 构造之间没有有意义的对应关系
- 建模限制 - 这些XML Schema构造不能在Enterprise Architect中建模：
 - 应用信息
 - 场地
 - 钥匙
 - keyref
 - 符号
 - 重新定义
 - 选择器
 - 替代组
 - 独特的

XSD图表

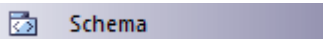
此示例图显示了一个类图，其中包含使用工具箱的“XSD架构箱”页面创建的图表特定元素。该图模拟了一个员工记录系统。



架构包


«XSDschema» 原型包充当 XSD 构造的容器，可以从中生成XML Schema。包中的所有类都在一个模式中定义；架构元素提供默认的架构范围设置。您可以通过从XML Schema工具箱中拖动架构图标并将其直接放到图表上来创建一个 «工具箱»包。

工具箱图标




访问

要显示选定 «XSDschema» 原型包的 XSD 模式 属性”对话框，请使用此处概述的方法之一：

功能区	设计>包>管理>属性
上下文菜单	右键单击 «XSDschema» 定型包 属性
其它	在浏览器窗口中，双击现有的 «XSDschema» 原型包，或 将  Schema 图标从工具箱拖到图表上（这将创建一个新的 «XSDschema» 原型包）

定义属性

字段/按钮	行动
架构名称	如果您不想使用模式包的默认名称，请用另一个名称改写它。
目标命名空间	(可选) 此架构包的目标命名空间中的类型。
前缀	(可选) 类型在缩写文本中代表目标命名空间。
默认命名空间	(可选) 所有无前缀 XSD 元素和 XSD 属性的默认命名空间中的类型。
架构文件	类型在或浏览 (单击 ) 要生成此包的XML Schema文件的文件路径。
XMLNS	<p>识别此架构包中使用的其他命名空间或命名空间前缀对。</p> <p>要添加命名空间或命名空间前缀对，请单击 新建”按钮；要编辑现有条目，请双击它。在任何一种情况下，都会显示 命名空间详细信息”对话框。</p> <ul style="list-style-type: none"> 前缀类型在缩略文本中代表命名空间 命名空间-命名命名空间名称中的类型 确定- 单击此按钮保存新信息并关闭“命名空间详细信息”对话框 取消 - 单击此按钮可放弃新信息并关闭“命名空间详细信息”对话框

	<ul style="list-style-type: none">帮助- 点击此按钮显示此帮助主题 要从列表中删除条目，请单击它并单击删除按钮。
确定	单击此按钮保存输入的模式数据并关闭 XSD 模式“属性”对话框。
取消	单击此按钮可放弃输入的模式数据并关闭 XSD 模式“属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	在编辑现有架构包信息时会显示此按钮。 单击 按钮打开架构元素的UML元素“属性”对话框。

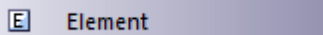
注记

- 默认的模式范围设置由标记值定义，您可以在模式元素审阅的 标签“属性”选项卡或元素的属性窗口中查看查看；如果需要，您可以编辑模式范围的设置，或者在各个 XSD 构造元素的属性和标记值中提供元素特定的覆盖

全球元素

«XSDtopLevelElement» 原型类充当 XSD 全局元素。您可以通过从XML Schema工具箱中拖动“元素”图标并将其直接放到图表上来创建它。

工具箱图标




访问

要为选定的 «XSDtopLevelElement» 原型类显示“XSD元素属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «XSDtopLevelElement» 原型类 属性
键盘快捷键	Alt+Enter
其它	在浏览器窗口中，双击 «XSDtopLevelElement» 原型类，或将  图标从工具箱拖到图表上（这将创建一个新的 «XSDtopLevelElement» 原型类）

定义属性

字段/按钮	行动
名称	如果您不想使用全局元素的默认名称，请使用其他名称改写它。
类型	任何一个： <ul style="list-style-type: none"> 类型数据类型的名称，或 单击下拉箭头并从列表中选择 XSD 内置数据类型，或 单击  按钮并浏览现有的 XSD 分类器元素，或 选择这两个复选框之一
嵌套的 complexType	选中此复选框以创建一个 XSDcomplexType 作为此全局元素的子元素。
嵌套的 simpleType	选中此复选框可创建一个 XSDsimpleType 作为此全局元素的子元素。
价值	(可选) 如果您在“类型”字段中输入了 XSD 内置数据类型，请输入一个值。
默认	选择此单选按钮可将值设置为默认值。

固定的	选择此单选按钮可将值设置为固定值。
注解	(可选) 类型在您需要此元素的任何注记中。
确定	单击此按钮保存输入的元素数据并关闭 XSD元素的“属性”对话框。
取消	单击此按钮可丢弃输入的元素数据并关闭 XSD元素的“属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 XSD元素信息时，会显示此按钮。 单击按钮打开全局元素的UML元素“属性”对话框。

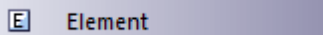
注记

- '类型'、'Nested complexType' 和 'Nested simpleType' 字段是互斥的；选择一个禁用其他
- 只有在创建新的全局元素时（而不是在编辑全局元素时），对话框中的“嵌套复杂类型”和“嵌套简单类型”字段才可用
- A全局元素：
 - 不能包含任何UML属性
 - 不能是源连接器的关联
 - 可以是复杂类型类或组元素类的关联连接器的目标
 - 不能成为概括器的目标
 - 可以是复杂类型类或简单类型类的一个概括器的源

当地元素

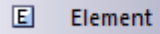
本地元素是A «XSDelement» 原型UML属性，它充当本地 XSD元素。您可以通过将 元素”图标从XML Schema工具箱中拖放到 «XSDcomplexType» 或 «XSDgroup» 原型类上来创建它。

工具箱图标




访问

要为选定的 «XSDelement» 原型UML属性显示 “XSD元素属性”对话框，请使用此处概述的方法之一。

功能区	在图表中选择特定的 «XSDelement» 原型属性： 设计>元素>特征>属性
上下文菜单	在图表中选择特定的 «XSDelement» 原型属性： 右键单击属性 视图属性
键盘快捷键	在图表中选择特定的 «XSDelement» 原型属性： F9
其它	双击 «XSDelement» 原型属性，或 将  图标拖到 «XSDcomplexType» 或 «XSDgroup» 原型类上，（这会在类中创建一个新的 «类»）

定义属性

字段/按钮	行动
名称	如果您不想使用本地元素的默认名称，请使用其他名称改写它。
类型	任何一个： <ul style="list-style-type: none"> 类型数据类型的名称，或 单击下拉箭头并从列表中选择 XSD 内置数据类型，或 单击  按钮并浏览现有的 XSDcomplexType 或 XSDsimpleType元素作为分类器
参考	（可选）在生成的 XSD 中指定是否使用 ref 属性（而不是 type 属性）来引用您在 “类型”字段中选择的 XSDcomplexType 或 XSDsimpleType元素。
价值	（可选）如果您在 “类型”字段中输入了 XSD 内置数据类型，请输入一个值。

默认	选择此单选按钮可将值设置为默认值。
固定的	选择此单选按钮可将值设置为固定值。
MinOccurs	(可选) 类型该元素必须在类中出现的最小次数。 类型'0' 表示该元素是可选的。 默认值为 ' 1 '。
最大发生次数	(可选) 类型该元素可以在类中出现的最大次数。 类型无界表示元素可以出现的次数没有限制。 默认值为1。
形式	(可选) 单击下拉箭头并选择是否限定元素 : <ul style="list-style-type: none"> • 合格 -使用架构包中定义的前缀来限定此元素 • unqualified - 不限定此元素
注解	(可选) 在您需要的任何注记类型输入此本地元素。
确定	单击此按钮保存输入的元素数据并关闭 XSD元素的“属性”对话框。
取消	单击此按钮可丢弃输入的元素数据并关闭 XSD元素的“属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 XSD元素信息时，会显示此按钮。 点击按钮打开本地元素的属性属性。

注记


- 只有 «Complex Type»、«Group» 和 «Model Group» 原型元素可以具有此UML属性

全局属性


全局属性是A «XSDtopLevelAttribute» 原型类。您可以通过从XML Schema工具箱中拖动 属性”图标并将其直接放到图表上来创建它。

访问


要为选定的 «属性» 原型元素显示 “XSD 属性属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «XSDtopLevelAttribute» 原型元素 属性
键盘快捷键	Alt+Enter
其它	双击 «XSDtopLevelAttribute» 原型类，或 从工具箱中拖动  Attribute 图标并直接拖放到图表上（这将创建一个新的 «XSDtopLevelAttribute» 原型类）

工具箱图标

 Attribute

定义属性

字段/按钮	行动
名称	如果您不想使用全局属性的默认名称，请用另一个名称改写它。
类型	任何一个： <ul style="list-style-type: none"> • 类型数据类型的名称，或 • 单击下拉箭头并从列表中选择 XSD 内置数据类型，或 • 单击  按钮并浏览现有的 XSDsimpleType元素作为分类器或者，选中 嵌套的 simpleType”复选框。
嵌套的 simpleType	选中此复选框可创建一个 XSDsimpleType元素作为此全局属性元素的子元素。
价值	（可选）如果您在 类型”字段中选择了 XSD 内置数据类型，请输入一个值。
默认	选择此单选按钮可将 值”字段设置为默认值。
固定的	选择此单选按钮可将 值”字段设置为固定值。

形式	(可选) 单击下拉箭头并选择： <ul style="list-style-type: none">• 有资格使用提供的架构前缀包限定此属性• unqualified 在属性名称上不显示限定前缀
注解	(可选) 类型在任何注记中您需要此属性。
确定	单击此按钮保存输入的属性数据并关闭 XSD 属性“属性”对话框。
取消	单击此按钮可放弃输入的属性数据并关闭 XSD 属性“属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 XSD 元素信息时，会显示此按钮。 单击按钮打开全局属性类的UML元素“属性”对话框。

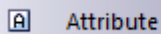
注记

- 只有在创建新的全局属性时（而不是在编辑全局属性时），对话框中的 嵌套 simpleType”字段才可用
- 类型”和 嵌套的简单类型”字段是互斥的；选择一个禁用另一个
- 全局属性A
 - 不能包含任何UML属性
 - 不能是源连接器的关联
 - 可以是复杂类型类的关联连接器的目标
 - 不能成为概括器的目标
 - 可以是一个概括类型类的连接器的源

本地属性

本地属性是A «XSDattribute» 原型UML属性。您可以通过从XML Schema工具箱中拖动 属性”图标并将其放到 «XSDcomplexType» 或 «XSDattributeGroup» 原型类上来创建它。

工具箱图标




访问

要为选定的 «属性» 原型UML属性显示 XSD 属性属性”对话框，请使用此处概述的方法之一。

功能区	在图表上选择特定的 «XSDattribute» 原型UML属性： 设计>元素>特征>属性
上下文菜单	在图表上选择特定的 «XSDattribute» 原型UML属性： 右键单击属性 视图属性
键盘快捷键	在图表上选择特定的 «XSDattribute» 原型UML属性： F9
其它	双击 «XSDattribute» 原型UML属性，或 将  Attribute 图标从工具箱拖放到 «XSDcomplexType» 或 «XSDattributeGroup» 原型类（这将创建一个新的 «XSDattribute» 原型UML属性）

定义属性

字段/按钮	行动
名称	如果您不想使用本地属性的默认名称，请用另一个名称改写它。
类型	任何一个： <ul style="list-style-type: none"> 类型数据类型的名称，或 单击下拉箭头并从列表中选择 XSD 内置数据类型，或 单击  按钮并浏览现有的 XSDsimpleType元素作为分类器
参考	（可选）指定是否在生成的 XSD 中使用 ref 属性（而不是 type 属性）来引用您在“元素类型”字段中选择的 XSDsimpleType。
价值	（可选）如果您在“类型”字段中输入了 XSD 内置数据类型，请输入一个值。

默认	选择此单选按钮可将值设置为默认值。
固定的	选择此单选按钮可将值设置为固定值。
形式	(可选) 单击下拉箭头并选择是否限定属性： <ul style="list-style-type: none">• 限定 -使用架构前缀中包的限定此属性• unqualified - 不限定此属性
注解	(可选) 在您需要的任何注记类型输入此本地属性。
确定	单击此按钮保存输入的属性数据并关闭 XSD 属性“属性”对话框。
取消	单击此按钮可丢弃输入的元素数据并关闭 XSD 属性“属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	在编辑现有 XSD 属性信息时会显示此按钮。 点击按钮打开本地属性的属性属性。

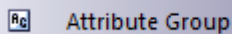
注记

- 只有复杂类型和属性组可以有这个UML属性

属性组

属性组类用于对一组 «XSDattribute» 原型UML属性和可以从 «类型» 原型类引用的 SimpleTypeClasses 进行分组。您可以通过从XML Schema工具箱中拖动 “属性组” 图标并将其直接放到图表上来创建它。

工具箱图标


 The image shows a toolbar icon for 'Attribute Group'. It consists of a small square icon with a stylized 'A' and 'G' inside, followed by the text 'Attribute Group' in a sans-serif font.

访问

要为选定的 «属性» 原型类显示 “XSD 属性组属性” 对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击«XSDattributeGroup»原型类 属性
键盘快捷键	Alt+Enter
其它	双击 «XSDattributeGroup» 原型类，或 将  图标从工具箱拖到图表上（这将创建一个新的 «XSDattributeGroup» 原型类）

定义属性

字段/按钮	行动
名称	如果您不想使用属性组的默认名称，请用另一个名称改写它。
注解	（可选）此属性组所需的任何注记中的类型。
确定	单击此按钮保存输入的属性组数据并关闭 XSD 属性组的 “属性” 对话框。
取消	单击此按钮可丢弃输入的属性组数据并关闭 XSD 属性组的 “属性” 对话框。
帮助	单击此按钮可显示此帮助主题。
UML	在编辑现有 XSD 属性组信息时会显示此按钮。 单击按钮打开属性组的UML元素“属性”对话框。

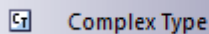
注记

- 一个属性组元素：
 - 不能是任何其他 XSD类的孩子
 - 只能包含 «类型» 原型UML属性和 SimpleTypeClasses
 - 可以是另一个属性组的关联连接器的源
 - 可以是复杂类型类的关联连接器的目标
 - 不能成为概括或连接的源或目标

复杂类型

«XSDcomplexType» 构造型应用于通用UML类，以定制架构中 `complexType` 定义的生成。您可以通过将复杂类型图标从XML Schema工具箱中拖放到图表上来创建一个 «类» 原型。

工具箱图标



访问

要为选定的 «XSDcomplexType» 原型类显示 “XSD complexType属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «XSDcomplexType» 原型类 属性
键盘快捷键	Alt+Enter
其它	双击 «XSDcomplexType» 原型类，或 将  图标从工具箱拖到图表上（这将创建一个新的 «XSDcomplexType» 原型类）

定义属性

字段/按钮	行动
名称	如果您不想使用 <code>complexType</code> 类的默认名称，请使用其他名称改写它。
集团模型	单击向下箭头并选择定义此 <code>complexType</code> 的子元素应如何出现在架构中的选项。 <ul style="list-style-type: none"> '序列' - 子元素必须以指定的顺序出现 'choice' - 只能出现一个子元素 'all' - 子元素可以以任何顺序出现
MinOccurs	(可选) 类型该元素必须在类中出现的最小次数。 类型0 表示该元素是可选的。 默认值为1。
最大发生次数	(可选) 类型该元素可以在类中出现的最大次数。 类型无界表示元素可以出现的次数没有限制。 默认值为1。

注解	(可选) 类型任何注记你需要的这个元素。
抽象的	(可选) 选中此复选框以在实例 XML 文件中使用此 complexType。
混合	(可选) 选中此复选框以允许字符数据显示在子元素中。
确定	单击此按钮保存输入的 complexType 数据并关闭 XSD complexType 的“属性”对话框。
取消	单击此按钮可丢弃输入的 complexType 数据并关闭 XSD complexType 的“属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 XSD complexType 信息时，会显示此按钮。 单击按钮打开 complexType类的UML元素“属性”对话框。

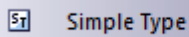
注记

- A complexType 可以：
 - 包含 XSDelement 和 XSDattribute 原型UML属性
 - 包含其他 complexTypes 作为子元素
 - 成为全球元素的孩子
 - 成为其他复杂类型、简单类型、属性组、组和模型组的关联连接器源
 - 最多有一个概括源或另一个复杂类型或简单类型类

简单类型

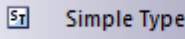
«XSDsimpleType» 构造型应用于通用UML类，以定制架构中 simpleType 定义的生成。您可以通过从XML Schema工具箱中拖动 SimpleType 图标并将其直接放到图表上来创建一个 «类型»类。

工具箱图标



访问

要为选定的 «XSDsimpleType» 原型类显示 “XSD simpleType属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «XSDsimpleType» 原型类 属性
键盘快捷键	Alt+Enter
其它	双击 «XSDsimpleType» 原型类，或 将  图标从工具箱拖到图表上（这将创建一个新的 «XSDsimpleType» 原型类）

定义属性

字段/按钮	行动
名称	如果您不想使用 simpleType元素的默认名称，请使用其他名称改写它。
类型	任何一个： <ul style="list-style-type: none"> 类型数据类型的名称，或 单击下拉箭头并从列表中选择 XSD 内置数据类型，或 单击  按钮并浏览现有的«XSDsimpleType»元素作为分类器
限制	选择此单选按钮可将此 simpleType 的值限制为所选类型的值。 类的各种限制（方面）可作为此类的标记值使用。
列表	选择此单选按钮可将此 simpleType 指定为所选类型的值列表。
注解	（可选）类型任何注记你需要的这个元素。
确定	单击此按钮保存输入的 simpleType 数据并关闭 XSD simpleType 的“属性”对话

	框。
取消	单击此按钮可丢弃输入的 simpleType 数据并关闭 XSD simpleType 的“属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 XSD simpleType 信息时，会显示此按钮。 单击按钮打开simpleType类的UML元素“属性”对话框。

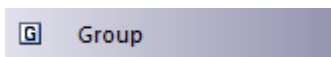
注记

- 简单类型A
- 不能包含任何 «XSDelement» 或 «XSDattribute» 原型UML属性
- 不能包含任何子类
- 不能是源连接器的关联
- 可以成为一个概括器的目标
- 最多可以有一个概括器到另一个简单类型类

团体

Group类用于对一组 «类型» 原型UML属性、ComplexTypeClasses 和 SimpleTypeClasses 进行分组，这些属性可以从 «类型»类中引用。您可以通过从XML Schema工具箱中拖动 Group 图标并将其直接放到图表上来创建这种类型的元素。

工具箱图标



访问

要为选定的 «XSDgroup» 原型类显示 “XSD 组属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击«XSDgroup»原型类 属性
键盘快捷键	Alt+Enter
其它	双击 «XSDgroup» 原型类，或 将  图标从工具箱拖到图表上（这将创建一个新的 «XSDgroup» 原型类）

定义属性

字段/按钮	行动
名称	如果您不想使用组元素的默认名称，请用其他名称改写它。
集团模型	单击下拉箭头并选择定义该组的子元素应如何出现在复杂类型类中的值： <ul style="list-style-type: none"> 序列指定子元素必须以指定的顺序出现 选择 - 指定只能出现一个子元素 all - 指定子元素可以以任何顺序出现
注解	(可选) 类型任何注记你需要的这个元素。
确定	单击此按钮保存输入的组数据并关闭 XSD 组的 “属性”对话框。
取消	单击此按钮可丢弃输入的组数据并关闭 XSD 组 “属性”对话框。
帮助	单击此按钮可显示此帮助主题。

UML	在编辑现有 XSD 组信息时会显示此按钮。 单击按钮打开 Group类的UML元素“属性”对话框。
-----	--

注记

- A Group元素可以：
- 仅包含 «XSDelement» 原型UML属性
- 包含复杂类型和简单类型作为子元素
- 成为其他复杂类型、简单类型和组的关联连接器的源
- 成为复杂类型元素的关联连接器的目标
- 概括连接或连接的源或目标

任何


«XSDany» 构造型类允许复杂类型类包含架构包中未指定的元素。您可以通过从XML Schema工具箱中拖动 Any 图标并将其直接放到图表上来创建它。

工具箱图标



访问

要为选定的 «XSDany» 原型类显示 “XSD 任何属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «XSDany» 刻板印象类 属性
键盘快捷键	Alt+Enter
其它	双击 «XSDany» 原型类，或 将  图标从工具箱拖到图表上（这将创建一个新的 «XSDany» 原型类）

定义属性

字段/按钮	行动
名称	如果您不想使用 Any元素的默认名称，请用另一个名称改写它。
命名空间	(可选) 类型命名空间包含可以在复杂类型中使用的元素。
过程内容	(可选) 单击下拉箭头并选择定义 XML解析器应如何验证这些元素的值： <ul style="list-style-type: none"> • 松懈 - 尝试根据其架构验证元素；获取不到架构时不提示错误 • skip - 跳过验证元素 • 严格 - 根据其架构验证元素并在无法获得架构时标记错误
注解	(可选) 类型任何注记你需要的这个元素。
确定	单击此按钮以保存输入的任何“元素数据并关闭 XSD 任何“属性”对话框。
取消	单击此按钮可丢弃输入的任何元素数据并关闭 XSD 组 “属性”对话框。
帮助	单击此按钮可显示此帮助主题。

UML	当您编辑现有的 «XSDany»元素信息时，会显示此按钮。 单击按钮打开 Any类的UML元素“属性”对话框。
-----	--

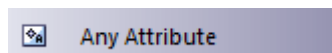
注记

- 任何类：
- 不能包含任何UML属性或子 XSD 类
- 不能是任何 XSD类的子级
- 不能成为概括器的目标
- 不能是源或概括或连接的关联
- 可以是复杂类型、组和模型组的关联连接器的目标
- 必须是至少一个传入关联连接器的目标

任何属性


«XSDany» 原型UML属性允许复杂元素类型或属性组元素包含未在架构包中指定的属性。您可以通过从XML Schema工具箱中拖动 “Any Attribute” 图标并将其放到 «XSDcomplexType» 或 «XSDattributeGroup» 原型类上来创建它。

工具箱图标



访问

要为选定的 «XSDany» 原型UML属性显示 “XSD anyAttribute属性”对话框，请使用此处概述的方法之一。

功能区	在图表上选择特定的 «XSDany» 原型UML属性： 设计>元素>特征>属性
上下文菜单	在图表上选择特定的 «XSDany» 原型UML属性： 右键单击属性 视图属性
键盘快捷键	在图表上选择特定的 «XSDany» 原型UML属性： F9
其它	双击 «XSDany» 原型UML属性，或 从工具箱中拖动  Any Attribute 图标并将其拖放到 «XSDcomplexType» 或 «XSDattributeGroup» 原型类（这将创建一个新的 «XSDany» 原型UML属性）

定义属性

字段/按钮	行动
名称	如果您不想使用属性的默认名称，请用另一个名称改写它。
命名空间	（可选）类型命名空间包含可以在复杂类型或属性组元素中使用的属性。
过程内容	（可选）单击下拉箭头并选择定义 XML解析器应如何验证这些属性的值： <ul style="list-style-type: none"> • 松懈 - 尝试根据其架构验证属性；获取不到架构时不提示错误 • skip - 跳过验证属性 • 严格 - 根据其架构验证属性并在无法获得架构时标记错误
注解	（可选）类型any注记你需要这个属性。

确定	单击此按钮保存输入的属性数据并关闭 XSD anyAttribute '属性' 对话框。
取消	单击此按钮可丢弃输入的属性数据并关闭 XSD anyAttribute '属性' 对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您正在编辑现有的 «XSDany» 属性信息时，会显示此按钮。 单击按钮打开 «属性» 属性的属性属性。


注记

- 只有复杂类型和属性组元素可以有这个UML属性

联盟


联合类是一个简单类型元素，它定义了一个简单类型的集合。您可以通过从XML Schema工具箱中拖动 Union 图标并将其直接放在图表上来创建它。

工具箱图标


 Union

访问

要为选定的 «XSDUnion» 原型类显示 «XSD 联合属性» 对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «XSDUnion» 刻板印象类 属性
键盘快捷键	Alt+Enter
其它	双击 «XSDUnion» 原型类，或 将  Union 图标从工具箱拖到图表上（这将创建一个新的 «XSDUnion» 原型类）

定义属性

字段/按钮	行动
名称	如果您不想使用联合的默认名称，请用另一个名称改写它。
会员类型	<p>单击  按钮以显示 «XSD 联合成员» 对话框，并选择内置 XSD 数据类型和简单类型元素作为集合的成员。</p> <ul style="list-style-type: none"> 选择 - 不是在“类名称”字段中键入或选择值，而是单击此按钮以显示“选择分类器”浏览器并找到并选择一个简单类型元素；单击确定按钮关闭浏览器并立即将所选元素添加到“类型详细信息”列表 该选项一般用于指定与 Union 元素在同一个包中的对象，但您也可以选择任何其他包中的对象 添加 - 单击此按钮可将“类名称”字段中指定的数据类型或元素添加到“类型详细信息”列表 接受分类器即使不在模型中 - 选中此复选框以包含已命名但不存在于与联合元素相同的模型包中的元素或数据类型 类型详细信息 - 审阅选定元素或数据类型的列表；如果您打算从列表中删除一个 object，请突出显示它并单击删除选定按钮 删除已选择 - 单击此按钮可从“类型详细信息”列表中删除当前选择的分类器

	<ul style="list-style-type: none"> • 关闭 - 单击此按钮可关闭“XSD 联合成员”对话框并在 成员类型”字段中列出所选元素和数据类型
注解	(可选) 类型任何注记你需要的这个元素。
确定	单击此按钮保存输入的属性数据并关闭 XSD 联合“属性”对话框。
取消	单击此按钮可丢弃输入的属性数据并关闭 XSD 联合“属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您正在编辑现有的 «XSDunion»元素信息时，会显示此按钮。 单击按钮打开 «XSDunion»元素的UML元素“属性”对话框。

注记

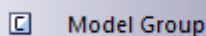
- 当您单击 属性联合元素”对话框上的关闭按钮时，将添加一个从概括联合到同一图表上的每个成员元素的图表；任何不在同一张图上的元素都列在 XSD 联合元素的右上角
- 如果未列出与元素不同的成员类型，请选择 开始>外观>首选项>首选项>图表>行为”并选中 显示隐藏的父项”复选框
- A联合元素：
 - 不能包含任何子类
 - 不能包含任何 «XSDelement» 或 «XSDattribute» 原型UML属性
 - 不能是源连接器的关联
 - 可以是复杂类型元素的关联连接器的目标
 - 可以是简单类型元素中的概括器的目标

集团模型

您可以通过从XML Schema工具箱中拖动模型组图标并将其直接放到图表上来创建 «XSDsequence»、«XSDchoice» 或 «XSDall» 原型类。

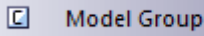
«XSDsequence»模型组（默认模型组类型）是类拥有的属性和关联的容器。模型组依次添加到类各自所有者的模型组中。类所有者指定的标记值一直持续到模型组的子元素；如果 memberNames 对于 complexType 不合格，则此模型组的子代在添加到该 complexType 时也是如此。

工具箱图标



访问

要为选定的 «XSDsequence»、«XSDchoice» 或 «XSDall» 原型类显示 “XSD模型组属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «XSDsequence»、«XSDchoice» 或 «XSDall»属性类
键盘快捷键	Alt+Enter
其它	<ul style="list-style-type: none"> • 双击 «XSDsequence»、«XSDchoice» 或 «XSDall» 原型类，或 • 将  图标从工具箱拖到图表上（这将创建一个新的模型组元素；您可以从 «XSDsequence»、«XSDchoice» 或 «XSDall» 构造型中进行选择，其中 «XSDsequence» 是默认值）

定义属性

字段/按钮	行动
名称	如果您不想使用模型组的默认名称，请用其他名称改写它。
集团模型	单击下拉箭头并选择定义该组的子元素应如何出现在复杂类型类中的值： <ul style="list-style-type: none"> • 序列指定子元素必须以指定的顺序出现；创建一个 «XSDsequence» 原型类 • 选择 - 指定只能出现一个子元素；创建一个 «XSDchoice» 原型类 • all - 指定子元素可以以任何顺序出现；创建一个 «XSDall» 原型类
MinOccurs	（可选）类型该元素必须在类中出现的最小次数。 类型0 表示该元素是可选的。 默认值为1。

最大发生次数	(可选) 类型该元素可以在类中出现的最大次数。 类型无界表示元素可以出现的次数没有限制。 默认值为1。
注解	(可选) 类型任何注记你需要的这个元素。
确定	单击此按钮保存输入的模型组数据并关闭 XSD元素的“属性”对话框。
取消	单击此按钮可丢弃输入的模型组数据并关闭 XSD元素的“属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	编辑现有模型组元素信息时会显示此按钮。 单击 按钮打开模型组类的UML元素“属性”对话框。

注记

- A模型组：
- 只能包含 «XSDelement» 构造型UML属性
- 可以包含复杂类型和简单类型作为子元素
- 可以是复杂类型、简单类型、组和模型组源的关联连接器的来源
- 必须是来自复杂类型的至少一个传入关联连接器的目标
- 不能是概括或连接的源或目标

枚举


枚举定义了类的可接受值的列表。您可以通过将 Enum 图标从XML Schema工具箱中拖放到图表上来创建枚举元素。

工具箱图标




访问

要为选定的 枚举“原型元素显示 XSD 枚举属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击«枚举»定型元素 属性
键盘快捷键	Alt+Enter
其它	双击 枚举“定型元素，或 将  Enum 图标从工具箱中拖放到图表上（这将创建一个新的 枚举“原型元素”）

定义属性

字段/按钮	行动
名称	如果您不想使用枚举的默认名称，请用另一个名称改写它。
类型	任何一个： <ul style="list-style-type: none"> 类型数据类型的名称，或 单击下拉箭头并从列表中选择 XSD 内置数据类型，或 单击  按钮并浏览现有的 XSDsimpleType元素
价值观	类型所选类型的每个值，以逗号分隔。 这些值列在作为属性的元素上。
注解	（可选）类型任何注记你需要的这个元素。
确定	单击此按钮保存输入的枚举元素数据并关闭 XSD 枚举的 属性”对话框。
取消	单击此按钮可丢弃输入的枚举元素数据并关闭 XSD 枚举 属性”对话框。

帮助	单击此按钮可显示此帮助主题。
UML	当您正在编辑现有的枚举元素信息时，会显示此按钮。 单击按钮打开枚举类的UML元素“属性”对话框。

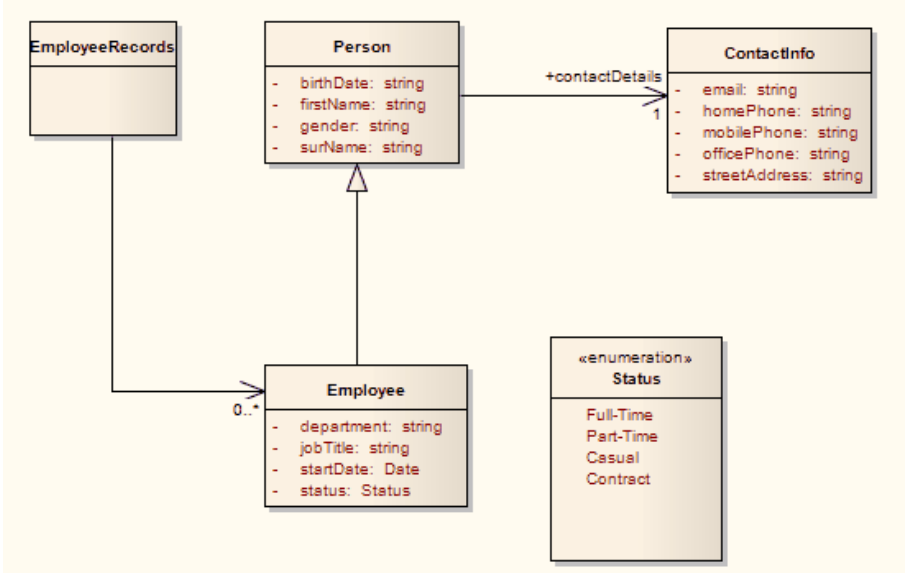
注记

- 一个枚举：
- 不能包含任何 «XSDelement» 或 «XSDattribute» 原型UML属性
- 不能包含任何子类
- 不能是源连接器的关联
- 可以成为一个概括器的目标
- 最多可以有一个概括类型的连接类

来自抽象类模型的 XML

您可以仅使用简单的抽象类模型来建模 XML 模式。例如，这使得架构师可以开始在更高的抽象级别上工作，而无需关心架构的实现细节。虽然这种抽象模型随后可以使用工具箱的“XML Schema”页面进行工具箱，但也可以直接由Enterprise Architect的架构生成器生成，在这种情况下，架构生成器应用一组默认映射来转换将抽象模型转换为 XSD 文件。

示例

结构	细节
图表	<p>这是早期员工详细信息示例模型的简单类元素版本。它不使用 XSD 特定的刻板印象或标记值。</p>  <pre> classDiagram class EmployeeRecords class Person { - birthDate: string - firstName: string - gender: string - surName: string } class Employee { - department: string - jobTitle: string - startDate: Date - status: Status } class ContactInfo { - email: string - homePhone: string - mobilePhone: string - officePhone: string - streetAddress: string } class Status { <<enumeration>> Full-Time Part-Time Casual Contract } EmployeeRecords --> Employee : 0..* Employee < -- Person Person --> ContactInfo : +contactDetails 1 </pre>
架构	<p>该模式片段可以从示例模型中生成：</p> <pre> <?xml 版本=" 1 .0"?> <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"> <xs:simpleType name="状态"> <xs:restriction base="xs: string "> <xs:enumeration value="全职"/> <xs:enumeration value="兼职"/> <xs:enumeration value="休闲"/> <xs:enumeration value="合约"/> </xs:限制> </xs:simpleType> <xs:元素名称=" Person " type=" Person "/> <xs:complexType name=" Person "> <xs:序列> </pre>

```
<xs:元素名="firstName" type="xs: string" />
<xs:元素名="surName" type="xs: string" />
<xs:元素名="birthDate" type="xs: string" />
<xs:元素名="gender" type="xs: string" />
<xs:元素="contactDetails" type="ContactInfo"/>
</xs:sequence>
</xs:complexType>
<xs:元素名称="Employee" type="Employee"/>
<xs:complexType name="员工">
<xs:复杂内容>
<xs:extension base=" Person ">
<xs:序列>
<xs:元素名称="状态" type="状态"/>
<xs:元素名="jobTitle" type="xs: string" />
<xs:元素名称="startDate" type="xs:date"/>
<xs:元素名="department" type="xs: string" />
</xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:元素名称="EmployeeRecords" type="EmployeeRecords"/>
<xs:complexType name="EmployeeRecords">
<xs:序列>
<xs:元素="Employee" type="Employee" minOccurs="0"
maxOccurs="unbounded"/>
</xs:sequence>
</xs:complexType>
<xs:元素名称="ContactInfo" type="ContactInfo"/>
<xs:complexType name="ContactInfo">
<xs:序列>
<xs:元素名="homePhone" type="xs: string" />
<xs:元素名="mobilePhone" type="xs: string" />
<xs:元素名="officePhone" type="xs: string" />
<xs:元素名="email" type="xs: string" />
<xs:元素名="streetAddress" type="xs: string" />
</xs:sequence>
</xs:complexType>
</xs:schema>
```

默认UML到 XSD 映射

当您使用抽象类模型定义简单模式时，Enterprise Architect架构生成器使用UML到 XSD 构造的默认映射将UML信息转换为 XSD。架构生成器也使用这些默认值在抽象模型中生成未定型的元素。

当您使用工具箱的“XML Schema”页面的模型XML Schema图表，工具箱元素的构造型和标记值将覆盖默认映射。

结构体

UML构建	默认 XSD 生产规则
包	<p>A目标包生成架构元素。如果目标包中包含另一个包中的Classes，并且设置了标记值targetNamespace 和targetNamespacePrefix，则这些都作为架构元素的属性包含在内。</p> <p>此外，为每个引用的包创建一个导入或包含元素：</p> <ul style="list-style-type: none"> 如果外部包与目标包共享相同的targetNamespace标记值，则使用包含元素值 在 targetNamespaces 不同的地方使用导入元素
类	<p>生成根级元素声明和 complexType A。元素名和类型与类名相同。还会生成一个 XSD序列模型组，以包含作为元素生成的UML属性。</p>
属性	<p>为每个类属性声明一个元素。元素名称设置为UML属性名称。这以类名为前缀，以使元素唯一。minOccurs 和 maxOccurs 属性设置为反映属性基数。</p> <p>如果属性引用另一个类，则元素声明后跟一个 complexType 定义，其中包含对适当 complexType 的引用。</p>
关联	<p>为类所拥有的每个关联声明一个元素。元素名称设置为关联角色的名称。minOccurs 和 maxOccurs 属性反映了关联的基数。</p>
(概括继承	<p>对于单一继承，会生成一个扩展元素，并将基本属性设置为基本类名。然后将子类的UML属性附加到扩展元素内的 XSDall模型组。</p>
枚举	<p>为枚举声明A simpleType元素，并将 name 属性设置为枚举名称。A Restriction元素是在 base 设置为string的情况下生成的。每个枚举属性都附加到限制元素作为 XSD枚举元素，其值设置为UML属性名称。模式生成器会忽略UML属性的任何类型规范。</p>

注记

- 如果未指定，则 minOccurs 和 maxOccurs 属性默认为1
- 如果未指定关联的方向，则假定所有者为源

生成XSD

当您开发XML Schema模型时，无论是作为抽象类模型还是定制的 XSD类模型，您都可以使用生成XML Schema特征将其转发工程为 W3C XML Schema (XSD) 文件。由于XML Schema对应于Enterprise Architect中的UML包，因此XML Schema生成是包级别的操作。

您可以在模型的架构包元素中定义要生成XML Schema的文件的位置。

访问

功能区	开发>架构建模>导出XSD
-----	---------------

生成架构文件

选项	行动
编码	任何一个： <ul style="list-style-type: none"> 单击下拉箭头并选择要使用的 XML 编码方案，或者 单击默认按钮以应用默认编码方案 (UTF-8)
生成所有全局 ComplexTypes 的全局元素 (“伊甸园”风格)	默认选择生成伊甸园风格的架构，包含全局元素。 如果要省略全局元素，请清除复选框。
为引用包生成XSD	选中该复选框可为在此对话框中选择的任何包引用的包生成架构。
缺少文件名时提示	如果缺少生成架构文件的路径，请选中复选框以在生成架构期间提示引用包的文件名。 如果未选择 “为引用包生成XSD” 选项，则此选项不可用。
使用相对路径来引用 XSD (如果 'schemaLocation' 标记为空)	选中复选框以在引用外部包时在 XSD 导入 (或 XSD 包含) 语句中使用相对路径，前提是被引用包上的 schemaLocation 标记为空。 您在 “XSD架构属性” 对话框 (架构元素的元素 属性”对话框) 上为引用和引用的属性原型包设置了 “架构文件” 字段，XSDschema正确确定相对路径。
子包生成XSD	选中复选框为所选包的子包生成架构，然后选择： <ul style="list-style-type: none"> 包括所有包 - 在列表框中列出父包下的所有子包，或 包括 <XSDschema> 包 - 仅列出那些具有构造型 «XSDschema» 的包 对于每个包，列表框显示包名和可以生成模式文件的文件路径 (如果已设置)。 要更改包的文件路径，请双击列表框中的条目，然后在提示字段中输入或浏览新文件路径。 如果包已经设置了文件路径，则默认选中其复选框，以生成 XSD 模式；如果您不想从该包生成 XSD 架构，您可以取消选中该复选框。 如果您对没有设置文件路径的包选择复选框，则会自动显示文件路径的提示。

生成	单击此按钮可为列表框中选定的每个包生成架构。
关	单击此按钮关闭对话框，而不保存您的选项选择。
视图架构	单击此按钮可查看为列表框中突出显示的包生成的架构。
进步	选择架构生成的进度。

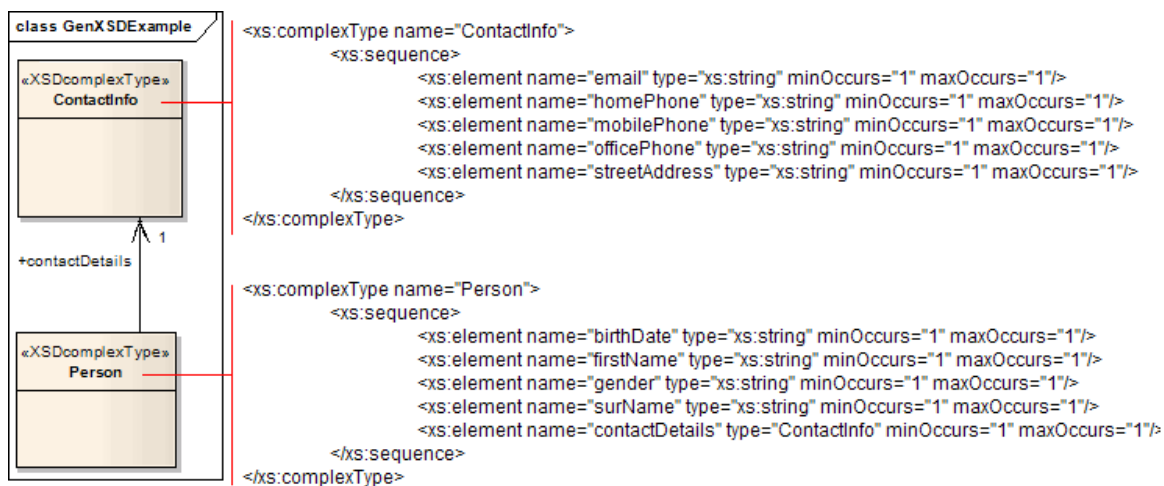
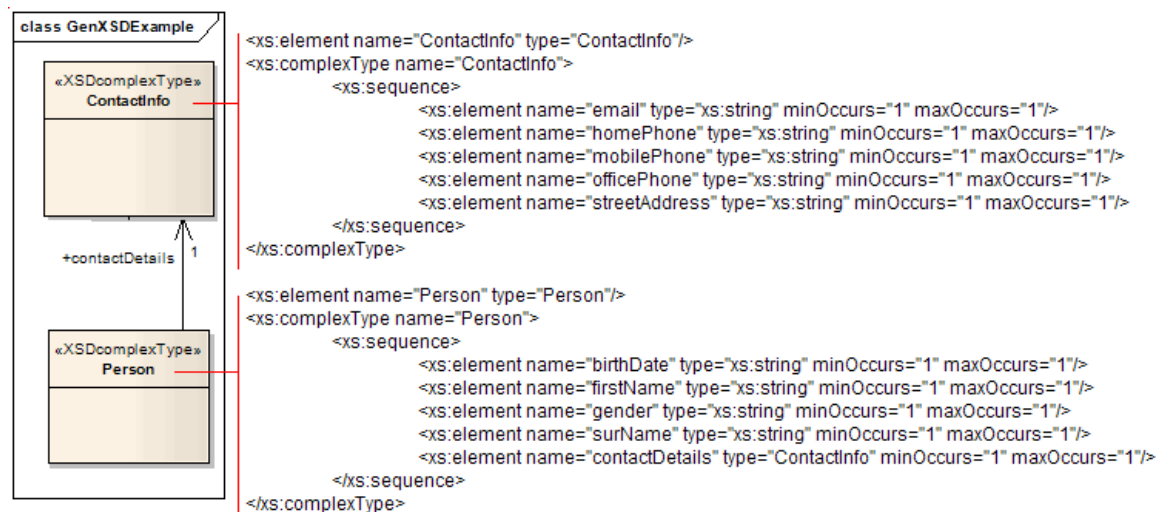
生成全局元素

默认情况下，Enterprise Architect以伊甸园风格生成XML Schema。对于每个全局 XSDcomplexType 原型类，系统都会生成一个全局元素。

示例

您可以通过取消选中 生成XML Schema “对话框上的 生成全局元素for all global ComplexTypes”复选框来更改指定的默认行为。然后，生成的 XSD 不再包含全局元素；也就是说，它不再有以下行：

- `<xs:element name="ContactInfo" type="ContactInfo"/>` 和
- `<xs:element name="Person" type="Person"/>`



导入XSD

要对 W3C XML Schema (XSD) 文件进行逆向工程以创建或覆盖UML类模型的包，您可以使用XML Schema导入功能。

访问

功能区	开发>架构建模>导入XSD
-----	---------------

导入架构文件

选项	行动
包	显示选定目标包的名称。
目录	类型在或浏览 (单击 ) 包含源XSD 文件的目录。
选定的文件	<p>列出当前可用于导入的XML Schema。</p> <ul style="list-style-type: none"> • 要选择单个文件，请单击它 • 选择多个单独的文件 Ctrl+单击每个文件 • 要选择文件范围，请按 Shift 并选择范围中的第一个和最后一个文件
以“类型”后缀导入全局元素	选中此复选框可将全局元素和它所指的 ComplexType 视为两个单独的实体。
导入引用的XML Schema (s)	选中此复选框以导入在“选定文件”字段中选择的任何文件所引用的任何XML Schema。
图表XML Schema (s)	选中此复选框以在每个导入XSDschema包下创建一个类图。
导入XSD元素/属性为	<p>选择适当的单选按钮以指示如何将内联 XSD 元素和 XSD 属性导入到类中，可以是：</p> <ul style="list-style-type: none"> • UML关联 • UML属性
导入	单击此按钮开始 XSD 导入。
关	单击此按钮关闭对话框，而不保存您的选项选择。
进步	<p>显示指示架构导入进度的系统消息。</p> <p>在包含大量外部引用的导入中，捕获进度消息以准确检查已导入的内容可能很有用。为此，请右键单击消息并：</p> <ul style="list-style-type: none"> • 将所选消息复制到剪贴板 (选择 将所选消息复制到剪贴板“菜单选项) • 将所有消息复制到剪贴板 (选择 全部复制到剪贴板“菜单选项) ， 或

- | | |
|--|--|
| | <ul style="list-style-type: none">• 将所有消息保存到文件 (选择 “保存到文件” 菜单选项) |
|--|--|

注记

- 如果模型中已经存在正在导入的XML Schema文件， Enterprise Architect将跳过导入文件
- 对 XSD原始类型的引用总是作为UML属性导入
- 外部架构文件中对 XSD 结构的引用始终作为UML属性导入
- Enterprise Architect使用XML Schema的 XSD导入和 XSD 包含元素中的 schemaLocation 属性来确定文件之间的依赖关系；此属性必须设置为有效的文件路径（而不是 URL），才能正确导入依赖的XML Schema

全局元素和复杂类型

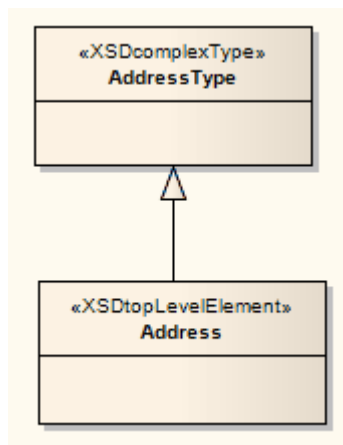
一些 XML Schema 具有与引用全局元素同名的 ComplexType 元素，但带有后缀“类型”，如下所示：

```
<xs:元素="Address" type="AddressType"/>
<xs:complexType name="AddressType">
<xs:序列/>
</xs:complexType>
```

默认情况下，在 XSD 导入时，Enterprise Architect 将此全局元素及其边界 ComplexType 视为单个实体，并创建一个与全局元素同名的单个 XSDcomplexType 原型类，如下所示：



您可以通过选择“导入XML Schema”对话框上的“使用类型”后缀的“导入全局元素”复选框来更改此默认行为。当您选择此选项时，系统会将全局元素和它所指的 ComplexType 视为两个单独的实体。例如，系统为全局元素创建一个«XSDtopLevelElement»原型类，为 ComplexType 创建一个«XSDcomplexType»原型类，连接如下所示：



注记

- Enterprise Architect 将这两个定义视为单独的实体，无论是否选中“导入带有类型”后缀的全局元素”复选框：

```
<xs:元素="HomeAddress" type="AddressType"/>
<xs:complexType name="AddressType">
<xs:序列/>
</xs:complexType>
```

XSL 转换

模型、作者和使用 XML 文档执行 XSLT 转换和样式表

Enterprise Architect提供建模和执行 XSL 转换的功能。XSLT 是一种可用于将 XML 输入文档转换为其他类型文档的技术。样式表是用于转换内容的 XSL 组件。功能包括：

- 用于对 XSLT 转换进行建模的专用图表和工具箱
- 用于样式表创作、调试和执行的专业编辑器。
- XML 文档验证
- XML Schema验证

模型使用 XML 转换图进行转换。在此图中，您可以创建 xml 文档和样式表，将它们链接到转换 (活动)，然后执行或调试转换。XSL Transform模型和 XML 文件的输入是 XSLT 和 XML文件，可以从工件工具箱中选择。这些工件通常通过将输出和 xsl 文件拖动到图表来创建。输出格式通常使用适当的输出变换来描述。转换的进度和成功/失败显示在系统输出窗口的“XSLT”选项卡上。

创建 XML 转换图表

节	行动
1	在浏览器窗口中，右键单击相应的包并选择“添加图表”选项。

工件元素在 XML变换工具箱

工件	描述
XML 转换	转换的模型参考，提供输入和可选输出。用于运行或调试转换。 输入：XML文件、XSLT 输出：输出 (工件)
2	在“新图表”对话框中，在“名称”字段中输入适当的图表名称 (如果需要)，然后单击“选择来源”列表中的“扩展”和“图表类型”列表中的“XML 转换”。点击确定按钮。 新图表打开，显示“图表变换”页面的图形工具箱。
XSLT	标识要执行的样式表。 输入A N 输出A N
XML文件	标识要转换的输入文档。 输入A N 输出A N
XSD	标识可用于 (可选) 自动对输出文档执行 XML 验证的模式。 输入：输出工件，XML文件，(或可选)

	输出A N
输出工件	使用此工件操作来定义输出。当工件输出由转换创建时使用的文件路径。要选择名称输出文件，工件文件或文件以显示其属性并在“文件”选项卡下输入路径。使用工件，从元素中画出一个跟踪连接器。

手动验证文档

使用Enterprise Architect，您可以对要转换的文档和 XSLT 样式表执行 XML 验证。

要运行验证，请右键单击 XSL 调试器中的 XML 文档或样式表，然后选择“XML 验证”。将显示A提示以确认您是根据文档类型定义还是XML Schema进行验证。

- 对于文档类型定义，只需单击确定按钮；验证进行
- 对于 XML 模式，选择适当的单选按钮来确定验证语法是在文档中还是在其他地方定义；如果在其他地方，请输入语法的名称空间和 URL 或文件路径

如果在调试运行过程中发现错误，将输出到调试窗口（按Alt+8显示此窗口）。

如果在正常验证运行过程中发现错误，将输出到系统输出窗口的“XSL”选项卡（如果该窗口没有自动显示，请按Alt+1并选择“系统输出”）。要查找文档中的错误，请双击错误消息。

XSLT 处理器和版本

这些特征中使用的 XSL 处理器是从[Apache Xalan 项目](#)（C++ 版本1.11）构建的

XSL变换模型

当您使用 XSL变换模型时，您可以绘制文件系统中已经存在的文件，也可以在模型元素中创建样式表和源的内容。

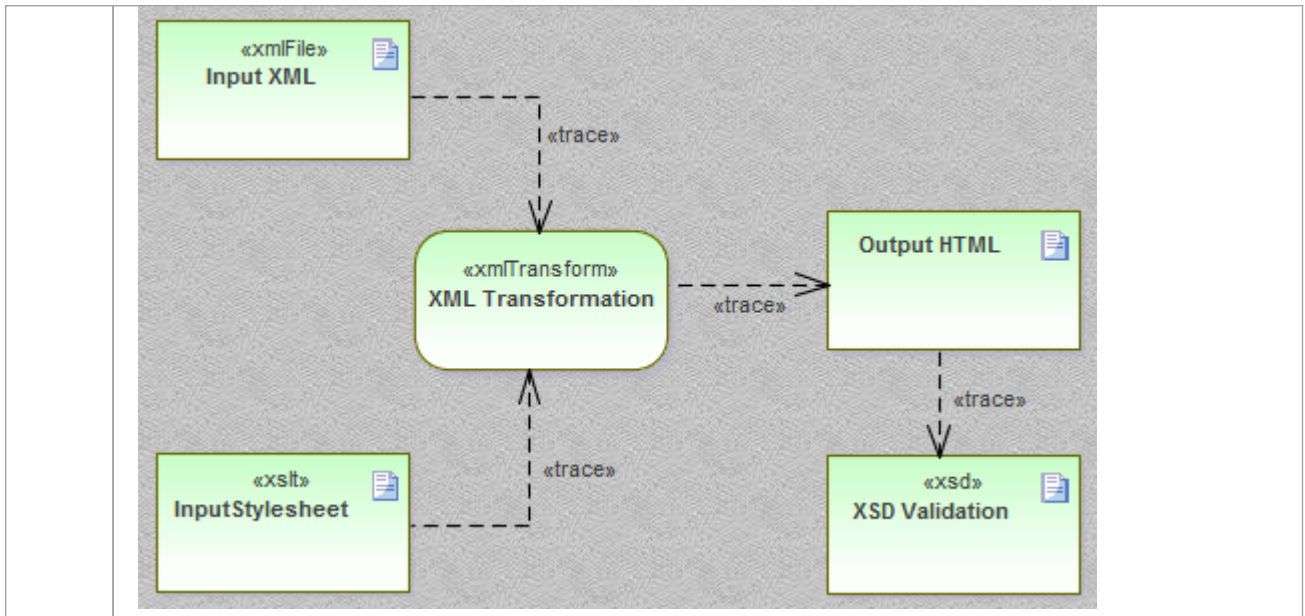
现有文件中的模型元素

这是建模转换的最简单和最常用的方法。当您把文件拖到 XML 转换图上时，元素工件然后，您可以使用跟踪连接器使用快速链接器将文件工件作为输入链接到元素XML 转换。

或者，您可以：

- 指定一个输出位置（文件），方法是从 XML 文件或输出链接工件到一个替代位置（从 XML 工件为跟踪连接器）
- 通过拖动 XSD 模式文件并将工件元素连接到 XML 转换元素上的任何输出结果来验证文档

节	行动
1	打开文件浏览器和 XML 转换图。
2	在文件浏览器中，单击输入文件并将其拖到 XML 转换图上。 将显示A提示以将文件另存为： <ul style="list-style-type: none"> • 工件，将 XML 文件工件作为文件系统中的文件的快捷方式提供服务 • 工件，将文件内容读取到 XML 文件工件中并存储在模型文件中的位置；您将选择此选项以使源文件内容对模型的其他用户可用
3	选择“外部工件”选项。 为输入文件生成一个 XML 文件工件元素。
4	在文件浏览器中，单击 XSL 样式表文件并将其拖到 XML 转换图上。 作为对外部提示的回应，选择“外部工件”选项。 为样式表文件生成一个 XSLT 元素。
5	将 XML 转换图标从工具箱拖到图表上，以创建 XML 转换活动元素。如果你愿意，可以给这个元素一个新名字。
6	从“公共”工具箱页面拖动跟踪图标，在以下关系之间创建： <ul style="list-style-type: none"> • 输入 XML 文件元素及转换活动元素 • XSLT 样式表文件元素和转换活动元素
7	（可选的。） <ul style="list-style-type: none"> • 如果您想在文件中捕获输出，在文件浏览器中找到相应的文件并将其拖到图表上以生成另一个文件工件到该图表上；使用跟踪连接器将其链接到 XML 转换元素 • 如果要验证输出文档，请在文件浏览器中找到 XSD 模式文件并将其拖到图表上以生成 XSD 元素；将此工件到变换元素的输出文件工件（或任何输出）
8	按 Ctrl+S 保存图表内容。 如果输出是 HTML，您的图表可能类似于以下内容：



从零开始建模元素

当您使用“XML变换”工具箱创建 XSLT 文件元素时，系统会将这些文件存储为模型文档。双击 XML 转换图上的元素，在 Enterprise Architect 的 XSLT 调试器中打开模型文档，您可以在其中编写和编辑文件内容。保存文档后，内容将保存回模型中。

否则，建模转换的过程与从现有文件建模元素中描述的过程相同。

使用 XML 编辑器编辑文档

Enterprise Architect 提供了一个强大且有用的 XML 编辑器，具有许多特征，包括：

- 智能感知
- 提供快速替代导航的上下文结构树（提示：按 Ctrl+1 切换文档树视图）
- XSL 和 XSD 文档元素的自定义图标
- 代码完成和
- 验证文档和引用的模式

当在 Enterprise Architect 中打开任何带有 XML 声明的文档时，XML 编辑器将打开。（或者，按 Ctrl+Shift+O。）XSLT 调试器并排使用两个 XML 编辑器来显示样式表和正在转换的文档。

执行 XSL 变换

在对 XSL 变换建模后，您可以直接从模型图中执行它。您还可以直接从 XSL 样式表和输入文件执行转换。

执行图表the变换

节	行动
1	<p>在 XML 变换图上，右键单击 XML 变换活动元素并选择 运行XSL变换”选项。</p> <p>将显示 XSLT调试器视图，其中显示了转换中使用的样式表 (.xsl) 文件和 XML 文档。</p> <p>系统输出窗口也会显示，在“XSL”选项卡中显示错误或成功消息。（如果系统输出窗口不显示，请按 Ctrl+Shift+8。）</p> <p>如果您设置了输出验证，系统输出窗口也会显示验证注释。</p>
2	<p>如果您已通过输出或文件工件将输出定向到文件，请按 F12 以查看输出。</p>

调试一个 XSL 变换

当您使用 XSLT 调试器运行转换时，您可以使用 Enterprise Architect 的调试器结合断点来控制过程并检查转换的状态。XSLT 调试器提供运行按钮和各种节按钮。您可以通过单击样式表的左边距来设置断点。

当一个步骤完成或遇到上下文时，可以在本地窗口窗口（执行>窗口>局部变量”）中查看转换的时间 - 包括模板调用的任何参数。也可以通过显示窗口(“调用堆栈>>调用堆栈)查看转换的当前状态是如何到达的。

调试变换

节	行动
1	<p>在 XML 变换图上，右键单击 XML 变换活动元素并选择 “调试XSL变换”选项。</p> <p>XSLT调试器视图显示，其中显示了自动启动的转换中使用的样式表 (.xsl) 文件和 XML 文档。样式表中当前正在执行的语句被突出显示。</p> <p>视图顶部是一个调试器工具栏，提供开始、暂停、节结束、节输入、节输出和停止调试过程的常规调试选项。工具栏中的最后一个图标提供了在视图中隐藏或显示 “.xml源”选项卡的选项。您可以使用这些按钮重复和控制调试过程。</p> <p>系统输出窗口还会显示 “XSLT变换”选项卡上的调试器进度消息。（如果系统输出窗口没有显示，请按 Alt+1 并选择 “系统输出”。）</p> <p>错误消息被定向到调试窗口（按 Alt+8）。您还可以使用调试窗口工具栏按钮和选项来控制 XSL 的变换。</p>
2	<p>如有必要，选择显示本地窗口和调用堆栈窗口。</p> <p>单击 XSLT调试器样式表面板的左边距并设置您要用于检查处理的任何断点。</p>
3	<p>再次运行调试器检查系统输出窗口、调用堆栈调试器和本地窗口以及您要应用的任何其他执行分析工具所指示的执行情况。</p>

XML 验证

Enterprise Architect提供 XML 文档的验证。可以根据 XML 模式或数据类型定义 (DTD) 验证文档。验证是在 Enterprise Architect编辑器中使用其上下文菜单执行的。通常，XML 文档将包含与其所遵循的模式相关的信息。但是，您可以选择覆盖它，根据任何模式验证文档，无论是在本地计算机上的路径上还是在 URL 上。此示例演示了如何将特征用于包含不正确属性的文档。

访问

上下文菜单	可从显示上下文内容的任何编辑器窗口的时间菜单访问。 在编辑器窗口中右键单击并选择“XML 验证”
-------	---

XML 文档验证

节	行动
1	打开要验证的 XML 文档。
2	使用编辑上下文菜单并选择“XML Validation”选项。
3	从可用选项中选择选择的语法： <ul style="list-style-type: none"> XML Schema (默认) 数据类型定义
4	选择架构位置。默认选择“在文档中定义”。XML 文档通常会指定管理其内容的模式。要选择与文档中定义的架构不同的架构，请选择“外部”并提供 URL 或文件路径。例子： <ul style="list-style-type: none"> http://mydomain/myschema.xsd c:\mydomain\myschema.xsd
5	点击确定。验证的输出将显示在系统输出窗口的“XML 验证”选项卡中。

XML 文档验证示例

```

1 <?xml version="1.0" standalone="no"?>
2 <portfolio xmlns:xsi="http://www.w3.org/200
3   xsi:noNamespaceSchemaLocation="portfolio
4     <stock exchange="nyss">
5       <name>zacx corp</name>
6       <symbol>ZCXM</symbol>
7       <price>28.875</price>
8     </stock>
9     <stock exchange="nasdaq">
10      <name>zaffymat inc</name>
11      <symbol>ZFFX</symbol>
12      <price>92.250</price>
13    </stock>
14    <stock exchange="nasdaq">
15      <name>zysmergy inc</name>
16      <symbol>ZYSZ</symbol>
17      <price>20.313</price>
18    </stock>
19  </portfolio>
20

```

图1：具有无效属性值“nyss”的 XML 文档

在此示例中，文档描述了具有无效交换代码“nyss”的库存项目。从这个模 可以看出，“exchange”属性的唯一有效值是“nyse”、“nasdaq”或“ftsi”。

```

1 <?xml version="1.0"?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema" el
3   <xs:element name="portfolio">
4     <xs:complexType>
5       <xs:sequence>
6         <xs:element name="stock" minOccurs="1" ma
7           <xs:complexType>
8             <xs:sequence>
9               <xs:element name="name" type=
10              <xs:element name="symbol" typ
11              <xs:element name="price" type
12            </xs:sequence>
13            <xs:attribute name="exchange" typ
14          </xs:complexType>
15        </xs:element>
16      </xs:sequence>
17    </xs:complexType>
18  </xs:element>
19  <xs:simpleType name="category">
20    <xs:restriction base="xs:string">
21      <xs:enumeration value="nyse"/>
22      <xs:enumeration value="nasdaq"/>
23      <xs:enumeration value="ftsi"/>
24    </xs:restriction>
25  </xs:simpleType>
26 </xs:schema>

```

图2：描述允许的证券交易所代码的XML Schema

此图像显示了验证中使用的架构。在这里可以看到“投资组合”元素的声明由一个或多个“股票”元素组成。反过来，每个股票元素都需要一个“交易所”属性来命名相关证券交易所的代码。

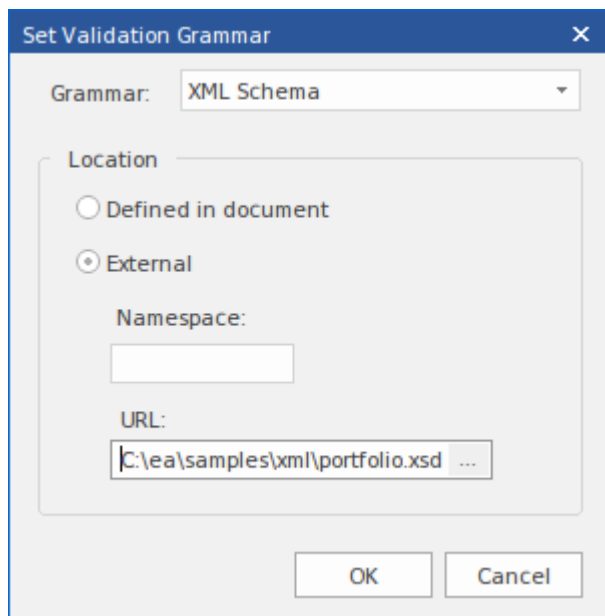


图 3：命名本地模式文件的“XML 验证”对话框

这是“XML 验证”对话框。它可以从 Enterprise Architect 中包含 XML 内容的任何编辑器的上下文菜单访问。您可以在此处选择要在验证中使用的架构。在示例中，处理器将使用本地模式文件验证文档。这恰好是文档命名的相同模式，但它可以是任何模式（例如模式的开发或更高版本）。

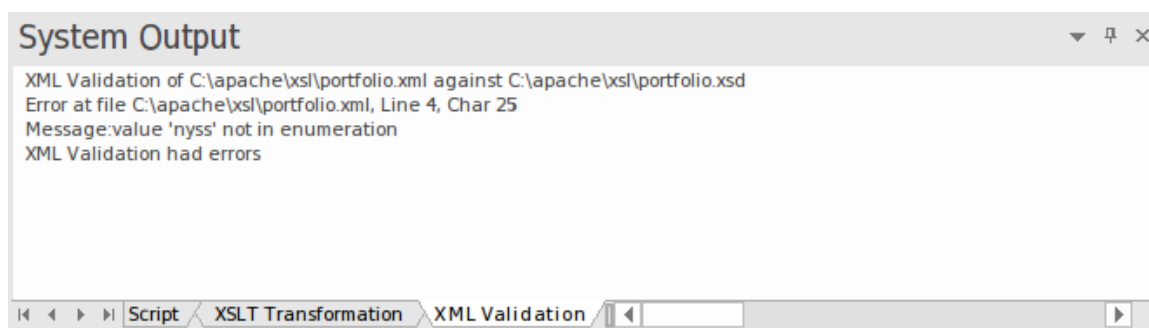


图 4：显示验证错误的系统输出窗口

此图像显示了验证的结果。根据模式描述的枚举，属性值“nyss”已被识别为不正确。双击错误将在编辑器中显示可以轻松更正的代码行。

面向 XML 服务架构

最大化敏捷性、规模和互操作性模型性的组织、社区和系统

面向服务架构(SOA) 是一种架构范式，用于定义人员、组织和系统如何提供和使用服务以实现结果。

服务是A定义明确的界面向另一个人提供价值，该界面可供社区（可能是一般公众）使用。A服务导致工作由另一个人提供。

面向服务的架构(SOA) 是一种组织和理解（表示）组织、社区和系统的方式，以最大限度地提高敏捷性、规模和互操作性。SOA 方法很简单——人员、组织和系统相互提供服务。这些服务使我们能够在不自己动手甚至不知道如何做完成某事——使我们更加高效和敏捷。服务还使我们能够向他人提供我们的能力以换取一些价值——从而建立一个社区、流程或市场。SOA 范式同样适用于集成现有功能以及创建和集成新功能。

（源自面向服务的架构建模语言 (SoaML) - UML配置文件和服务元模型 (UPMS) 规范 (OMG 文档 ad/2008-11-01) ；第 25-26 页。）

在Enterprise Architect中建模和开发完成面向服务的架构时，您可以使用以下任何或全部：

- XML Schema Definition (XSD)，也称为XML Schema一种 XML 技术，用于指定 XML 文档必须遵守的规则；Enterprise Architect提供了Schema 编辑器界面，帮助您模型和生成 XML 模式
- XSL 转换使用 XSL 样式表将输入文档转换为 XML 或其他类型的文档，为此您可以使用 XSLT 编辑器和调试器来建模和执行转换
- 网络服务描述语言 1.1 (WSDL) - 一种基于 XML 的关键语言，用于描述 Web 服务
- 面向服务的架构建模语言 (SoaML) - 使用Unified Modeling Language (UML) 设计和建模 SOA 解决方案的标准方法
- 面向服务的建模框架 (SOMF) - 一种面向服务的开发生命周期方法论，提供了许多有助于成功的面向服务的生命周期管理和建模的建模实践和学科
- 国家信息交换NIEM建模- 一个通用框架，用于定义如何在系统、政府机构和部门之间共享信息
- 元对象功能(MOF) - 一种物件管理组 (OMG) 标准，作为一种元建模架构来定义UML，因此提供了一种定义语言或数据的结构或抽象语法的方法

WSDL

网络服务描述语言1.1 (WSDL) 是一种关键的基于 XML 的万维网络联盟 (W3C) 语言，用于描述 Web 服务。WSDL 支持对于开发完整的完成服务的架构(SOA) 至关重要，UML 2.5 和 XML 的耦合为在组织内指定、构造和部署基于 XML 的 SOA 工件提供了自然机制。

使用Enterprise Architect ，您可以快速模型、正向工程和反向工程 WSDL 文件。

WSDL 1。 1模型结构

规范1下A网络服务描述语言 (WSDL)。 1，定义在一个«WSDLnamespace»原型包中，它表示WSDL 元素的顶级容器。从概念上讲，它映射到 WSDL 定义元素中的 targetNamespace。

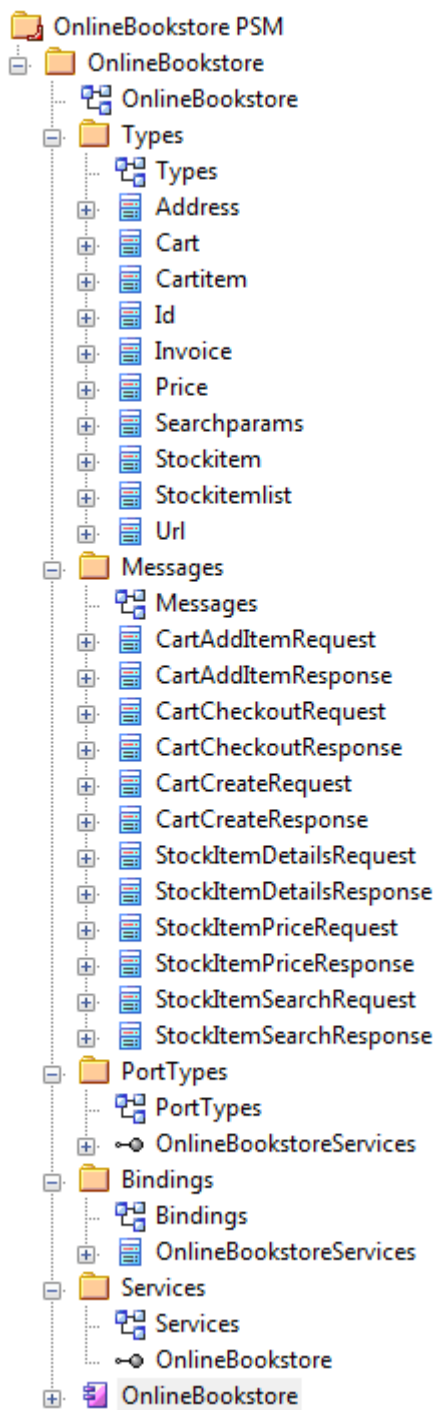
当您创建 WSDL模型时， Enterprise Architect会创建命名空间并提供一组子包，每个子包都包含一个用于定义模型的组成元素的图表，以及用于在子包之间导航的概览图。您通过序列中的子包来定义后面的对象使用的对象，这些对象本身被调用到更后面的对象中。

WSDL 结构开发

WSDL元素类型	描述
类型	在 XSD架构中定义，这些是 Web 服务使用并由 WSDL 消息通信的 XSD 数据类型；您将 «XSDelement»、«XSDsimpleType» 和 «图表» 原型元素从图形工具箱的“ XML Schema ”页面拖到类型图上。
留言	WSDL 消息标识 Web 服务正在传送的数据。每个信息元素都包含一个或多个信息部分，这些信息部分是属性，每个都标识正在通信的 XSD 数据类型。
端口类型	WSDL端口类型是 Web 服务的基本核心，定义了服务的接口。每个端口类型操作由一组端口类型组成，每个端口类型都将消息的交换（数据输入到接口的输出和从接口输出的数据标识为该操作）。端口类型操作还可以识别作为故障指示器的消息。
绑定	捆绑指定为特定端口类型定义的操作和消息A协议和数据格式。每个«binding»类实现（实现）«接口»接口指定的操作——端口类型元素中的端口类型操作会自动复制到捆绑元素中作为捆绑操作。
服务	WSDL 服务定义了 Web 服务A正式接口。它描述了暴露特定捆绑的端口类型的集合，每个暴露的捆绑都有一个关联。因此，它封装了模型中定义的一组其他数据结构——如果不是所有数据结构的话。
文件	WSDL 文档由具有构造型 «WSDL» 的组件表示。这是生成 WSDL 文件的元素。 您可以创建多个文档以跨多个物理 WSDL 文档（无论是在配置中还是在不同的配置中）重复使用一个命名空间的模式类型、消息、端口类型、绑定和服务。

示例

此图显示了一个示例 WSDL 名称空间 OnlineBookstore PSM，它包括一个 WSDL 文档 OnlineBookstore（位于层次结构的底部）。



注记

- 您还可以使用 WSDL 模型变换从 UML 接口生成包结构

WSDL模型

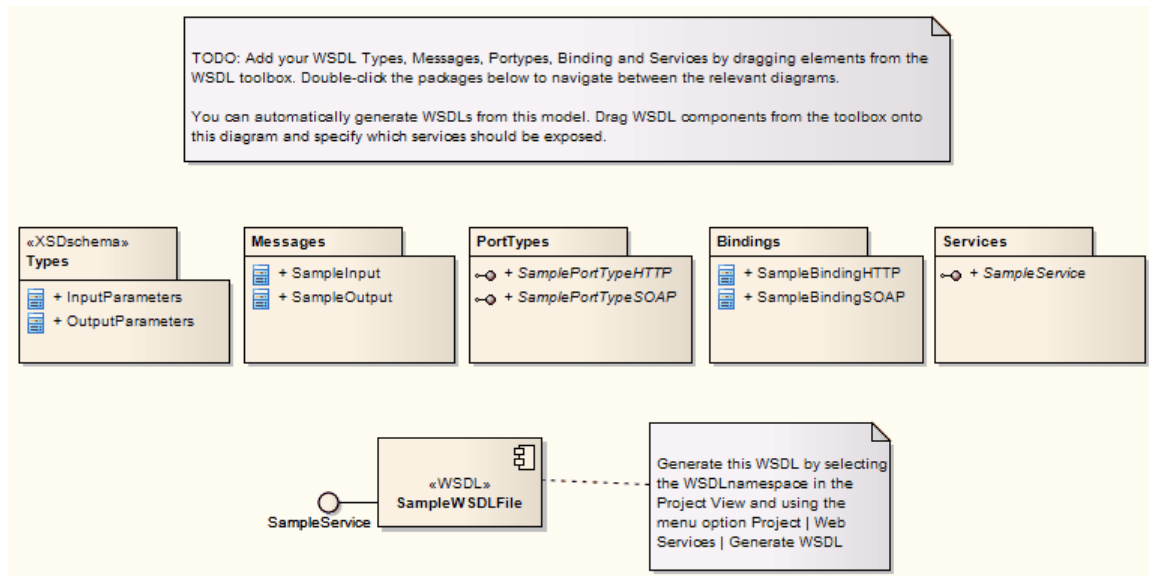
您可以使用图形工具箱的 WSDL 页面快速模型地为网络服务定义中的图表工具箱。作为第一步，您可以使用 WSDL 页面中的命名空间图标在浏览器窗口中创建一个示例 WSDL包结构。您可以将此示例包结构用作开发 WSDL 的模板。

创建一个新的 WSDL包结构

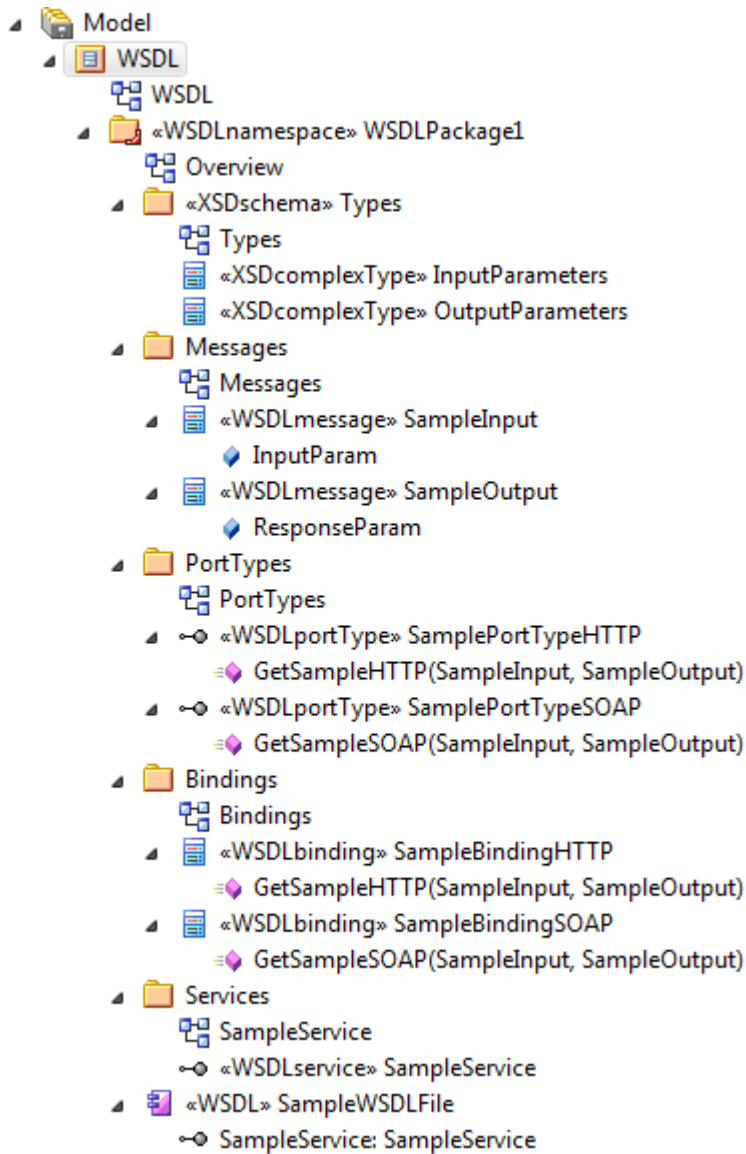
节	行动
1	在浏览器窗口中，创建您需要的顶层项目结构（模型和视图），然后单击相应的视图。
2	单击浏览器窗口标题下拉列表中的“新包”选项。 将显示“新模型包”对话框。
3	在“名称”字段中输入新包的名称，然后选择“创建图表”单选按钮。
4	单击确定按钮。 将显示“新图表”对话框。
5	在“名称”字段中输入新图表的名称。 在“Select From”面板中选择“UML Structural”，在“图表”面板中选择“类”。
6	单击确定按钮。 在浏览器窗口中，双击新图表名称旁边的图标；图表在“图表视图”中打开，类页面显示在“图表工具箱”中。
7	在  工具箱显示“查找工具箱项”对话框并指定“WSDL”，然后从结果中选择工具箱页面。 显示“WSDL”工具箱页面。
8	单击工具箱中的“命名空间”图标并将其拖到类图中。 将显示“WSDL命名空间属性”对话框。 WSDL包名和目标命名空间的 URL 中的类型。您可以稍后编辑这些值。
9	单击确定按钮。 在图表上创建样本«WSDLnamespace»原型包结构，并在浏览器窗口中显示、展开完整模型结构。 模型结构由这些子包组成，并带有概览图以在它们之间导航： <ul style="list-style-type: none"> • 类型：包含由 Web 服务通信的数据的 XSD 类型，在类型图上 • 消息：包含 WSDL 消息，建模为 UML 类，用构造型 «WSDLmessage» 标记 • PortTypes：包含 WSDL 端口类型，建模为 UML 接口，标有构造型 «WSDLportType» • 绑定：包含 WSDL 绑定，建模为实现 PortTypes 的 UML 类 • 服务：包含 WSDL 服务，建模为 UML 接口，并关联到每个暴露捆绑
10	在其对应的包中对每个模型构造进行建模。

模板 WSDL模型-图表

WSDL 命名空间充当 WSDL 结构的容器。



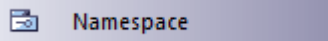
Template WSDL模型-浏览器Window Hierarchy



WSDL命名空间

«A» 原型包代表Enterprise Architect中 WSDL 构造的顶级容器。您可以通过将 WSDL工具箱页面中的命名空间图标拖放到图表上来创建命名空间包。

工具箱图标



访问

要为选定的 «属性» 原型包显示 “WSDL 命名空间属性”对话框，请使用此处概述的方法之一。

功能区	设计>包>管理>属性
上下文菜单	右键单击 «WSDLnamespace» 原型包 属性
其它	在浏览器窗口中，双击 «WSDLnamespace» 原型包，或 将  图标从工具箱拖到图表上（这将创建一个新的 «WSDLnamespace» 原型包）

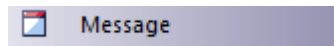
定义属性

选项	行动
WSDL包名称	类型在 WSDL命名空间包元素的名称中。
目标命名空间	(可选) 类型在 WSDL命名空间包的 URL 中。
确定	单击此按钮保存输入的值并关闭 WSDL命名空间的“属性”对话框。 如果您刚刚创建了命名空间，则会打开一个包含示例模板WSDL模型的新包图。
取消	单击此按钮放弃输入的数据并关闭“WSDL命名空间属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您正在编辑现有的 WSDL命名空间元素信息时，会显示此按钮。 单击 按钮打开命名空间包元素的UML元素“属性”对话框。

WSDL信息

«WSDLmessage» 原型类表示A WSDL信息并充当一个或多个 WSDL信息部件的容器。您可以通过将信息图标从 WSDL工具箱拖放到消息图上 (在 WSDL包结构中的 Messages包下) 来创建 WSDL 消息。

工具箱图标



访问

要为选定的 «WSDLmessage» 原型类显示 “WSDL信息”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «WSDLmessage» 原型类 属性
键盘快捷键	Alt+Enter
其它	双击 «WSDLmessage» 原型类，或 将  图标从工具箱中拖放到 Messages 图表上，在 WSDL包结构中的 Messages包下。 (这会创建一个新的 «WSDLmessage» 原型类。)

定义属性

选项	行动
名称	WSDL信息名称中的类型。
文档	(可选) 类型在您需要此元素的任何注记中。
确定	单击此按钮保存输入的数据并关闭 WSDL信息对话框。
取消	单击此按钮放弃输入的数据并关闭“WSDL信息”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 WSDL信息元素信息时，会显示此按钮。 单击按钮打开元素的UML类的 “属性”对话框。

注记

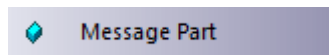
- WSDL 消息只能在 WSDL包结构中的 Messages包下创建
- WSDL信息的名称在 WSDL 内的所有 WSDL 消息中应该是唯一的

WSDL信息部件

WSDL信息部件是 WSDL信息A一部分，它标识由信息通信的数据的 XSD 数据类型。如果信息传递的数据不止一种数据类型，则每种数据类型都在单独的信息部件中标识。

在Enterprise Architect中，WSDL信息部件由 WSDL信息类的UML属性表示。您可以通过从 WSDL工具箱中拖动“信息部件”图标并将其放到«部件»原型类上来创建信息元件属性。

工具箱图标




访问

要显示选定信息元件的“部件信息部件”对话框，请使用此处列出的方法之一。

功能区	在图表上的 WSDL信息中选择特定的信息部件（属性）： 设计>元素>特征>属性
上下文菜单	在图表上的 WSDL信息中选择特定的信息部件（属性）： 右键单击属性 视图属性
键盘快捷键	在图表上的 WSDL信息中选择特定的信息部件（属性）： F9
其它	双击 «部件» 原型类中的信息元件（属性），或 将  Message Part 图标从工具箱拖放到 «WSDLmessage» 原型类上（这会在 «WSDLmessage» 原型类中创建一个新的信息部件（作为属性）。）

定义属性

选项	行动
名称	WSDL信息部件属性名称中的类型。
类型	任何一个： <ul style="list-style-type: none"> • 类型数据类型的名称，或 • 单击下拉箭头并从列表中选择 XSD 内置数据类型，或 • 单击  按钮并浏览现有的 «XSDelement»、«XSDcomplexType» 或 «XSDsimpleType»元素作为分类器 XSD 类型可以定义在： <ul style="list-style-type: none"> • WSDL包结构下的Types包或

	<ul style="list-style-type: none">• 模型中的任何其他包
确定	单击此按钮保存输入的数据并关闭“WSDL信息部件”对话框。
取消	单击此按钮可放弃输入的数据并关闭“WSDL信息部件”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 WSDL信息部件属性信息时，会显示此按钮。 单击按钮打开信息部件的属性属性。

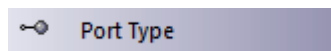
注记

- WSDLmessage 构造型类只能有效地包含信息部件属性；如果在类元素中添加其他属性，它们将被重新转换为信息部件

WSDL端口类型

«A» 原型接口表示 WSDL PortType。它描述了 WSDL 公开的操作，充当一个或多个 WSDL 端口操作类型的容器。您可以通过将 WSDL 端口类型图标从 WSDL 工具箱直接拖放到 PortTypes 图（在 WSDL 包结构中的 PortTypes 包下）来创建 WSDL PortType 元素。

工具箱图标



访问

要为选定的 «WSDLportType» 原型接口显示 “WSDL PortType” 对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击«WSDLportType»原型接口 属性
键盘快捷键	Alt+Enter
其它	<ul style="list-style-type: none"> • 双击 «WSDLportType» 原型接口，或 • 将  图标从工具箱中拖放到 PortTypes 图上，在 WSDL 包结构中的 PortTypes 包下（这会创建一个新的 «WSDLportType» 原型接口。）

定义属性

选项	行动
名称	WSDL PortType 名称中的类型。
文档	（可选）类型在您需要此元素的任何注释中。
确定	单击此按钮以保存输入的数据并关闭 WSDL PortType 对话框。
取消	单击此按钮放弃输入的数据并关闭“WSDL PortType”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 WSDL PortType 元素信息时，会显示此按钮。单击按钮打开该元素的 UML 元素“属性”对话框。

注记

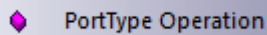
- WSDL PortTypes 只能在 WSDL包结构中的 PortTypes包下创建
- WSDL PortType 的名称在 WSDL 内的所有 WSDL PortType 中应该是唯一的

WSDL端口类型操作

A类型端口标识消息的交换（作为操作的接口的数据输入和输出）。端口类型操作还可以识别作为故障指示器的消息。


在Enterprise Architect中，WSDL PortType 操作由 WSDL PortType接口的UML操作表示。您可以通过将 PortType Operation 图标从 WSDL 工具箱拖放到 «工具箱» 原型接口上来创建 PortType Operation。

工具箱图标



访问

要为选定的 PortType 操作显示 “WSDL PortType 操作”对话框，请使用此处概述的方法之一。

功能区	在图表上的“接口”原型接口中选择特定的端口类型操作： 设计>元素>特征>操作
上下文菜单	在图表上的“接口”原型接口中选择特定的端口类型操作： 右键单击属性 视图属性
键盘快捷键	在图表上的“接口”原型接口中选择特定的端口类型操作： F10
其它	双击 «WSDLportType» 原型接口中的 PortType 操作，或 将  PortType Operation 图标从工具箱中拖放到 «WSDLportType» 原型接口上。（这会在 «WSDLportType» 原型接口中创建一个新的 PortType 操作（作为UML操作）。）

定义属性

选项	行动
名称	WSDL PortType 操作的名称中的类型。
文档	（可选）此操作所需的任何标记中的类型。
操作类型	单击下拉箭头并选择一种受支持的 PortType 操作类型： <ul style="list-style-type: none"> • 单程 • 请求-响应 • 请求响应 • 通知

输入	<p>如果您选择了通知作为操作类型，此部分将显示为灰色。</p> <ul style="list-style-type: none"> 名称- 默认为与操作类型平行的名称。如果您不想使用默认值，请键入输入信息的替代名称。 信息- 单击下拉箭头并选择先前在消息包中创建的 WSDL 消息之一。 文档 - (可选) 输入信息所需的任何注记类型。
输出	<p>如果您已选择 OneWay 作为操作类型，此部分将显示为灰色。</p> <ul style="list-style-type: none"> 名称- 默认为与操作类型平行的名称。如果您不想使用默认值，请键入输出信息的替代名称。 信息- 单击下拉箭头并选择先前在消息包中创建的 WSDL 消息之一。 文档 - (可选) 类型在您需要此输出信息的任何注记中。
故障	<p>审阅可以充当故障的 WSDL 消息的详细信息。</p> <p>故障显示在此列表中，最近创建的在顶部，最旧的在最后。如果定义了四个以上的故障消息，请使用垂直滚动条显示列表的其余部分。</p> <p>要添加信息，请单击新建按钮。将显示 “WSDL PortType Operation Fault”对话框。</p> <ul style="list-style-type: none"> 名称- 默认为 'Fault<n>';如果您不想使用默认值，请键入故障信息的替代名称 信息- 单击下拉箭头并选择先前在信息包中创建的 WSDL 消息之一 文档 - (可选) 类型in any注记you need for this fault信息 确定- 单击此按钮保存输入的数据并关闭“WSDL PortType Operation Fault”对话框 取消 - 单击此按钮放弃输入的数据并关闭“WSDL PortType Operation Fault”对话框 帮助- 单击此按钮显示此帮助主题 <p>要从列表中删除信息，请单击它并单击删除按钮。</p>
确定	单击此按钮保存输入的数据并关闭“WSDL PortType Operation”对话框。
取消	单击此按钮放弃输入的数据并关闭“WSDL PortType Operation”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	<p>当您正在编辑现有的 WSDL 端口类型操作信息时，会显示此按钮。</p> <p>单击按钮打开元素的UML操作“属性”对话框。</p>

注记

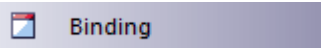
- WSDL操作只能由 WSDL PortTypes 包含
- 为 PortType 操作中的输入、输出或故障信息提供的名称在 WSDL PortType 中的输入、输出和故障消息中必须分别是唯一的
- 在UML操作 属性”对话框中，标识为 Input、输出和 Fault 的 Messages 可以作为操作的参数进行检查

WSDL 捆绑

A WSDL 捆绑元素实现由特定的 «接口» 原型接口指定的操作，并描述此 WSDL PortType 定义的操作和消息的消息格式和协议细节。A WSDL 捆绑元素由一个 «WSDLbinding» 原型类表示。

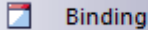
您可以通过将 WSDL 工具中的捆绑图标直接拖到 WSDL 包结构中工具箱包下的 Bindings 图上来创建一个 WSDL 捆绑元素。

工具箱图标



访问

要为选定的 «WSDLbinding» 原型类显示 “WSDL 捆绑”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击 «WSDLbinding» 原型类 属性
键盘快捷键	Alt+Enter
其它	<p>双击一个 «WSDLbinding» 原型类，或者</p> <p>将  图标从工具箱中拖放到 Bindings 图上，在 WSDL 包结构中的 Bindings 包下。</p> <p>(这会创建一个新的 «WSDLbinding» 原型类。)</p>

定义属性

选项	行动
名称	类型在 WSDL 捆绑元素的名称中。
端口类型	点击下拉箭头，选择此 WSDL 捆绑要实现 WSDL 捆绑。
协议	<p>单击下拉箭头并选择传输所选 WSDL PortType 操作的协议。支持的协议有：</p> <ul style="list-style-type: none"> 肥皂 HTTP
运输	<p>如果您选择了 HTTP 协议，此字段将被禁用。</p> <p>默认为 http://schemas.xmlsoap.org/soap/http。</p> <p>如有必要，输入 SOAP 协议的替代 URL。</p>

风格	如果您选择了 HTTP 协议，此字段将被禁用。 单击下拉箭头并选择 SOAP 协议的样式。
动词	如果您选择了 SOAP 协议，此字段将被禁用。 单击下拉箭头并选择适当的 HTTP 动词。支持的动词有： <ul style="list-style-type: none">• 得到• 邮政
文档	(可选) 类型在您需要此元素的任何注记中。
确定	单击此按钮保存输入的数据并关闭“WSDL 捆绑”对话框。
取消	单击此按钮可丢弃输入的数据并关闭“WSDL 捆绑”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您正在编辑现有的 WSDL 捆绑元素信息时，会显示此按钮。 单击按钮打开该元素的 UML 元素“属性”对话框。

注记

- A WSDL 捆绑必须实现一个 WSDL PortType；因此，应该在创建 WSDL 绑定之前定义 WSDL PortTypes
- WSDL Bindings 只能在 WSDL 包结构中的 Bindings 包下创建
- WSDL 绑定的名称在 WSDL 内的所有 WSDL 捆绑中应该是唯一的

WSDL 捆绑操作

当你保存一个新创建的 «WSDLbinding» 原型类时，系统：

1. 添加到捆绑图中，WSDL 捆绑实现的 WSDL 端口类型元素。
2. 从捆绑到 PortType 绘制一个实现连接器。
3. 使用捆绑中的所有 UML 操作自动填充捆绑，作为 WSDL 捆绑操作。

访问

要显示所选捆绑操作的 “WSDL 捆绑操作详细信息” 对话框，请使用此处概述的方法之一。

功能区	在图表上的 «捆绑» 原型类中选择特定的捆绑操作： 设计>元素>特征>操作
上下文菜单	在图表上的 «捆绑» 原型类中选择特定的捆绑操作： 右键单击属性 视图属性
键盘快捷键	在图表上的 «捆绑» 原型类中选择特定的捆绑操作： F10
其它	双击«捆绑»类中的捆绑操作

定义属性

选项	行动
操作名称	显示从 WSDL PortType 元素复制的操作的名称。 此字段中的值无法编辑。
行动	如果父级 WSDL 捆绑元素的协议定义为 HTTP，则该字段灰显。 此操作的 SOAP 行动标头 (URL) 中的类型。
风格	如果父级 WSDL 捆绑元素的协议定义为 HTTP，则该字段灰显。 单击下拉箭头并选择操作的 SOAP 样式。
地点	如果父 WSDL 捆绑元素的协议定义为 SOAP，则该字段灰显。 此操作的相对 URL 中的类型。
文档	(可选) 此操作所需的任何注记中的类型。
参数	单击此按钮可定义此操作的参数。 将显示 “WSDL 捆绑操作参数” 对话框，显示操作输入、输出和故障的名称。您不能更改这些名称。 单击详细信息按钮以指定输入、输出和故障操作 (信息) 参数的详细信息。请

	<p>注记，详细信息按钮在：</p> <ul style="list-style-type: none"> • 通知操作类型的输入部分被禁用 • 单向操作类型禁用输出部分 • 如果没有故障消息，则故障部分被禁用 • 使用-如果父WSDL捆绑元素的协议定义为HTTP，则该字段灰显；单击下拉箭头并选择要使用的编码 • Encoding Style - 如果父 WSDL捆绑元素的协议定义为 HTTP，则该字段灰显；如果 使用“设置为 编码”，请输入要应用的样式（URL） • 命名空间-如果父WSDL捆绑元素的协议定义为HTTP，则该字段灰显；（可选）在命名空间中输入 • Parts - 如果父 WSDL捆绑元素的协议定义为 HTTP，则该字段灰显；此字段也不适用于故障 - （可选）出现在 SOAP体部分中的信息部件属性中的类型 • Header - 此字段不适用于故障；（可选）输入 SOAP/HTTP 标头的文本 • 文档 - （可选）类型在您需要此消息的任何注记中 • 确定- 单击此按钮保存输入的数据并关闭“WSDL捆绑参数详细信息”对话框 • 取消 - 单击此按钮放弃输入的数据并关闭“WSDL捆绑参数详细信息”对话框 • 帮助- 单击此按钮显示此帮助主题
确定	单击此按钮保存输入的数据并关闭“WSDL捆绑操作详细信息”对话框。
取消	单击此按钮可丢弃输入的数据并关闭“WSDL捆绑操作详细信息”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您正在编辑现有的 WSDL捆绑操作信息时，会显示此按钮。 单击按钮打开元素的UML操作“属性”对话框。

注记

- 如果后续更改了 WSDL端口类型的操作，可以通过删除实现连接器并重新建立来刷新捆绑操作；将显示 覆盖和实现”对话框，您可以在其中选择要建立的更新操作
- 您可以通过在图表中突出显示操作或浏览器窗口并展开属性窗口中的条目来捆绑审阅的参数

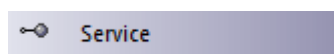
WSDL 服务

WSDL 服务由A «接口» 原型接口表示；它描述了暴露特定捆绑的端口集合。您可以通过将服务图标从 WSDL 工具箱拖放到您的 WSDL模型的 Services包中的图表上来创建 WSDL 服务元素。

当您保存新创建的 «接口» 原型接口时，系统：

1. 将 WSDL 服务公开的 WSDL捆绑元素添加到服务图中。
2. 从服务元素到每个捆绑元素绘制一个关联连接器。
3. 用相应的端口名称标记每个连接器。

工具箱图标



访问

要为选定的 «接口» 原型接口显示 “WSDL 服务”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击«WSDLservice»原型接口 属性
键盘快捷键	Alt+Enter
其它	双击 «WSDLservice» 原型接口，或 将  图标从工具箱中拖放到 SampleService 图上，在 WSDL包结构中的 Services包下。（这会创建一个新的 «WSDLservice» 原型接口。）

定义属性

选项	行动
名称	WSDL 服务名称中的类型。
文档	(可选) 类型在您需要此元素的任何注释中。
端口	标识此 WSDL 服务的端口 (或端点)。 要将端口添加到列表中，请单击 “新建”按钮。将显示 “WSDL端口”对话框。 <ul style="list-style-type: none"> • 端口名称-类型中的端口名称 • 捆绑-点击下拉箭头，从 BindingsPackage 中创建的所有 WSDL Bindings 列表选择一个捆绑元素 • Location -端口的 URL 类型

	<ul style="list-style-type: none"> • 文档 - (可选) 在您需要此端口类型的任何注记中 • 确定- 单击此按钮保存输入的值并关闭“WSDL端口”对话框 • 取消 - 单击此按钮放弃输入的值并关闭“WSDL端口”对话框 • 帮助- 单击此按钮显示此帮助主题 <p>端口排列在列表中，最新的在顶部，最旧的在最后。 要从列表中删除条目，请单击它并单击删除按钮。</p>
确定	单击此按钮以保存输入的数据并关闭 WSDL 服务对话框。
取消	单击此按钮放弃输入的数据并关闭“WSDL 服务”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 WSDL 服务元素信息时，会显示此按钮。 单击按钮打开该元素的UML元素“属性”对话框。

注记

- WSDL 服务只能在 WSDL包结构中的 Service包下创建
- WSDL 服务的名称在 WSDL 内的所有 WSDL 服务中应该是唯一的

WSDL 文档

WSDL 文档封装了在«WSDLnamespace» 原型包中定义A网络服务，并且是生成WSDL 文件的源。它由«WSDL» 原型组件元素表示为«部件» 原型包的直接子元素。您可以在单个 WSDL命名空间下拥有多个 WSDL 文档，以便跨多个 WSDL 重用和公开该名称空间的 WSDL 服务。

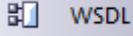
当您创建命名空间包结构时，会自动创建一个 «WSDL» 定型部件元素。您可以通过将 WSDL 图标从 WSDL 工具箱拖放到名称空间概览图上来添加更多 WSDL 元素。

工具箱图标



访问

要为选定的 «WSDL» 原型部件显示 “WSDL 文档属性”对话框，请使用此处概述的方法之一。

功能区	设计>元素>编辑>属性
上下文菜单	右键单击«WSDL»构造型部件 属性
键盘快捷键	Alt+Enter
其它	双击一个«WSDL»原型部件，或者 将  图标从工具箱中拖放到概述图上，在 WSDL包结构中的 «WSDLnamespace» 原型包下。（这将创建一个新的 WSDL 文档，由“WSDL”原型部件表示。）

定义属性

选项	行动
名称	WSDL 文档名称中的类型。
文件名称	类型WSDL 1所在的文件路径。将生成1文件。
文档	(可选) 类型在您需要此元素的任何注释中。
XMLNS	识别此 WSDL 文档中使用的附加名称空间或名称空间前缀对。 要添加命名空间或命名空间前缀对，请单击 “新建” 按钮；要编辑现有条目，请双击它。在任何一种情况下，都会显示 “命名空间详细信息”对话框。 <ul style="list-style-type: none"> 前缀类型在缩略文本中代表命名空间 命名空间-命名命名空间名称中的类型 确定- 单击此按钮保存新信息并关闭“命名空间详细信息”对话框

	<ul style="list-style-type: none">• 取消 - 单击此按钮可放弃新信息并关闭“命名空间详细信息”对话框• 帮助- 单击此按钮显示此帮助主题 要从列表中删除条目，请单击它并单击删除按钮。
服务	审阅服务包中存在的 WSDL 服务。 选中要包含在当前 WSDL 文件中的服务对应的复选框。
确定	单击此按钮保存输入的数据并关闭 WSDL 文档的“属性”对话框。
取消	单击此按钮放弃输入的数据并关闭“WSDL 文档属性”对话框。
帮助	单击此按钮可显示此帮助主题。
UML	当您编辑现有的 WSDL 文档元素信息时，会显示此按钮。 单击按钮打开该元素的UML元素“属性”对话框。

生成WSDL

如果您已经在UML中开发了一个 WSDL模型，您可以将它转发工程到 WSDL 1中。1文件使用生成的 WSDL特征。该特征作用于 WSDLnamespace“原型包或“WSDL”原型部件（文档），并生成目标 WSDLnamespace“结构拥有的任何或所有WSDL组件。

访问

功能区	开发>架构建模>导出WSDL
-----	----------------

生成WSDL 1.1文件

选项	行动
WSDL包	显示包含要从中生成 WSDL 文件的源部件的 WSDL命名空间的名称。
编码	任何一个： <ul style="list-style-type: none"> 单击下拉箭头并选择您需要的 XML 编码方案，或 单击默认按钮以应用默认编码方案 (UTF-8)
选择要生成组件	单击要为其生成 WSDL 文件的列表中的 «WSDL» 原型部件。 至： <ul style="list-style-type: none"> 使用 Ctrl+单击选择多个单独的组件 使用 Shift+单击选择范围 选择列表中的所有条目单击 全选”按钮 清除列表中的所有条目单击 选择无”按钮 提供为组件生成 WSDL 文件的文件路径和名称，双击组件名称；将显示“部件文件名称”对话框，请表中的说明
生成	单击此按钮为选定的 «WSDL» 原型组件生成 WSDL 文件。 生成完成时会显示A消息；单击消息上的确定按钮将其关闭。
视图WSDL	单击此按钮可显示最近生成的 WSDL。
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主题。
进步	监视 WSDL 文件生成的进度。

部件文件名称对话框

字段/按钮	描述
名称	显示选定的 «WSDL» 原型部件的名称。
前缀	如果已指定多个前缀，请单击下拉箭头并为 WSDL命名空间选择适当的前缀。
文件名称	类型输入或浏览 (单击 ) 要生成 WSDL 文件的文件路径和名称。
确定	单击此按钮保存输入的数据并关闭“部件文件名称”对话框。
取消	单击此按钮放弃输入的数据并关闭“部件文件名称”对话框。
帮助	单击此按钮可显示此帮助主题。

注记

- 您还可以通过自动化接口生成 WSDL 文件

导入WSDL

如果您有 WSDL 1.1 Enterprise Architect外部的文件，如果您想逆向工程为UML类模型，您可以使用 WSDL导入功能将它们导入系统。

访问

浏览器窗口 | 点击根包包含导入的文件，然后：

功能区	开发>架构建模>导入WSDL
-----	----------------

导入一个 WSDL文件

选项	行动
包	显示要导入WSDL文件的根包的名称。
文件名	类型输入或浏览（单击  ）要导入的 WSDL 文件的名称和路径。
目标包	默认为正在导入的 WSDL 文件的名称，作为包的名称来表示导入的文件。如果您不想使用默认名称，请键入不同的名称。
导入	单击此按钮以启动 WSDL 导入。 导入完成时会显示A消息；单击消息上的确定按钮将其关闭。
关	单击此按钮可关闭此对话框。
进步	监视 WSDL 导入的进度。

注记

- Enterprise Architect不能导入引用存在于该文件之外的 WSDL 构造的 WSDL 文件；如果其他文件中有引用的结构，请将所有文件合并到一个文件中并导入该合并文件
- 可导入文件的示例：http://www.w3.org/TR/wsdl.html#_wsdl
- 不可导入文件的示例：http://www.w3.org/TR/wsdl.html#_style；尝试导入此文件会导致错误消息无法导入 Split Files

SoaML

面向服务的架构建模语言 (SoaML) 是一种使用 Unified Modeling Language (UML) 设计和建模 SOA 解决方案的标准方法。

本文来源于面向服务的架构建模语言 (SoaML) —— UML 配置文件和服务元模型 (UPMS) 规范 (OMG 文档 ad/2008-11-01) ; 第 25-26 页 :

A 服务是通过定义明确的界面向他人提供价值, 并可供社区 (可能是一般公众) A 。服务导致工作由他人提供。 ”

面向服务的架构(SOA) 是一种组织和理解 (表示) 组织、社区和系统的方式, 以最大限度地提高敏捷性、规模和互操作性。SOA 方法很简单——人员、组织和系统相互提供服务。这些服务让我们不用自己做, 甚至不知道怎么做就能完成某事——使我们更高效、更敏捷。服务还使我们能够将我们的能力提供给他人以换取一些价值——从而建立一个社区、流程或市场。SOA 范式在集成现有功能和创建和集成新功能方面同样适用。

“SOA ... 是一种架构范式, 用于定义人员、组织和系统如何提供和使用服务来实现结果模型... 提供了一种使用 Unified Modeling Language (UML) 来构建和建模 SOA 解决方案的标准方法。配置文件使用UML的内置扩展机制根据现有UML概念定义 SOA 概念。”

“..... 使用 SOA 的最大优势来自于将社区、流程或企业理解为一组相互关联的服务, 并.....用支持服务的系统支持面向服务的企业。SoaML 使面向业务和面向系统的服务架构能够相互协作地支持企业使命。... SoaML 依靠模型驱动架构® (MDA®) 来帮助将业务和系统架构、企业设计映射到支持 SOA 的技术, 例如 Web 服务和 CORBA®。”

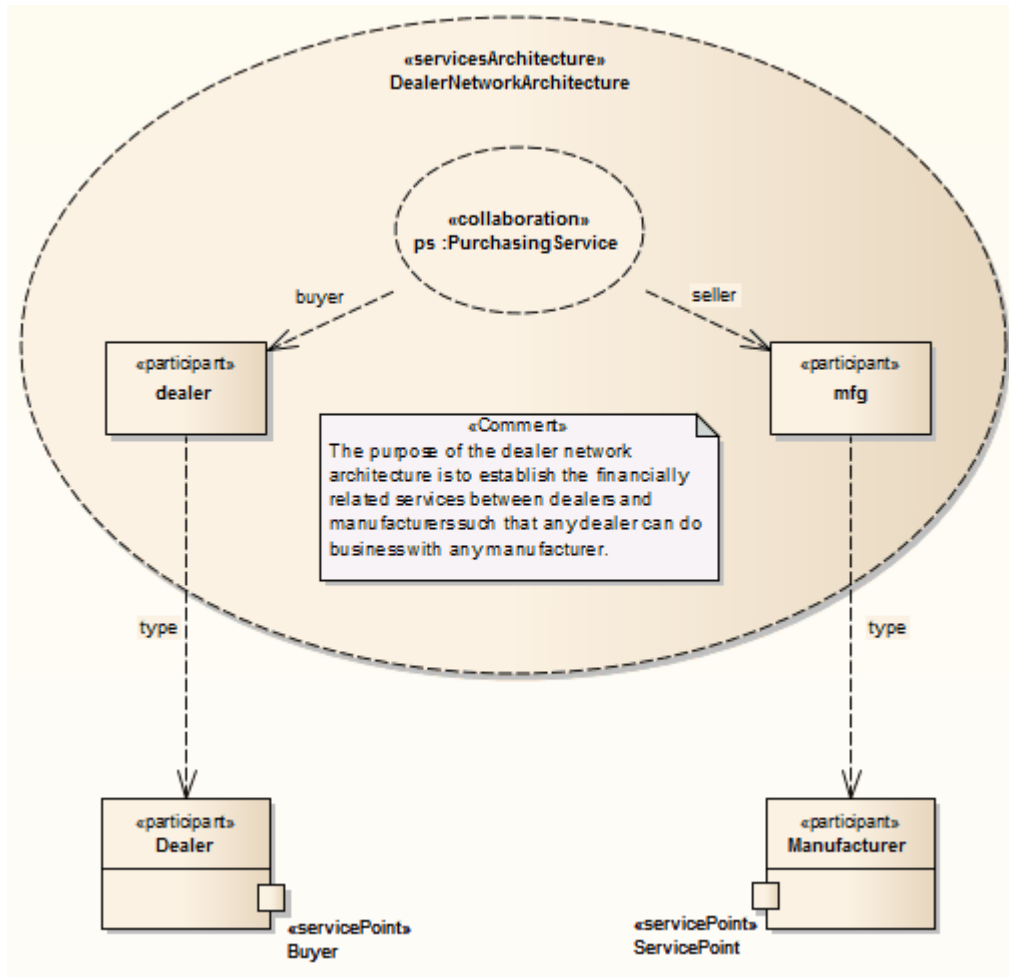
有关 SoaML 概念的更多信息, 请参阅 OMG 网站 SoaML 文档页面上的规范文档。”

Enterprise Architect中的 SoaML

在Enterprise Architect中, 您可以通过使用与Enterprise Architect安装程序集成的模型MDG 技术快速简单地对服务架构进行建模。 SoaML功能以以下形式提供 :

- 两种 SoaML 图类型 - SoaML 部件图和 SoaML 序列图 - 通过 “新图表” 对话框访问
- 工具箱中的图表页面
- “工具箱快捷方” 菜单和快速链接器中的 SoaML 元素和关系条目

示例SoaML图表



注记

- Enterprise Architect的企业版、统一版和终极版支持面向服务的架构建模语言 (SoaML)

SoaML工具箱Pages

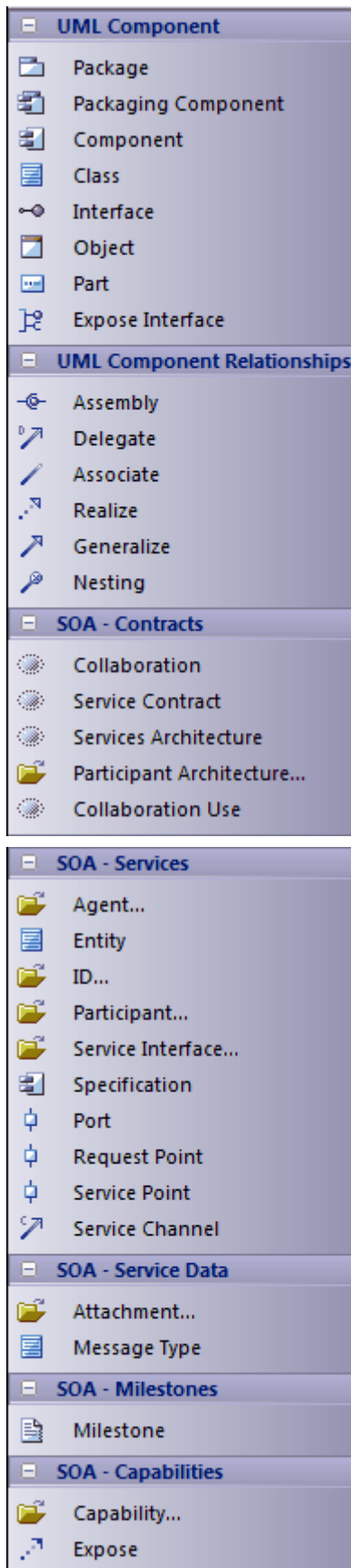
您可以使用工具箱的“SoaML”页面创建图表模型的元素和关系。这两种 SoaML 图类型中的每一种都有一组单独的页面，尽管这两组中的最后五个（SOA 特定）页面是相同的。

访问

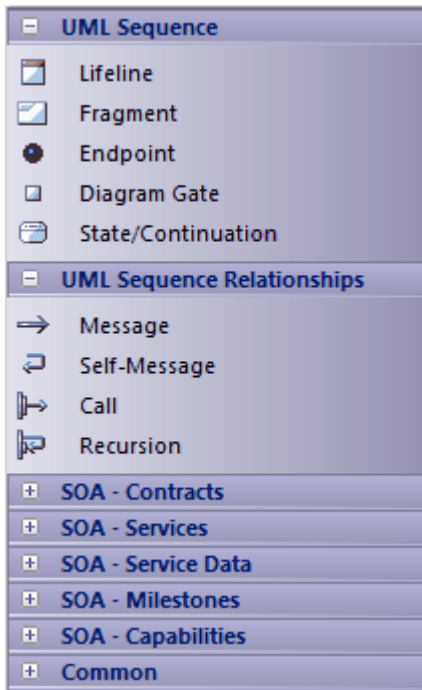
功能区	设计>  工具箱图表查找工具箱项”对话框并指定“部件工具工具”或“SoaML序列”
键盘快捷键	Ctrl+Shift+3 :  > 在 查找工具箱项”对话框中指定“SoaML部件”或“SoaML序列”
其它	图表标题栏 单击  图标以显示  工具箱图表 查找工具箱项”对话框中指定“SoaML部件”或“SoaML序列”

工具箱页面

图表部件工具箱



图表序列工具箱



SOMF 2.1

面向服务的建模框架 (SOMF) 是一种面向服务的开发生命周期方法论，提供了许多有助于成功的面向服务的生命周期管理和建模的建模实践和规则。这段文字来自于广泛的 Wikipedia 上关于面向服务的建模的条目：

'面向服务的建模框架 (作者) 已由 Michael Bell 提出，作为一种用于软件开发的整体和拟人化建模语言，它采用学科和通用语言为企业问题提供战术和战略解决方案。术语“整体语言”属于一种建模语言，可用于设计任何应用程序、业务和技术环境，无论是本地的还是分布式的。这种普遍性可能包括应用程序级和企业级解决方案的设计，包括 SOA 环境或云计算环境。另一方面，术语“拟人化”将 SOMF 语言与实现的直观性和使用的简单性联系起来。

'SOMF ...说明了确定服务开发计划“做什么”方面的主要元素。这些建模支柱将使从业者能够制定有效的项目计划并确定面向服务的计划的里程碑——无论是小型企业还是大型企业或技术企业。

Enterprise Architect中的 SOMF

在Enterprise Architect中，SOMF 2.1被实现为与Enterprise Architect安装程序集成的MDG 技术中的配置文件。SOMF 2.1功能以以下形式提供：

- 11 种 SOMF 图类型，可通过“新图表”对话框访问：
 - 概念
 - 分析
 - 云计算
 - 逻辑设计关系
 - 逻辑设计组合
 - 业务集成
 - 概念架构
 - 资产利用
 - 交易
 - 交易目录
 - 参考架构
- 工具箱中的 SOMF 页面工具箱Enterprise Architect为每个 SOMF 2 包括几个建模结构的工具箱页面。1 图表类型，通过工具箱搜索功能定位；这些提供了广泛的 SOMF 建模功能
- 工具箱快捷菜单和快速链接器中的工具箱元素和关系条目

全国信息交换NIEM建模2.1

国家信息交换NIEM建模提供了一个通用框架，用于定义如何在系统、政府机构和部门之间共享信息。NIEM UML配置文件帮助您：

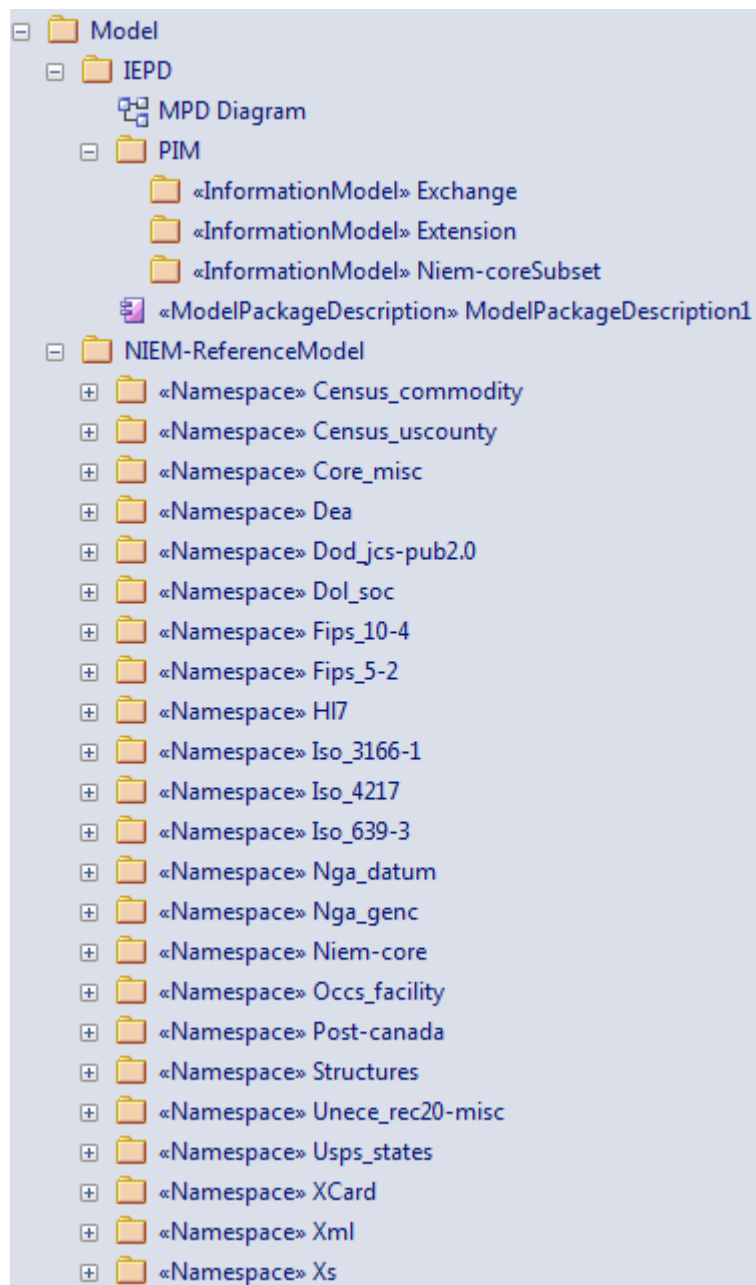
- 创建和开发基于 UML 的信息交换包文档 (IEPD) 模型，或者：
 - 从Enterprise Architect模式生成 IEPD 以生成所有必要的交换文件，静态工件、元数据和目录文件，或
 - 使用Schema 编辑器自动生成您自己的NIEM子集命名空间检测相互依赖关系，并使用生成的子集模式来构建您自己的 IEPD
- 使用NIEM工具箱页面创建 PIM、PSM 和模型包描述(MPD) 图
- NIEM导入参考架构您的模型
- 从您的模型生成NIEM架构

从模式创建NIEM IEPD模型

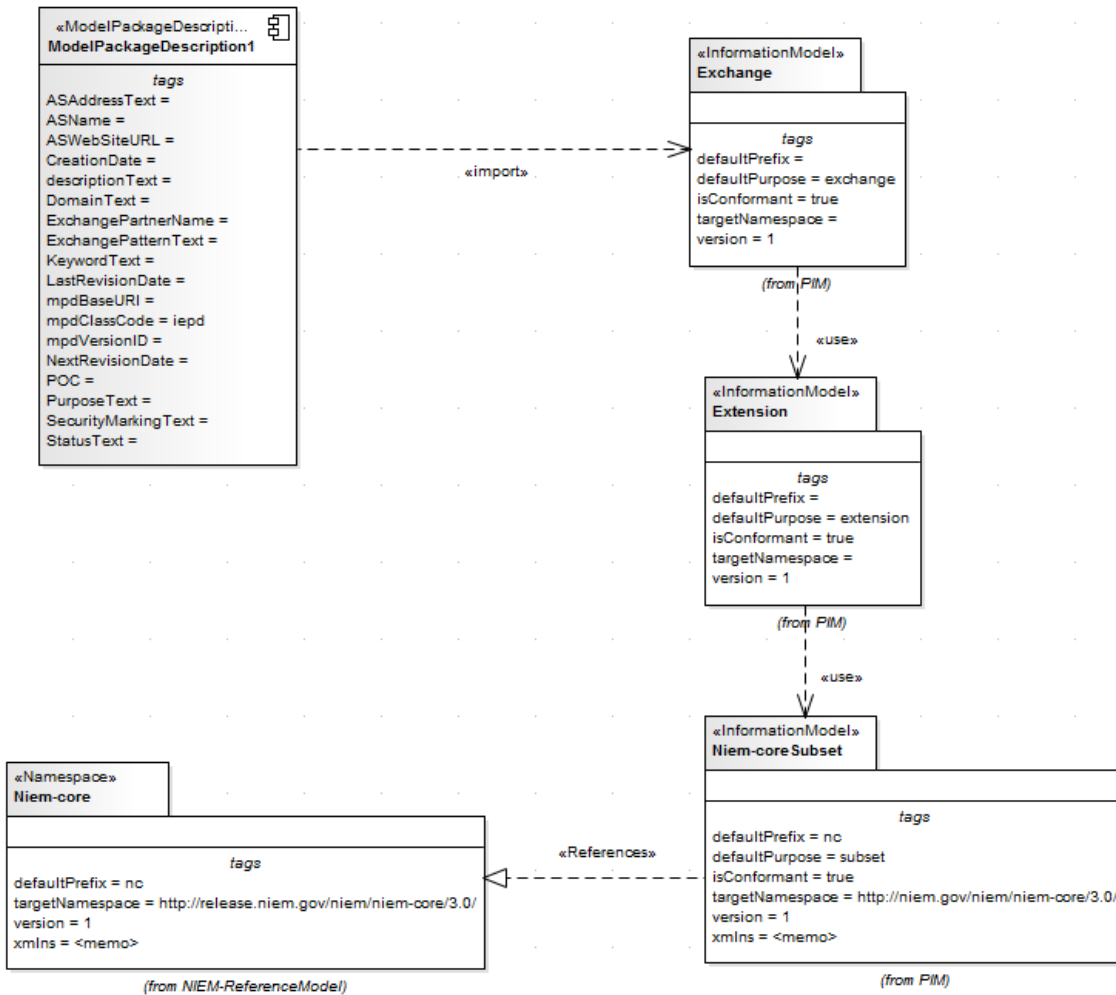
NIEM UML配置文件提供了用于构建 IEPD 模型的模型模式。您可以使用模型构建器在NIEM项目中应用此模式。

1. 打开模型构建器 (Ctrl+Shift+M) ，然后在“<<perspective>>”字段中选择 信息交换> NIEM ”。
2. 在“ NIEM 3 和 4”列表中，滚动浏览技术并单击所需的 “NIEM *nn*参考模型”，然后按 Ctrl + 单击相应的 “ NIEM *n* IEPD Starter模型”。
3. 单击创建模型按钮。

系统生成一个新模型包含一个IEPD包 (本身包含一个PIM包) 和一个NIEM参考模型包。下载参考模型可能需要一些时间。



IEPD包包含一个顶级模型包描述 (MPD) 图 (如图所示), 其中包含MPD部件以及与其相关的所有命名空间和文件。



PIM包由IEPD的所有命名空间和子集命名空间组成。命名空间和MPD部件之间的关系如MPD图所示。MPD部件必须导入至少一个命名空间才能成功生成NIEM模式。

NIEM ReferenceModel包包括所选NIEM版本的所有NIEM参考模式模型。

NIEM图表

您还可以从NIEM图表集和相应的NIEM图表页面创建所有适当的工具箱。这些图表分为三种类型：

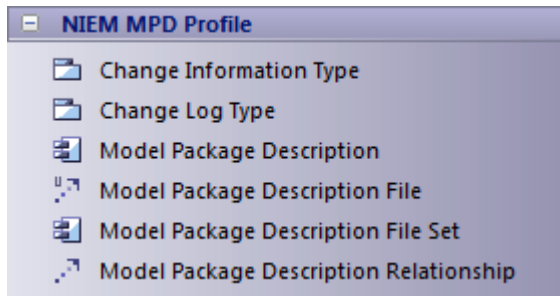
- NIEM平台独立模型 (PIM) 图
- NIEM平台特定模型 (PSM) 图
- NIEM模型包描述(MPD)图

用于开发这些图表的模板可通过 新建图表”对话框获得。

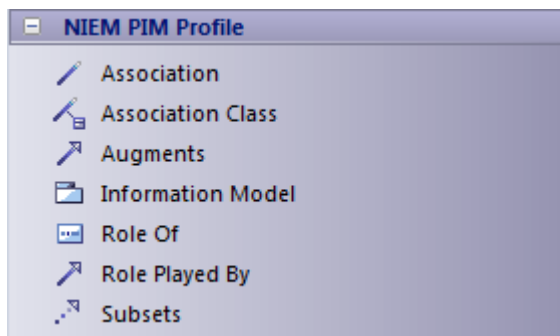
NIEM工具箱Pages

在图表工具箱中，每种图表类型都有自己的元素和连接图表工具箱。NIEM UML配置文件还提供了所有三种图表类型共有的元素和连接器页面。

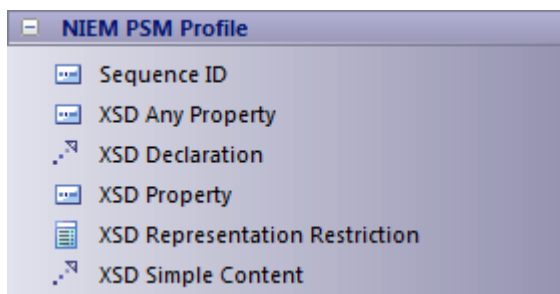
NIEM模型包描述(MPD)配置文件工具箱



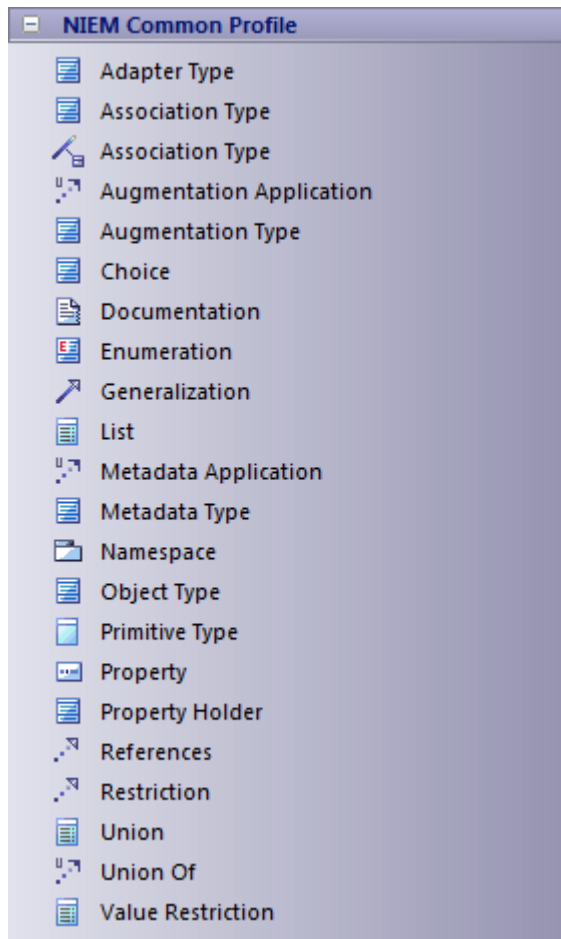
NIEM平台独立模型工具箱



NIEM平台特定工具箱



NIEM公共配置文件工具箱



导入NIEM参考架构

节	行动
1	在浏览器窗口中，右键单击目标包并选择 特定 NIEM 2. 1 导入NIEM 2. 1架构的菜单选项。
2	在“导入XML Schema ”对话框的 目录”字段中，输入或浏览包含要导入的模式的目录，然后选择要导入的 .xsd 模式文件。
3	在“导入XSD元素/属性为：”下选择 “UML属性”单选按钮。
4	点击导入按钮。架构的NIEM模型将被导入到选定的包中。

生成NIEM子集命名空间

您可以使用Enterprise Architect Schema 编辑器生成子集命名空间。这要求NIEM参考模型在模型中可用，因为它是 IEPD模型模式的一部分。

节	行动

1	选择 开发>架构建模> Schema 编辑器>打开Schema 编辑器”功能区选项。
2	单击 配置文件”字段右侧的新建按钮。 将显示 新信息”对话框。
3	在”名称”字段中，键入子集的名称，在”命名空间”字段中，键入名称空间的 http 地址。
4	在 架构集”字段中单击下拉箭头并选择 国家信息交换模型 (NIEM) ”选项。
5	在”保存在”面板中，选择 模型”工件按钮。
6	单击  图标，使用导航员在IEPD中选择namespace/information模型包 PIM，在其下生成子集。
7	在浏览器窗口中，打开NIEM ReferenceModel包NIEM-core。 将此包中的活动拖到Schema 编辑器的左侧列中。 这个元素的属性列在Schema 编辑器的中间一列。
8	单击您需要的每个属性的复选框 - 例如， ActivityName 和 ActivityDateRepresentation。 相应的类/ NIEM object类型被添加到左侧列，而右侧列将它们显示为子项。
9	单击更新按钮以保存子集配置文件。 状态的状态显示在左侧列中的项目名称和列底部的面板中。
10	单击生成按钮。 将显示 架构导出”对话框。
11	在 技术”面板中选中要生成的项目的复选框。必须选择 ‘NIEM模型子集”。
12	单击生成按钮，然后在 查找包”对话框中，选择要在其中生成子集的名称空间/信息模型。
13	单击确定按钮，然后单击第二个确定按钮。子集模型已生成。

NIEM架构一代

一旦您的NIEM IEPD模型及其扩展信息模型、交换信息模型和子集信息模型完成，您就可以从中生成模式。

节	行动
1	右键单击导入 Exchange模型的 MPD部件，然后选择 特定 NIEM 2. 1 生成NIEM 2. 1架构的选项。 将显示 生成NIEM MPD 模式”对话框。
2	在 目录”字段中，键入或浏览要生成架构的目录路径。
3	在” NIEM版本”字段中，单击下拉箭头并选择要为其生成架构的NIEM版本。 将生成的常见工件 (C、Metadata) 是在”MPD 相关工件”面板中列出的静态 MPD 工件和路径，每个都有其对应的路径。 “命名空间架构”面板显示将为信息模型生成的架构文件。

4	点击生成按钮。 生成成功完成后，单击视图架构按钮以访问目录文件。
---	-------------------------------------

全国信息交换NIEM建模

国家信息交换模型 (NIEM) 提供了一个通用框架，用于定义如何在系统、政府机构和组织之间共享信息。Enterprise Architect的NIEM UML配置文件帮助您：

- 创建和开发基于 UML 的信息交换包文档 (IEPD) 模型，通过提供启动模型、模型和许多工具箱页面来创建 IEPD 模型和模式模型
- 从您的 IEPD模型生成完成IEPD
- 从您的信息模型中生成符合NIEM的模式
- NIEM导入参考架构您的模型
- 创建NIEM子集命名空间，由NIEM参考架构的元素组成
- 使用NIEM工具箱页面创建 PIM、PSM 和模型包描述(MPD) 图

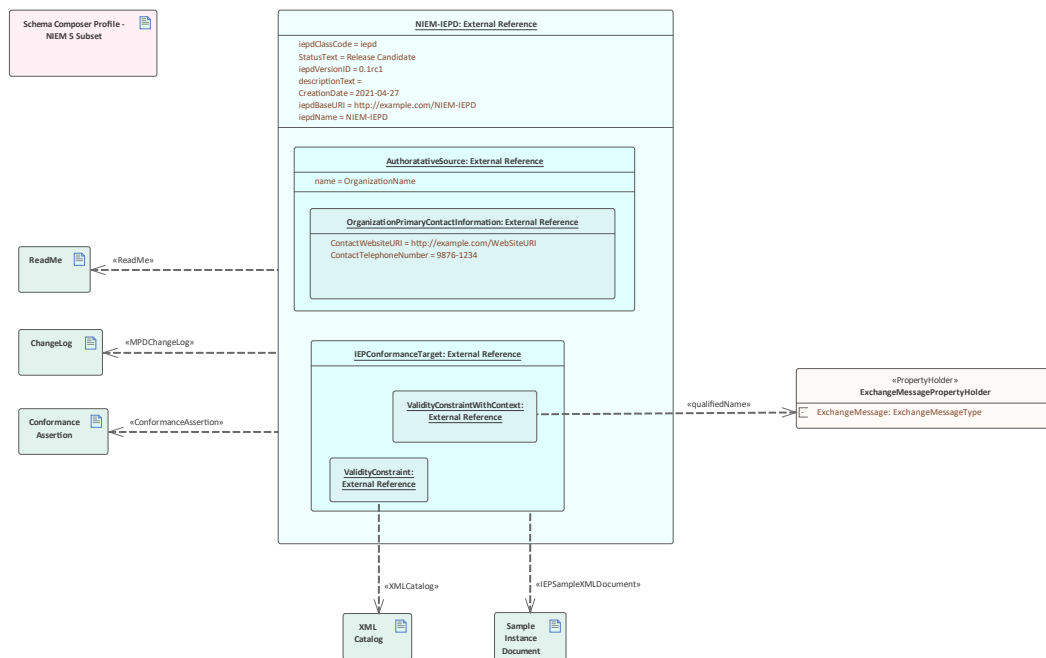
此图显示了NIEM 5.0 starter模型，这是Enterprise Architect中作为NIEM的一部分提供的模式。（请参阅创建 NIEM IEPD帮助主题。）

IEPD Overview

This IEPD Overview diagram provides a quick overview of what is contained in the NIEM Starter Model.

In the top-left corner is a Schema Composer profile artifact. You can double-click on this artifact to open the Schema Composer, pre-configured to create a NIEM subset schema.

The other diagrams in the Starter Model, can be opened using the buttons to the right. Each of these diagrams focus in on just one aspect of the IEPD model, giving you room to add additional items without causing the diagram to become cluttered.




NIEM的UML配置文件


Enterprise Architect与 NIEM 的UML配置文件（支持NIEM NIEM和 3）集成，以及许多模型模式，以帮助您开始为您的NIEM项目建模。

该配置文件定义了一组用于构建NIEM模型的构造型。它还定义了三种不同的图表类型：模型包描述（MPD）图表、平台独立模型（PIM）图表和平台特定模型（PSM）图表。这些图表类型中的每一个都有相应的图表工具箱页面，您可以从中选择要添加到模型中的项目，方法是将它们拖放到图表上。


访问

使用此处工具箱的任何方法来显示图表，然后单击使用以显示  工具箱“对话框并指定 ‘NIEM 3.0 MPD’（或‘PIM’或‘PSM’）。

当您打开该类型的图表时，对应于特定图表类型的图表工具箱就会变为活动状态。但是，您也可以随时使用此方法访问任何图表工具箱：

- 在  工具箱图表 查找工具箱项“对话框并指定 <profile> <toolbox>”

要将工具箱重置为当前图表的默认类型，只需关闭然后重新打开图表即可。

功能区	设计>图表>工具箱
键盘快捷键	Ctrl+Shift+3
其它	单击图形图表的  图标以显示图表工具箱





图表工具箱

NIEM图表提供对特定类型图表中常用元素和连接器的快速访问。

图表工具箱分为多个单独的页面：模型模式、关系、文件用途类型和架构文档用途。PIM 和 PSM 图共享一个共同的工具箱页面，并且每个都有自己特定的工具箱页面。

公共工具箱项

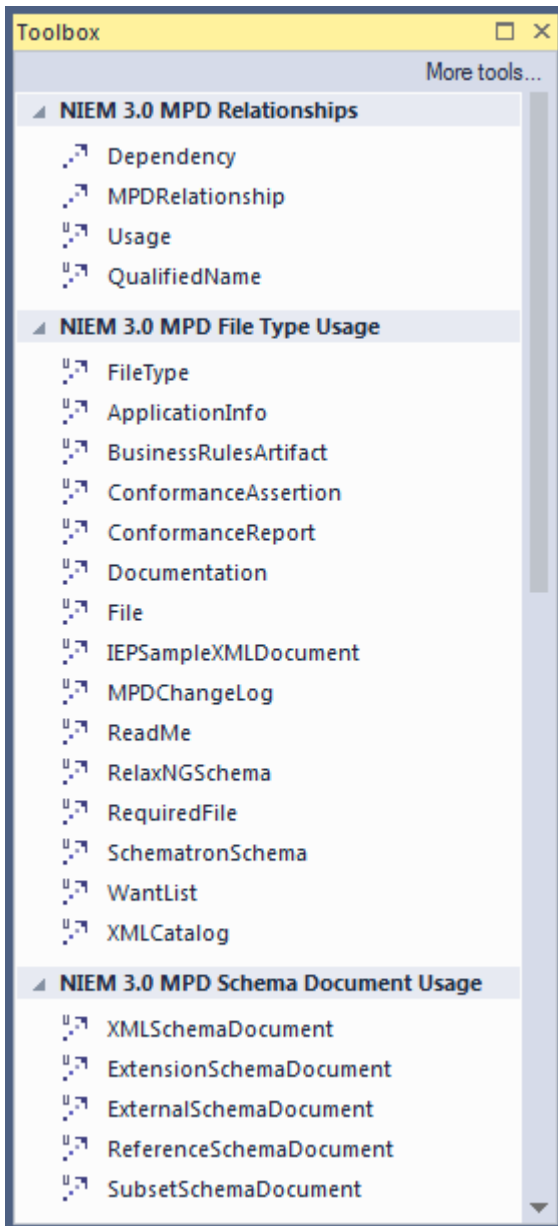
NIEM公共配置文件由在NIEM PIM配置文件和NIEM PSM配置文件中使用的构造型组成。

图标	描述
	NIEM适配器类型是A NIEM object类型，它调整外部组件以在NIEM中使用。
	NIEM关联类型A对象之间建立关系，以及该关系的属性。
	NIEM关联类型A对象之间建立关系，以及该关系的属性。
	NIEM扩充类型是A复杂类型，它提供可重用的数据块，可以添加到object类型或关联类型。

选择	A选择类将一组其值互斥的属性分组。
文档	文档注释是拥有它的元素A数据定义。
概括	A UML概括
列表	List 是A DataType，其值由另一个 DataType 的值的有限长度（可能为空）序列组成，该 DataType 是 List 的项类型。
本地词汇	本地词汇定义了一组特定领域的术语或缩写，然后可以在NIEM名称和定义中使用。
本地术语	LocalTerm 原型定义了 LocalVocabulary 中使用的特定领域的单词、短语、首字母缩写词或其他string。
元数据应用	«MetadataApplication» 原型适用于 «MetadataType»类与另一个 «MetadataType»类或属性之间的用途。它表示对NIEM «MetadataType» 的约束，将NIEM «MetadataType» 的应用程序限制为特定的模式类型或模式元素。
元数据类型	NIEM元数据类型描述有关数据A数据，即不描述对象及其关系但描述数据本身的信息。
命名空间	命名空间包表示由目标命名空间URI标识A NIEM命名空间。
NIEM类型	A是表示NIEM复杂类型（即可以具有属性结构的类型）的特定语义类型之一的类。NIEMType 是抽象的。
ObjectType	NIEM object类型代表某种object：具有自己生命周期且存在A事物。
原始类型	NIEM原始类型库定义了一组预定义的UML原始类型，用于 NIEM-UML 模型。为了确保在 PIM 级别使用的类型系统与NIEM兼容模式生成的完整性和一致性，该库中的原始类型基于 XML 模式原始类型。
属性	
财产持有人	PropertyHolder 是A持有全局属性的类，它不是任何特定NIEM类型的主题。这种类型的属性声明定义了属性的object类型，属性限制它的使用特定类型的主题。
参考	引用构造型适用于属性、类或包之间的实现。它允许通过属性另一个类中的属性来定义一个类中的属性。
表示	NIEM表示模式，允许一个类型包含一个表示元素，并且该元素类型的各种表示都在该表示元素的替换组中。
限制	Restriction Realization 表示两种类型定义之间的关系：第一种是通过第二种定义A限制导出的。
联盟	Union 是A DataType，其值空间是一个或多个其他 DataTypes 的并集，这些 DataTypes 是 Union 的成员类型。
联盟	UnionOf 构造型应用于用途依赖关系，其客户端必须是 Union DataType，而

	提供者必须是表示合法联合成员类型的 <code>DataType</code> 。A 依赖项指定供应商 <code>DataType</code> 是客户端 Union 的成员类型。
价值限制	

NIEM 3.0 MPD 工具箱



MPD 工具箱项

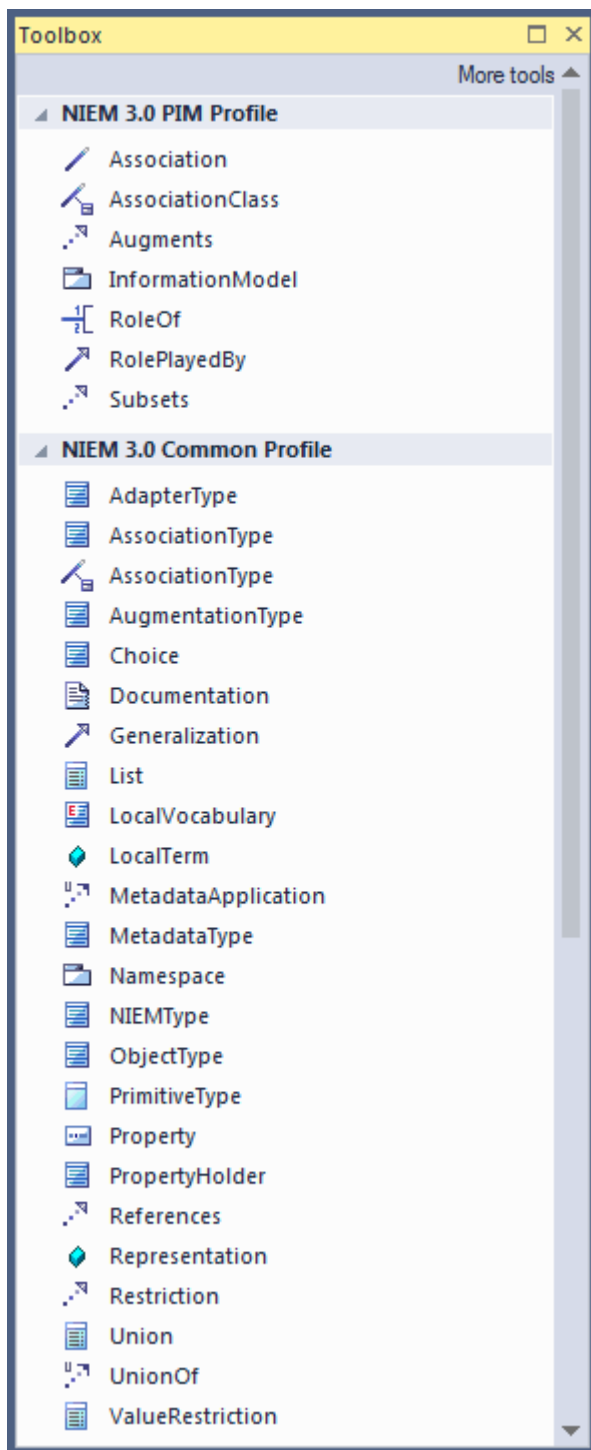
模型包描述配置文件包含用于模型 NIEM MPD 的构造型和工件。

图标	描述
----	----

关系	
依赖	UML依赖关系A
MPD关系	关系原型适用于表示 MPD 之间或 MPD 与另一个资源（例如NIEM规范；如符合的情况）之间的关系的依赖项。
用途	A UML使用关系
合格名称	<p><<qualifiedName>>用途连接器用于指定 IEP 的文档元素。</p> <p>在UML中识别 IEP 的文档元素：</p> <ul style="list-style-type: none"> 将实例 IEPConformanceTargetType 添加到 ModelPackageDescription 的 ModelPackageDeConformanceTarget 槽的工件实例 将 QualifiedNamesType 实例添加到新 IEPConformanceTargetType 实例的 ValidityConstraintWithContext 槽 添加具有应用构造型«qualifiedName»的用途，其中客户端是新的用途属性实例，供应商是表示 XSD元素的
文件类型用途	
文件类型	<<用于描述用途文件类型的连接器是一种数据工件。它也是许多其他 <<FileType用途>>连接器派生的基本类型。
应用信息	<<ApplicationInfo>> 连接器扩展了 <<FileType>> 使用连接器。它指定了一个工件工具，用于软件工具使用（例如，输入和输出）。
商业规则神器	<<BusinessRulesArtifact>> 连接器扩展了 <<FileType>> 使用连接器。它指定了一个工件，用于包含业务规则和约束的交换内容。
一致性断言	<<ConformanceAssertion>> 连接器扩展了 <<FileType>> 使用连接器。它指定一个工件用于表示一个 NI IEPD 符合NIEM的声明。
一致性报告	<<ConformanceReport>> 连接器扩展了 <<FileType>> 使用连接器。它指定了一个 MPD 软件和其他由工件感知工具自动生成或手动准备的软件，用于检查 NIEM符合性质量并呈现详细的结果报告。
文档	<<Documentation>> 连接器扩展了 <<FileType>> 使用连接器。它指定 MPD 用于工件文件的一种形式。
文件	<<File>> 连接器扩展了 <<FileType>> 使用连接器。它是用于指定工件的通用电子文件；存储在计算机系统上的文件。
IEPSampleXML 文档	<<IEPSampleXMLDocument>> 连接器扩展了 <<FileType>> 使用连接器。它用于指定示例 MPD 文档或工件实例。
MPDChangeLog	<<MPDChangeLog>> 连接器扩展了 <<FileType>> 使用连接器。它指定用于包含 MP 更改记录的工件。
自述文件	<<ReadMe>> 连接器扩展了 <<FileType>> 使用连接器。它用于指定工件读取。
松弛NGSchema	<<RelaxNG>> 连接器扩展了 <<FileType>> 使用连接器。它用于指定一个

	RelaxNG 模式。
所需文件	<<RequiredFile>> 连接器扩展了 <<FileType>> 使用连接器。它是用于与 MPD 文件工件的，而不应指定另一个工件于
SchematronSchema	<<SchematronSchema>> 连接器扩展了 <<FileType>> 使用连接器。它用于指定 Schematron 模式文档。
想要列表	<<WantList>> 连接器扩展了 <<FileType>> 使用连接器。它指定一个 MPD 架构子集，用于表示一个 NIEM 架构子集，并用作 NIEM 工件的导入或导出
XML 目录	<<XMLCatalog>> 连接器扩展了 <<FileType>> 使用连接器。它指定了一个工件用于 OASIS XML 目录。
架构文档用途	
XMLSchema 文档	<<XMLSchemaDocument>> 连接器扩展了 <<FileType>> 使用连接器。它指定了一个工件，该 MPD 不是 XML 模式文档，XSD 不一定是 NIEM 子集，扩展或引用模式)。
扩展架构文档	<<ExtensionSchemaDocument>> 连接器扩展了 <<XMLSchemaDocument>> 使用连接器。它用于指定一个工件扩展架构文档，该文档是一个 NIEM 扩展架构文档。
外部架构文档	<<ExternalSchemaDocument>> 连接器扩展了 <<XMLSchemaDocument>> 使用连接器。它指定一个工件用于外部的一个架构文档，用于 NIEM
参考架构文档	<<ReferenceSchemaDocument>> 连接器扩展了 <<XMLSchemaDocument>> 使用连接器。它指定了一个工件，用于从参考架构文档 (参考架构文档、发布、域更新或核心更新)。
子集架构文档	<<SubsetSchemaDocument>> 连接器扩展了 <<XMLSchemaDocument>> 使用连接器。它指定了一个工件，它是一个子集模式文档。

NIEM 3.0 PIM 工具箱



PIM工具箱项

NIEM PIM配置文件包含用于NIEM PIM 但不用于NIEM PSM 的构造型。

图标	描述
关联	UML关联A
关联类	A UML关联类。

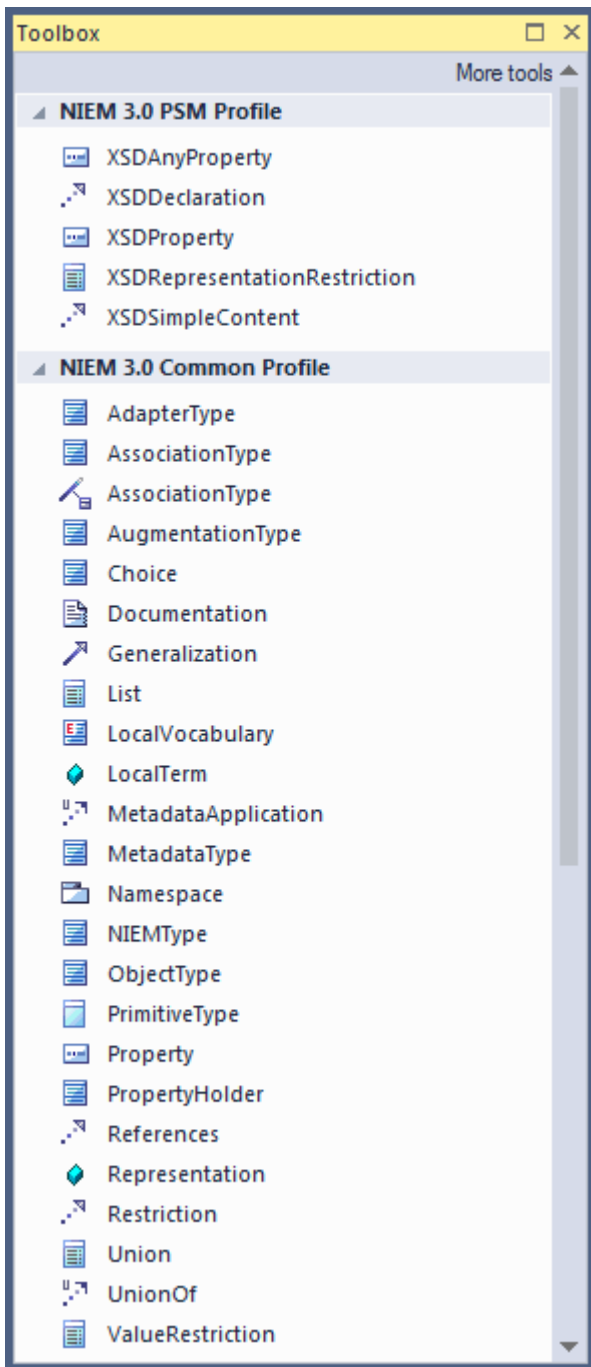
增强	A原型的实现连接器，用于指定一个类（供应商）增强另一个类（客户端）。
信息模型	InformationModel 是一个原型包，它为在NIEM消息中交换的信息结构提供了与平台无关的视角。它表示NIEM命名空间，但也可以指定默认用途，例如子集、交换或扩展。
的角色	RoleOf 构造型应用于 AssociationEnd 以指定关联属性的角色类型。
角色扮演者	A原型概括实例扮演的角色连接器指定一般类的角色是特殊类的类型。
子集	Subsets 连接器是一个原型实现，它指定子集客户端（派生元素）和它的参考供应商（基本元素）之间的子集关系。

PSM工具箱项

NIEM PSM配置文件包含在NIEM PSM 中使用的构造型。这些构造型不需要与NIEM PIM 一起使用，但它们可以用于提供额外的特定于平台的标记。

图标	描述
XSDAnyProperty	属性构造型表示其类型不受限制的属性，该类型在XML Schema中实现为xs:any 粒子。
XSD声明	XSDDeclaration 原型是公共引用原型的特化。
XSD 属性	属性表示NIEM属性，它在XML Schema中实现为属性声明和使用或元素声明和粒子
XSDRepresentationRestriction	XSDRepresentationRestriction 指定对基本 DataType 的值在 XML 模式中的表示形式的限制。
XSDS简单内容	«XSDSimpleContent» 构造型表示两个类型定义之间的关系：第一个是具有简单内容的复杂类型定义，第二个是简单类型。

NIEM 3.0 PSM工具箱



下载NIEM参考模型

NIEM 5参考模型是NIEM 5 Release包XSD文件内容的UML表示。

它包含代表 NIEM 核心的包，以及NIEM 5 版本中包含的各种域模式、它们的相关代码列表和其他相关包。NIEM 5 参考模型可从Sparx Systems可重用资产服务器下载到您的Enterprise Architect项目中。

早期版本的NIEM参考模型也可从Sparx Systems可重用资产服务器下载。

访问

使用此处概述的任何方法显示模型构建器。

在模型生成器对话框中，选择蓝图 信息交换 > NIEM”，然后选择 “NIEM 3、4 和 5”。

根据需要选择参考模型、MPD 类型和入门模型。

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)

创建NIEM IEPD

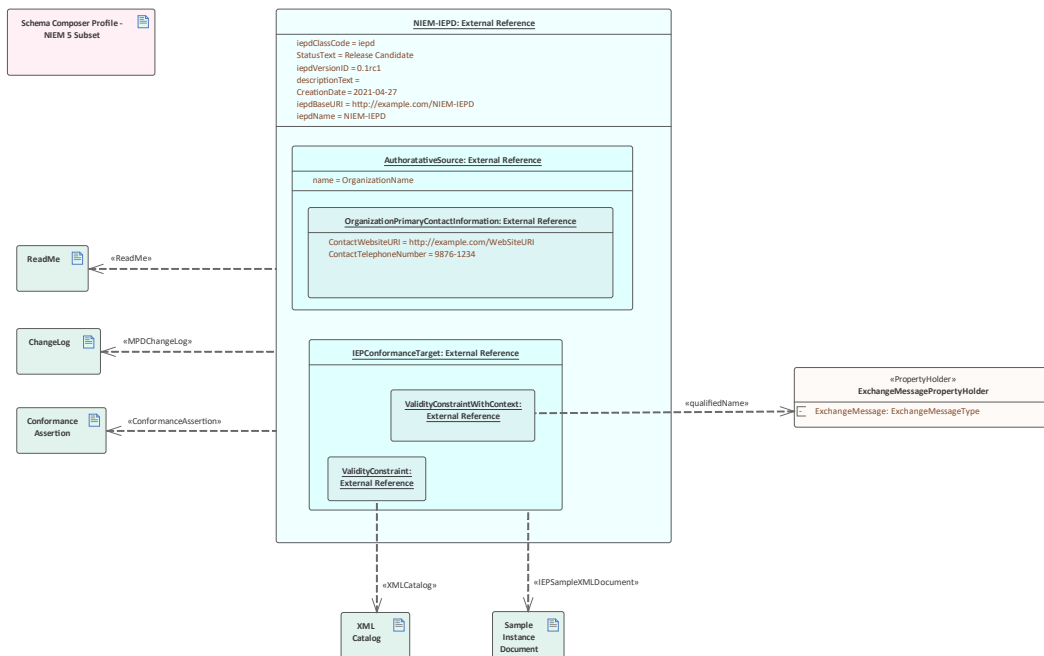
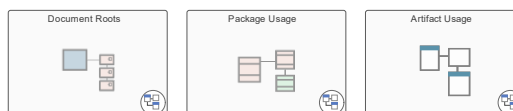
Enterprise Architect的NIEM配置文件提供了一个基本的 IEPD模型作为起点，您可以从此构建自己的 IEPD模型。您可以使用模型生成器将 IEPD 启动模型添加到您的项目中。

IEPD Overview

This IEPD Overview diagram provides a quick overview of what is contained in the NIEM Starter Model.

In the top-left corner is a Schema Composer profile artifact. You can double-click on this artifact to open the Schema Composer, pre-configured to create a NIEM subset schema.

The other diagrams in the Starter Model, can be opened using the buttons to the right. Each of these diagrams focus in on just one aspect of the IEPD model, giving you room to add additional items without causing the diagram to become cluttered.



IEPD 入门模型模式，可从模型构建器获得。

本主题概述了在Enterprise Architect中创建新的NIEM IEPD模型以及从该模型生成 IEPD 所需的步骤。

创建NIEM IEPD模型并生成NIEM IEPD

步骤	描述
创建一个新的Enterprise Architect项目	<p>启动Enterprise Architect并创建一个新项目。</p> <p>从功能区中选择 设计>包>模型构建器”。</p> <p>在模型生成器对话框中，单击 <视角名称> 按钮并选择 信息交换 NIEM ”并展开 “NIEM 3、4 和 5” 蓝图。</p> <p>您的NIEM项目必须包含NIEM IEPD 类型和至少一个NIEM参考模型。</p> <p>选择 “NIEM 5.0参考模型”以及“ NIEM 5 IEPD 类型”。</p> <p>点击创建模型按钮下载并将选定的模型导入到您的项目中。</p> <p>模型生成器中还提供了基本NIEM IEPD 的模型模式。这旨在作为您的NIEM项目的起点。</p> <p>或者，选择 “NIEM 5 IEPD Starter模型”，然后单击 创建模型按钮。</p>

创建 IEPD模型	<p>如果您在上一步中选择不包含 IEPD模型，您现在可以创建自己的模型。</p> <p>在浏览器窗口中，创建一个新包或（视图节点）来保存您的 IEPD模型。</p> <p>在新包中，创建一个NIEM IEPD 图。</p> <p>您可以将NIEM IEPD Types包中可用类型的实例添加到您的图表（以及您的 IEPD模型）中，方法是将其拖到您的 IEPD 图表上。</p> <p>使用浏览器窗口找到您需要的类类型，然后在将元素拖到图表上的位置时按 Ctrl。系统提示您选择一个动作；任何一个：</p> <ul style="list-style-type: none"> • 在图表上放置类的链接，或 • 创建并添加类的新实例规范 <p>对于 IEPD模型，您通常会使用物件实例。</p> <p>首先，创建 IEPD类的object实例。（你很快就会看到，你需要一个 IEPD 类的实例来驱动你的 IEPD 的生成。）</p>
自定义您的模型	<p>IEPD 类的实例携带识别 IEPD 的信息。它与各种其他模型工件之间的链接决定了在生成模式文件和目录文件时生成什么（以及生成到哪里）。</p> <p>无论您选择下载 IEPD Starter模型，还是通过从浏览器窗口拖动实例来创建您自己的 IEPD模型，您都必须设置适合您正在创建的模型的物件实例的属性值。这是通过设置 IEPD模型中使用的各种object实例的运行状态属性来实现的。</p> <p>有关为 IEPD 实例对象设置值的详细信息，请参阅帮助主题自定义您的帮助模型</p>
创建您的数据模型	<p>这是您模型将在您的信息交换消息中发送的数据。</p> <p>在NIEM中，这通常在具有 <<InformationModel>> 构造型的包内建模，表示模型中使用的不同名称空间。这些包通常包括一个 NIEM-core包，它是 NIEM-core参考模型包的子集，以及两个扩展包，它们扩展了 NIEM-core 中可用的内容，其中一个代表交换消息。</p> <p>您的项目可能还需要其他NIEM模式的子集，例如来自 Biometrics 或 EmergencyManagement 域的那些。</p> <p>有关创建数据模型的更多信息，请参阅帮助主题创建NIEM数据模型帮助使用 Schema 编辑器对NIEM进行子集化。</p>
生成IEPD	<p>在您从中生成 IEPD 之前，您的NIEM模型不必完成。</p> <p>生成 IEPD 可以被认为是一个迭代过程。在完成 IEPD 和定义一致性目标之前，您可能只生成命名空间模式。在定义信息模型之前，您可能会使用完整描述的 IEPD 实例和一致性目标来生成。您可以继续更新您的模型并生成您认为合适的 IEPD。</p> <p>要生成您的 IEPD，请在图表或浏览器窗口中选择 IEPD 实例规范。转到“特定”功能区并选择选项 技术> NIEM >生成NIEM架构”。</p> <p>生成NIEM IEPD Schemas 窗口打开。</p> <p>此窗口列出了模型中使用的命名空间模式，您可以选择生成其中的哪些。您还可以选择在生成中包含哪些NIEM基础架构模式。</p> <p>在此窗口中，您还可以设置生成输出文件的根目录。</p> <p>选择并指定输出文件夹后，单击 生成”按钮开始生成 IEPD。</p> <p>有关生成NIEM IEPD Schemas 窗口的详细信息，请参阅帮助主题 NIEM帮助 NIEM。</p>

注记

- 所有包含NIEM模型的项目都必须包含通过模型生成器下载的NIEM IEPD 类型包；IEPD 实例是您的NIEM模型的核心实例和与 <<InformationModel>>包的关系以及其他工件用于驱动 IEPD 生成；如果模型中没有 IEPD 实例，您将无法生成 IEPD
- 通常，您必须至少将一个NIEM参考模型导入到您的项目中；参考模型包含 NIEM 核心参考模式的UML表示，以及许多特定于域的参考模式，如果您打算使用Enterprise Architect的Schema 编辑器创建子集模式，则必须在您的项目中提供这些模式

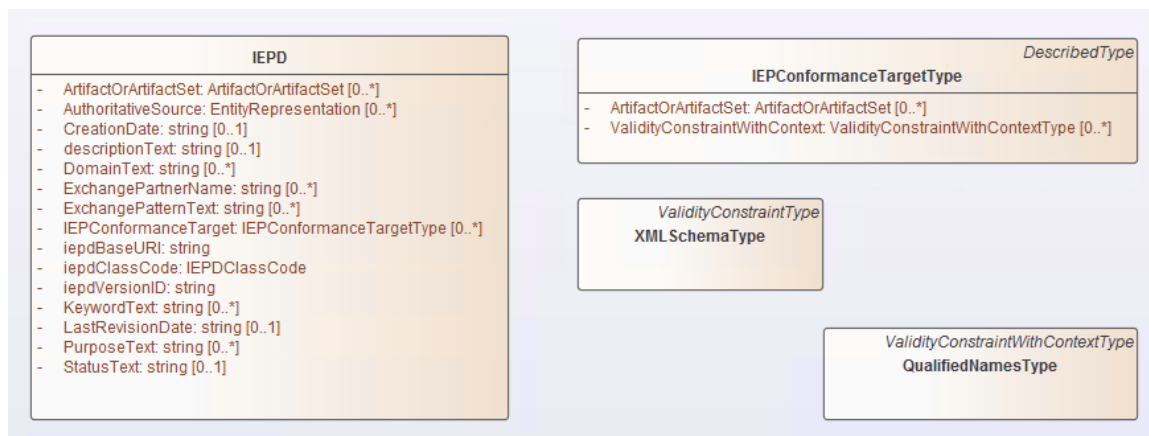
自定义您的 IEPD模型

无论您选择下载 IEPD Starter模型，还是通过从浏览器窗口拖动实例来创建自己的 IEPD模型，您都必须设置适合您正在创建的模型的物件实例的属性值。这是通过设置 IEPD模型中使用的各种object实例的运行状态属性来实现的。

NIEM IEPD 类型

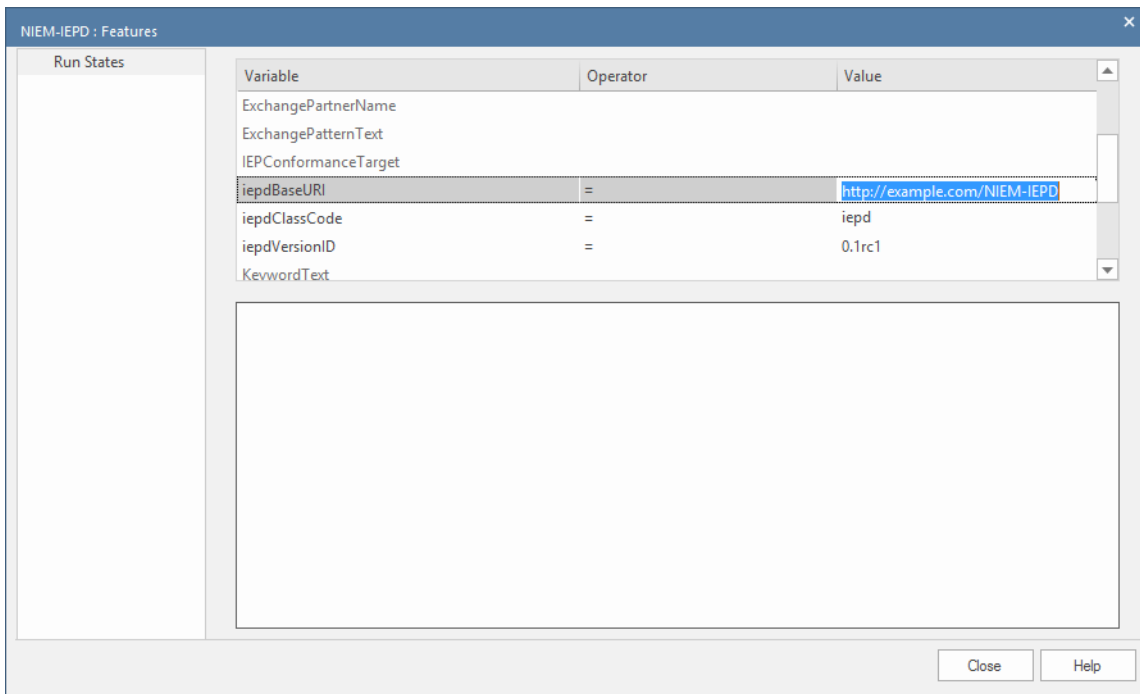
包' NIEM IEPD Types'包含类'IEPD'的定义，以及许多其他类。这些其他类被引用为'IEPD'类属性的分类器。在这个包中定义的各种类之间的关系可以在图表'NIEM-UML IEPD Types'上查看。

'IEPD'类有许多简单string类型的属性，以及一些按在' NIEM IEPD Types'包中定义的类型分类的属性。



设置 IEPD 对象的运行状态

在运行object中为简单类型的属性设置运行状态值时，可以使用“设置运行状态”命令。这可以通过右键单击物件上的物件，然后选择“特征”来访问。设置运行状态...”（或按 Ctrl+Shift+R）。

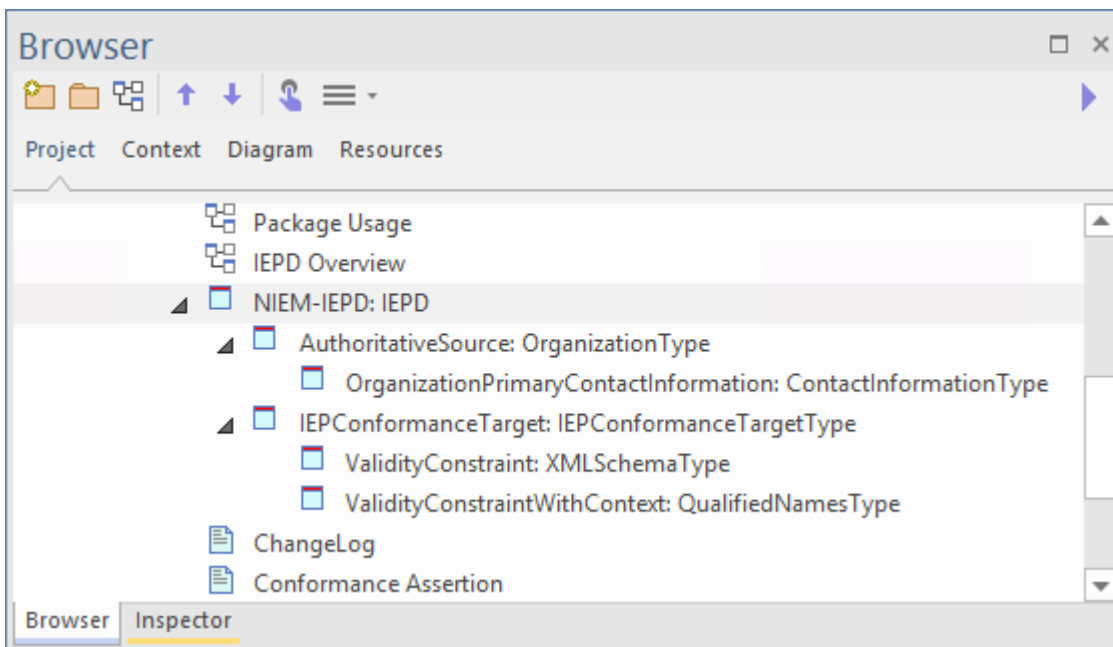


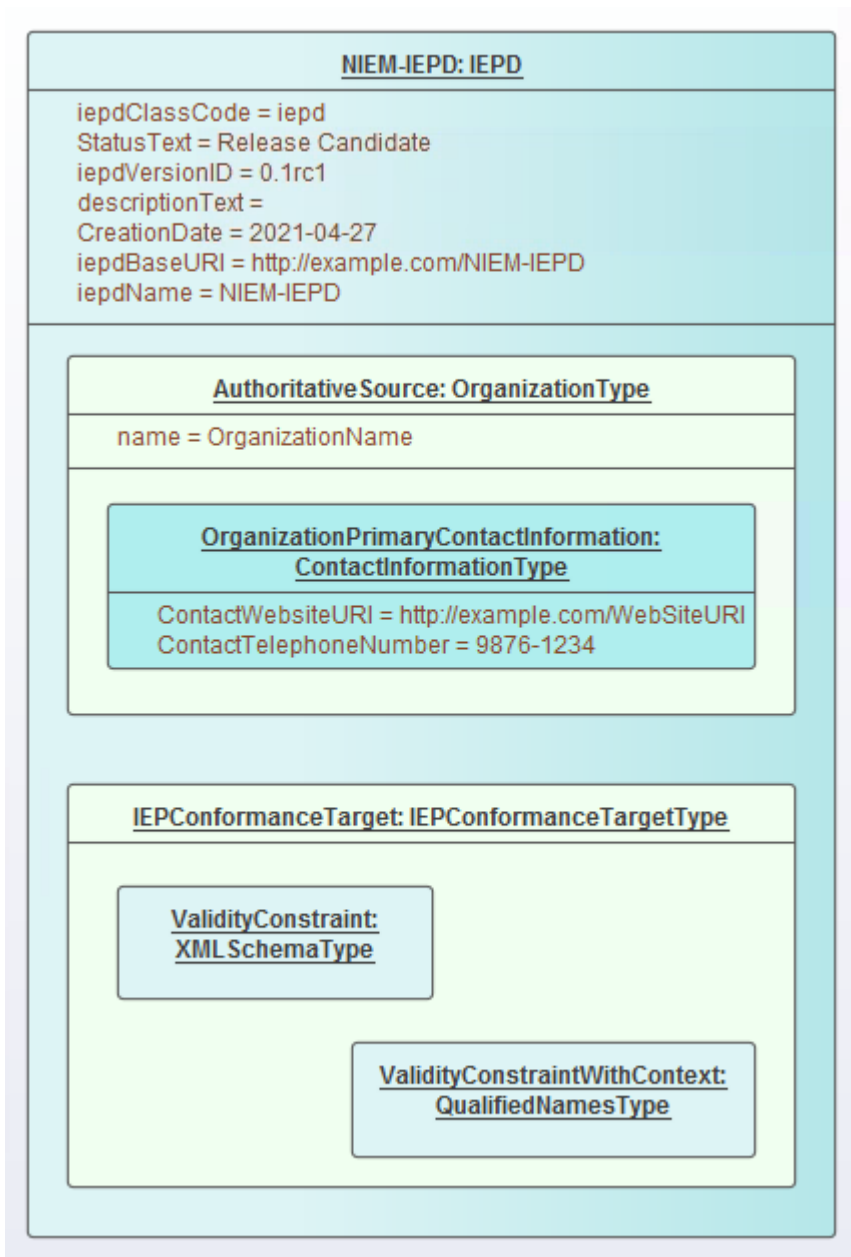
在属性引用其他类作为其类型的情况下，不能简单地输入运行状态值。

Enterprise Architect支持两种为这些属性指定值的方法，每种方法都需要创建被引用类的物件实例。

您属性为属性设置的属性对象关联两个嵌套的物件物件作为子物件并使用正在设置的属性的名称命名子物件。关联物件时，物件的名称并不重要，但角色名称必须与设置的属性名称相匹配。

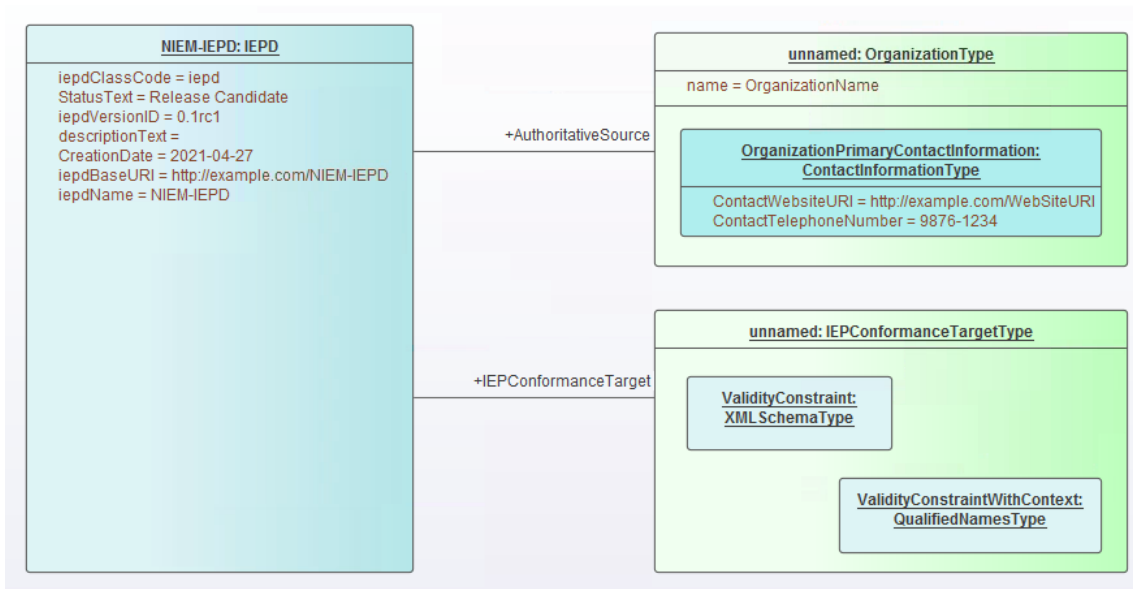
例如，您可以在 IEPD 中创建类型为物件的物件和嵌套的物件。在这种情况下，子物件必须命名为“IEPConformanceTarget”，以与类定义中该名称的属性相对应。通过检查浏览器窗口中显示的层次结构，确保子物件确实嵌套在父级中。





如果在关联上使用角色名称，则将“属性”物件创建为单独的（非嵌套）物件实例，然后创建从“所有者”到“object”的关联，最后为目标物件指定角色名称。例如，创建从模型包描述物件到关联物件实例的关联。打开关联的“属性”对话框并将目标的角色命名为“IEPConformanceTarget”，以对应于“IEPD”类中的属性名称。同样，在这种情况下，物件本身的名称并不重要，它甚至可以是匿名的，但角色名称必须与您正在设置其值的属性的名称匹配。

注册一个 IEPD 物件可能指定许多 IEPConformanceTargets。您必须为每个实例创建一个物件实例，并且每个实例都必须命名为“IEPConformanceTarget”。

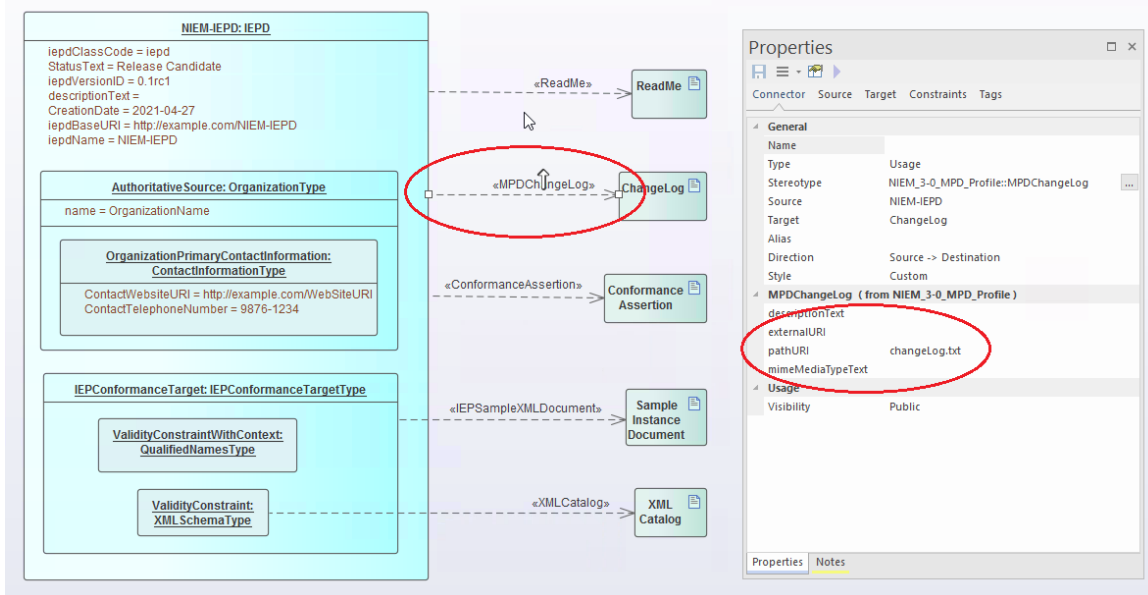


然后可以使用这些技术中的任何一种来设置属性物件中的属性。例如，要设置物件实例的类（可能是 QualifiedType 类型的实例）以及它的嵌套名称，为派生的 ValidityNamestraintType 属性和名称和角色关联。

建模文件用途

使用文件类型可以通过将使用文件类型与用途中所需的连接器添加到工件中来建模。

使用相关用途连接器的属性中指定的值，将各种文件名生成到 IEPD 的 XML 目录文件中。



NIEM IEPD 生成

生成 IEPD 可以被认为是一个迭代过程。您不必等到您的NIEM模型完成后才能从中生成 IEPD。

您的NIEM 'IEPD Overview' 图应包含 IEPD 的实例规范。IEPD 实例及其与一致性目标实例以及其他工件的关系是 IEPD 目录的表示。当您从您的模型生成 IEPD 时，Enterprise Architect将根据您的 IEPD模型中的项目生成目录文件和其他工件。它还将为您的模型引用的 <<InformationModel>> 包生成NIEM模式。结果将是一组文件输出到您为生成过程指定的目录中。

生成 IEPD 的步骤

节	行动
1	您的NIEM IEPD 图应包含 IEPD 的实例规范。 在图表或浏览器窗口中选择 IEPD 实例。
2	从 特定”功能区中，选择 技术> NIEM >生成NIEM架构”选项。 将显示 生成NIEM IEPD 模式”对话框。
3	在 目录”字段中，键入或浏览生成 IEPD 的目录路径。
4	“NIEM版本”字段默认为 5.0”。如果生成NIEM 3 或 4 IEPD，请将此字段设置为适当的值。 '工件' 面板列出了此模型工件静态目录中使用的工件和静态目录，每个都有其相对路径。 选择或取消选择这些项目旁边的复选框以生成或跳过这些项目。 “命名空间架构”面板显示将为信息模型生成的架构文件。 选择或取消选择命名空间架构旁边的复选框以生成或跳过该架构。 选择一个架构命名空间架构以显示该架构的包详细信息。
5	点击生成按钮。 生成成功完成后，单击查看架构按钮打开视图，显示用于生成的输出目录的内容窗口如果当前的架构工件将被取消选择，则单击视图按钮打开一个编辑器以查看与所选架构命名空间包关联的架构。

注记

- 为包生成的模式文件的输出位置由将包与 IEPD 实例规范相关联的使用连接器上的 用途”标记值指定；默认值由Schema 编辑器在创建子集包时设置，但用户可以覆盖这些值

创建NIEM数据模型

NIEM背后的基本原则之一是重用通用参考词汇表——一组预定义的数据元素和定义，用于定义信息交换。为此，构建NIEM数据模型的核心任务之一是创建NIEM参考模式的子集。目标是通过重用已在NIEM参考模型中定义的类型和元素来模型尽可能多的数据交换。

A NIEM数据模型通常由多个应用了 <<InformationModel>> 原型的包组成。

通常，一个模型将有一个包表示 NIEM 核心子集模式，一些其他包表示特定域模式的子集，以及一个或多个包表示扩展模式。扩展模式包提供模型所需的那些元素，这些元素不能从NIEM参考模型中获得。通常，交换消息的根元素与更多的一般元素分离，并建模在专用于特定交换的扩展模式包中。

创建NIEM数据模型的步骤

节	细节
导入NIEM参考模型	<p>创建NIEM模型所涉及的许多活动都依赖于使用NIEM参考模型。如果您还没有这样做，请在继续之前将参考模型导入您的Enterprise Architect项目。</p> <p>有关详细信息，请参阅帮助主题下载帮助Download the NIEM Reference Model</p>
创建 NIEM-core参考包的子集	<p>在创建NIEM IEPD 时创建NIEM命名空间模式子集的原因有很多，但最重要的两个原因是：</p> <ul style="list-style-type: none"> 参考模式非常大；子集生成的模式文件要小得多，这反过来又会导致更快地验证模式 参考模式中的元素受到非常松散的约束；子集化过程允许建模者施加更严格的约束，例如限制基数和允许值，以更准确地反映实际业务需求 <p>在Enterprise Architect中，子集化过程是使用Schema 编辑器执行的。</p> <p>Schema 编辑器允许建模者从源包中选择所需类的子集，并为每个选择的类选择所需属性的子集。然后将选定的类及其减少的属性集复制到目标包。大多数情况下，源包是来自NIEM参考模型的 NIEM-core 命名空间包。在这种情况下，目标包也将是一个名为“NIEM-core”的命名空间包，但它将是您的NIEM IEPD模型的一部分。</p> <p>参考模型中的其他其它空间包，例如 Domain包，也可以以相同的方式进行子集化。</p> <p>使用Enterprise Architect的Schema 编辑器工具将 NIEM-core 参考包的子集复制到 IEPD模型中的 NIEM-core 子集包。目标是通过重用已在 NIEM-core参考模型中定义的类型和元素来模型尽可能多的数据交换。</p> <p>如果您的模型也将使用NIEM域包，则应为您使用的每个域包重复此子设置过程。</p> <p>有关更多信息，请参阅帮助主题帮助<i>NIEM with the Schema</i> 编辑器。</p>
创建扩展包	<p>创建NIEM数据模型时，目标是使用NIEM参考模型中的类型和元素对尽可能多的数据交换进行模型。无法通过重用现有NIEM元素进行建模的内容随后将在“扩展”命名空间包中进行建模，方法是使用 NIEM-UML 配置文件中的元素创建新的类型和元素，所有类型最终都源自 XML 模式原始类型。</p> <p>NIEM Starter模型（来自模型构建器）和 IEPD Starter模型模式（来自图表工具箱）都提供了 <<InformationModel>>包，可以在其中对各种模式模型。使用这些包中的 PIM 图，您可以通过添加图表工具箱中的元素来构建不同模式的模型。</p> <p>建议您使用“exchange”包中的图表，根据需要来自其他架构包类型和元素来组装交换的高级模型。</p> <p>大多数 IEPD 需要扩展模式来定义特定类型和属性，这些类型和属性对于所定</p>

义的数据交换是唯一的。但是， NIEM模型没有定义用于组装交换中的所有对象的特定消息类型或结构。因此， IEPD 的创建者需要编写一个扩展模式来声明消息的根元素和基本结构。交换的根元素将交换中定义的所有对象和关联汇集在一起。

虽然您不需要创建单独的架构来声明消息的根元素和基本结构，但将特定于消息的扩展分离到“交换”架构中并将更通用的扩展分离到“扩展”架构中可能会有所帮助。交换架构包含特定于消息类型或消息类型组的定义。这通常仅包括根元素及其类型，以及可能构成消息基本框架的一些结构元素。

将模式元素组织成“交换”和通用“扩展”分组还可以提供在多个 IEPD 之间共享更通用模式的可能性，而“交换”模式通常特定于一个特定的 IEPD。您还可以拥有多个“交换”模式，以表示不同的消息类型或不同消息类型的组。

使用Schema 编辑器对NIEM进行子集化

Enterprise Architect的Schema 编辑器是一个可以大大简化从NIEM模型参考命名空间包创建子集的过程的工具。

访问

使用此处列出的任何一种方法来显示Schema 编辑器模型使用对话框，
输入新模型转换的名称，然后从 架构集”下拉列表中选择 国家信息交换模型 (NIEM) ”。
在您的项目中保存一个合适的工件模型的配置文件 (在您的工件包很容易找到合适的根包)。

功能区	开发 > 架构建模 > Schema 编辑器 > 打开Schema 编辑器：新建 > 模型转换
-----	---

创建子集模型

NIEM专家建议第一步是为您的 XML 交换创建一个UML模型，因为它允许您捕获业务需求，而不会受到NIEM中的操作方式的过度影响。

一旦您有了用于交换的UML模型的初稿，您就可以开始使用NIEM重新创建该模型。

最初，在NIEM参考模型中找到合适的类型和属性似乎是一项不可能完成的任务。随着您获得经验并熟悉NIEM模型的内容，这将变得更加容易。

您通常使用的大多数NIEM类型，例如 PersonType、OrganizationType、 DocumentType或 ActivityType，都有许多属性，您通常只需要其中的几个。这是子集变得有用的地方。

如果您尝试使用他们的姓名、地址和生日来模型一个人，您可以从 NIEM-core 中选择 PersonType 和 AddressType。从这些类型中，仅选择您的模型所需的属性。

如果所选属性引用了其他类型，这些类型将自动添加到Schema 编辑器中。

当您 生成”您的子集时， Enterprise Architect创建子集所需的目标模式包，然后将所选类型及其缩减的属性集复制到目标包中。

进一步完善您的子集

创建子集后，您可以通过调整类型内属性的基数或限制属性的允许值来进一步属性它。

要调整基数或限制属性的允许值，请在Schema 编辑器的中央属性中选择该属性，然后右键单击并选择 限制此属性”。将显示 属性限制”对话框，您可以在其中调整基数或根据需要对属性应用限制。

单击 更新”以保存对模型转换配置文件的更改，然后单击 生成”以重新生成具有应用限制的子集模型。

NIEM子集化通常是一个迭代过程。使用保存的模型转换配置文件，您可以在 IEPD 开发的各个阶段根据需要重新加载、更新和重新生成子集。

使用Schema 编辑器对NIEM进行子集化

节	行动

1	打开Schema 编辑器。(见访问:功能区)
2	<p>创建一个新的Schema 编辑器配置文件。</p> <p>单击新建按钮并选择“模型转换”。</p> <p>在打开的对话框中,指定配置文件的名称并在“架构集”字段中选择“NIEM”。</p> <p>(此对话框中的“命名空间”字段不用“NIEM”,因为NIEM使用其模型包上的标记值来指定命名空间。)</p> <p>选择一个位置来保存您的新个人资料,然后单击确定按钮。</p>
3	<p>在浏览器窗口中,在参考模型的NIEM-core包中找到所需的PersonType和AddressType类型。</p> <p>将所需类型从浏览器窗口拖放到Schema编辑器的“类”窗格中。</p>
4	<p>现在在Schema编辑器的“类”窗格中选择一种类型,比如PersonType。</p> <p>PersonType的完整属性列表显示在“属性”窗格中。</p>
5	<p>使用“属性”列表中的复选框来选择PersonType的属性以在您的交换模型中使用。在这种情况下,请选中PersonBirthDate和PersonName复选框。</p> <p>当您选择这些属性时,Schema编辑器会自动将类型“DateType”和“PersonNameType”添加到类列表中,因为这些类型被您刚刚选择的属性引用。</p>
6	<p>现在在“类”窗格中选择“日期类型”。</p> <p>'DateType'有四个属性,DateAccuracyAbstract、DateAugmentationPoint、DateMarginOfErrorDuration和DateRepresentation。这些属性中的前三个是日期元数据——它们不包含日期值。第四个,DateRepresentation,是一个抽象属性,所以它也不直接保存日期值。它用作最终保存日期值的属性的占位符。</p> <p>NIEM模型通常使用XML Schema抽象元素和替换组。</p> <p>抽象元素为子集的创建增加了一些复杂性,因为您需要添加抽象元素以及将替换抽象元素的那些元素。</p> <p>例如,大多数与日期相关的类型都包含抽象元素nc:DateRepresentation,可以用nc:Date、nc:DateTime等替代。</p>
7	<p>选择属性DateType.DateRepresentation。</p> <p>您会注意到另一种类型DateRepresentationPropertyHolder已添加到“类”列表中。</p>
8	<p>在“类”列表中选择DateRepresentationPropertyHolder。</p> <p>无类型属性DateRepresentation被称为替换组的“头”。必须在替代的客户端DateType以及替代的供应商DateRepresentationPropertyHolder中选择此属性。作为替换组首的属性是为您预先选择的,因此您只需在DateType中选择最终将替换为DateRepresentation的属性。选择属性Date:date - 它将用作实际保存数据值的DateRepresentation。</p> <p>在涉及替换组的情况下,简单地添加抽象元素而不添加来自相关PropertyHolder类型的可替换元素是一个常见的错误。</p>
9	<p>通过从PersonNameType类中选择PersonGivenName、PersonMiddleName和PersonSurName,对PersonName属性重复该过程。</p>
10	<p>要将您当前选择的类和属性保存到您正在创建的配置文件中,请单击更新按钮。</p> <p>这将使用您当前的选择更新配置文件,如果您需要对其执行进一步的工作,则可以在以后重新加载它。这有助于创建子集包的迭代过程。</p>
11	<p>现在单击“生成”选项。</p> <p>在“架构导出”对话框中选择“NIEM模型子集”,然后单击“生成”按钮。</p>

	导航到包含您正在构建的 IEPD 的包层次结构。选择将包含子包的父包，然后单击确定按钮。
12	您在Schema 编辑器中选择的类将被复制到目标包，仅包含您选择的属性子集。

注记

- 请通读每个演练示例 - 每个示例都包含重要信息
- 支持NIEM开发的Schema 编辑器功能，协助创建 Subset Schemas；它无助于产生扩展模式

浏览示例

如果您不熟悉使用NIEM的Schema 编辑器，请花时间阅读这些示例。每个示例都包含重要信息，有助于确保您的模型使用有效的NIEM子集，最终生成有效的 XML 模式文件。

示例1：添加类和选择属性

这个“演练”示例演示了如何使用Enterprise Architect的Schema 编辑器来执行添加类和选择要包含在NIEM子集包中的属性的基本操作。

节	描述
1	<p>打开一个包含NIEM 5.0参考模型和NIEM IEPD类型的Enterprise Architect项目。</p> <p>如果您没有这样的项目，则请打开一个新项目并使用模型生成器 (Ctrl + Shift + M) 加载参考模型和 IEPD 类型。</p>
2	<p>使用模型生成器，将NIEM 5 IEPD Starter模型的新副本添加到您的项目中。</p> <p>您应该将object实例“NIEM-IEPD”重命名为更有意义的名称。生成 IEPD 时，此object实例的名称用于命名创建 IEPD 的根文件夹。</p> <p>如果您愿意，也可以将包“NIEM 5 Starter模型”重命名为更合适的名称。</p>
3	<p>启动模型包含一个Schema 编辑器编辑器配置文件，名为“工件Schema 编辑器配置文件 - NIEM 5 子集”。找到这个工件，然后在浏览器窗口中双击它。这将打开Schema 编辑器并加载配置文件“Schema 编辑器配置文件 - NIEM 5 子集”。</p> <p>Schema 编辑器的下部包含三列。从左到右，它们被标记为“类”、“属性”和“架构”。</p>
4	<p>使用浏览器窗口，在NIEM 5.0参考模型中找到包“niem-core”。</p> <p>在那个包中，找到类'AircraftType'。</p> <p>将“AircraftType”拖放到Schema 编辑器的左侧列（标记为“类”）。</p> <p>您会注意到“ConveyanceType”和“ItemType”类被自动添加到类列表中。</p> <p>“ItemType”和“ConveyanceType”是派生“AircraftType”的超类型。</p>
5	<p>在“类”列中选择 AircraftType。</p> <p>您会注意到中心列“属性”显示了属于此类的属性的完整类。</p> <p>还列出了父类的属性。</p> <p>要在子集模式中包含一个属性，您只需在它旁边放置一个复选标记。</p> <p>（您应该只选择当前在“类”列表中的类的属性。</p> <p>如果您需要父类的属性，请选择类，然后选择其属性。）</p> <p>在 AircraftTailIdentification 旁边打勾。AircraftTailIdentification 的类型是 IdentificationType。</p> <p>请注意，IdentificationType 已添加到类列表中。</p> <p>Enterprise Architect自动将那些被引用为您选择的属性类型的分类器添加到“类”列表中。</p>
6	<p>在Schema 编辑器的左侧栏中选择类IdentificationType。</p> <p>在中心列中，在属性 IdentificationID 旁边放置一个复选标记。IdentificationID 的类型是“string”。</p> <p>'string' 类型是原始类型 - 它没有添加到类列表中。</p>
7	<p>现在，选择Schema 编辑器左侧栏中的类ConveyanceType。</p> <p>在属性 ConveyanceMotorizedIndicator 旁边放置一个复选标记。</p>

	'boolean' 类型是原始类型 - 它没有添加到类列表中。
8	<p>在Schema 编辑器的左侧栏中选择类ItemType。</p> <p>在属性 ItemMakeName、ItemModelName 和 ItemModelYearDate 旁边放置复选标记。</p> <p>类型 ProperNameTextType 和 TextType 会自动添加到类列表中。 TextType 是 ProperNameTextType 的基础类。</p>
9	<p>单击更新按钮将选定的类和属性保存到配置文件，然后单击生成按钮。</p> <p>在打开的窗口中，选择 'NIEM模型子集'，然后单击生成按钮。</p> <p>系统将提示您选择将在其中创建子模型的包。通常，您会选择作为 Exchange 架构包的父级的包。在 starter模型中，交换包名为 'IEPD-Exchange'，其父包名为 'NIEM 5模型'，尽管您可能在前面的步骤 2 中重命名了这些包。</p> <p>选择包'NIEM 5模型'，然后点击确定按钮。</p> <p>注记：当创建更复杂的模型时，您的子集可能包括来自多个不同 <<InformationModel>>包的类。Enterprise Architect的Schema 编辑器会自动创建所需的目标包，并将您正在设置子集的Classes 复制到标记值'targetNamespace' 与提取原始类的源包的标记值匹配的目标包中。子集 <<InformationModel>>包将被创建为您选择作为生成目标的包的子项。</p>
10	<p>一旦生成完成，展开目标<<InformationModel>>包。</p> <p>您将看到您选择的类及其减少的属性集。</p>

示例2：使用关联类型

这个“演练”示例演示了如何使用Enterprise Architect的Schema 编辑器将关联类型和它们引用的类型添加到您的NIEM子集包中。

节	描述
1	<p>打开一个包含NIEM 5.0参考模型和NIEM IEPD类型的Enterprise Architect项目。</p> <p>如果您没有这样的项目，那么请打开一个新项目并使用模型生成器 (Ctrl + Shift + M) 加载参考模型和 IEPD 类型。</p>
2	<p>使用模型生成器将NIEM 5 IEPD Starter模型的新副本添加到您的项目中。</p> <p>您应该将object实例“NIEM-IEPD”重命名为更有意义的名称。生成 IEPD 时，此object实例的名称用于命名创建 IEPD 的根文件夹。</p> <p>如果您愿意，也可以将包“NIEM 5 Starter模型”重命名为更合适的名称。</p>
3	<p>启动模型包含一个Schema 编辑器编辑器配置文件，名为 工件Schema 编辑器配置文件 - NIEM 5 子集”。找到这个工件，然后在浏览器窗口中双击它。这将打开Schema 编辑器并加载配置文件“ Schema 编辑器配置文件 - NIEM 5 子集”。</p> <p>Schema 编辑器的下部包含三列。从左到右，它们被标记为“类”、“属性”和“架构”。</p>
4	<p>使用浏览器窗口，在NIEM 5.0参考模型中找到包“niem-core”。</p> <p>在那个包中，找到类'PersonLocationAssociationType'。</p> <p>将“PersonLocationAssociationType”拖放到Schema 编辑器的左侧列 (标记为“类”) 。</p> <p>您会注意到中心列“属性”显示 PersonLocationAssociationType.Attributes 以及关联。</p> <p>在关联 Location 和Person旁边放置复选标记。</p> <p>LocationType 和 PersonType 类型会自动添加到Schema 编辑器的“类”列表中。</p>

5	<p>类PersonLocationAssociationType 派生自超类型'nc:AssociationType'，但在这种情况下，超类型不会自动添加到类列表中。</p> <p>如果您想在生成的子集中包含超类型“nc:AssociationType”的任何属性，则必须手动将“nc:AssociationType”添加到Schema 编辑器的类列表中，然后选择所需的属性。</p> <p>如果您不想特别包含“nc:AssociationType”的属性，则无需将其添加到 Classes 列表中。</p> <p>当模式文件最终从子集包生成时，Enterprise Architect将在需要时为“nc:AssociationType”生成元素和类型定义，即使它没有显式建模。</p>
6	<p>单击更新按钮，然后单击生成按钮。</p> <p>在打开的窗口中，选择“NIEM模型子集”，然后单击生成按钮。</p> <p>系统将提示您选择将在其中创建子模型的包。通常，您会选择作为 Exchange 架构包的父级的包。在 starter模型中，交换包名为“IEPD-Exchange”，其父包名为“NIEM 5模型”，尽管您可能在前面的步骤 2 中重命名了这些包。</p> <p>选择包'NIEM 5模型'，然后点击确定按钮。</p>
7	<p>在子模型中找到名为“niem-core”的<<InformationModel>>包。在这个包中创建一个NIEM PIM 图，然后将这个包中的三个类拖放到图上。您会注意到属性“Person”和“Location”被建模为关联 类型 PersonType 和LocationType 之间关联的关联端。</p>

示例3：使用替换组和属性持有者

这个“演练”示例演示了如何使用Enterprise Architect的Schema 编辑器属性正确地将替换组和持有者添加到您的NIEM子集包中。

节	描述
1	<p>打开一个包含NIEM 5.0参考模型和NIEM IEPD类型的Enterprise Architect项目。</p> <p>如果您没有这样的项目，那么请打开一个新项目并使用模型生成器加载参考模型和 IEPD 类型。</p>
2	<p>使用模型生成器，将NIEM 5 IEPD Starter模型的新副本添加到您的项目中。</p> <p>您应该将object实例“NIEM-IEPD”重命名为更有意义的名称。生成 IEPD 时，此object实例的名称用于命名创建 IEPD 的根文件夹。</p> <p>如果您愿意，也可以将包“NIEM 5 Starter模型”重命名为更合适的名称。</p>
3	<p>启动模型包含一个Schema 编辑器编辑器配置文件，名为“工件Schema 编辑器配置文件 - NIEM 5 子集”。找到这个工件，然后在浏览器窗口中双击它。这将打开Schema 编辑器并加载配置文件“Schema 编辑器配置文件 - NIEM 5 子集”。</p> <p>Schema 编辑器的下部包含三列。从左到右，它们被标记为“类”、“属性”和“架构”。</p>
4	<p>使用浏览器窗口，在NIEM 5.0参考模型中找到包“niem-core”。</p> <p>在那个包中，找到类'AircraftType'。</p> <p>将“AircraftType”拖放到Schema 编辑器的左侧列（标记为“类”）。</p> <p>您会注意到类 ConveyanceType 和 ItemType 被自动添加到类列表中。</p> <p>ItemType 和 ConveyanceType 是派生自 AircraftType 的超类型。</p>
5	<p>在Schema 编辑器的左侧栏中选择类'AircraftType'。</p> <p>在中心列中，在属性 AircraftWingColorAbstract 旁边放置一个复选标记（注意该属性没有指定类型）。</p>

	类AircraftWingColorAbstractPropertyHolder 会自动添加到类列表中。
6	<p>选择左侧栏中的类'AircraftWingColorAbstractPropertyHolder'。请注意，这个类还有一个名为“AircraftWingColorAbstract”的属性，它没有指定类型。该属性是为您预先选择的 - 它应该保持选中状态。</p> <p>只需在 AircraftWingColorText 旁边打勾。</p>
7	<p>在这种情况下，属性 AircraftWingColorAbstract 是替换组的头部，并提供客户端类AircraftType 和 供应商类AircraftWingColorAbstractPropertyHolder 之间的连接。</p> <p>AircraftWingColorText 是将添加到 AircraftType 的实际属性（ TextType 类型）。</p>
8	<p>某些 PropertyHolder 类型将具有多个属性 - 替换组头，以及许多其他属性。必须始终在客户端和 供应商类中都选择作为替代组头的属性。 Enterprise Architect在供应商类（ PropertyHolder ）中为您预先选择了这个属性。然后，您只需从供应商处选择要替换的属性来代替替换组的负责人。</p>

示例NIEM架构

本页概述了从头到尾定义新的符合NIEM模式。

NIEM建模所需的框架包已在前面的主题中描述过。模型构建器还提供了一个包，可作为定义 IEPD 的便捷起点。将其导入到模型中后，您将发现包含 IEPD 类型实例的图表，其中的运行状态设置为显示您最有可能需要设置的核心属性。

本节介绍从模式中获取示例 IEPD 并创建 “Hello World” 样 消息的过程，其中根据面部图像发出个性化消息请求。响应将是照片中人物的身份和针对他们的个性化消息。

导入NIEM框架包

使用Enterprise Architect中的NIEM进行建模从NIEM技术架构委员会和物件管理组 NIEM-UML 规范定义的标准类型开始，如此处所述。这些可从我们的可重用资产服务器和模型生成器中获得。

要将这些导入到您的模型中：

- 打开模型生成器 (Ctrl+Shift+M)
- 在模型生成器对话框中，单击 <视角名称> 按钮，然后选择视角 信息交换 > NIEM ”
- 展开项目 “NIEM 3、4 和 5”
- 选择您的模型所需的包
- 点击创建模型按钮将选定的模式导入到您的模型中

注记：

- 所有NIEM 5 模型都需要NIEM IEPD 类型包以及NIEM参考模型包之一
- 所有NIEM 3 和 4 模型都需要NIEM MPD 类型包以及NIEM参考模型包之一
- 所有NIEM 2. 1模型都需要NIEM 2. 1参考模型包但不需要 MPD 类型包，因为NIEM 2. 1 MPD 元素可从NIEM 2. 1 MPD图表工具箱获取

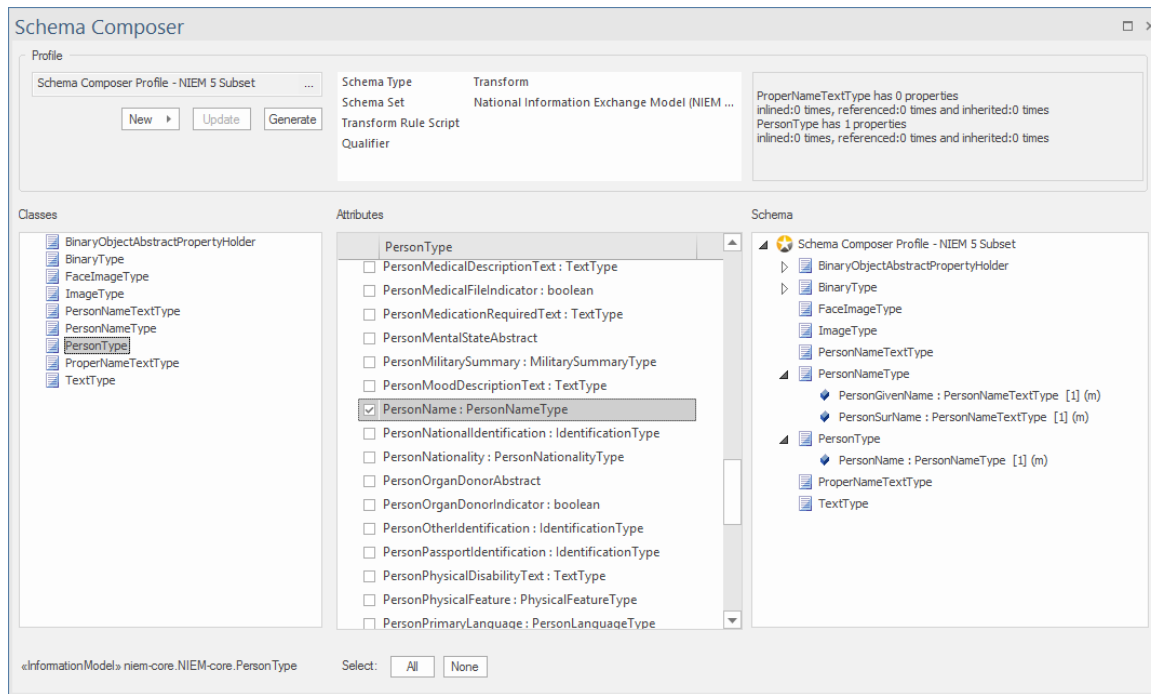
部件	细节
NIEM框架	NIEM的强大功能主要来自于其丰富的类型库，您可以使用它来构建自己的模式。Enterprise Enterprise Architect为NIEM 5 以及所有版本的NIEM 3 和NIEM 4 提供了完成NIEM框架。这些框架都可以从模型构建器获得。 本教程使用NIEM 5.0 框架，因此选择该模式进行导入。
来自 NIEM-UML 的 IEPD 类型	A定义的NIEM模式是围绕 IEPD 构建的，IEPD 为模式的消费者定义了如何使用所包含的各种 XSD 文件以及定义了哪些消息类型。 在UML中建模时，使用UML配置文件中定义的多个类的实例创建 IEPD。Enterprise Enterprise Architect在模型生成器中提供的包中提供了这些类。 所有NIEM 5 模型都需要这些 IEPD 类型，因此选择 “NIEM 5 IEPD 类型” 模进行导入。

子集NIEM命名空间

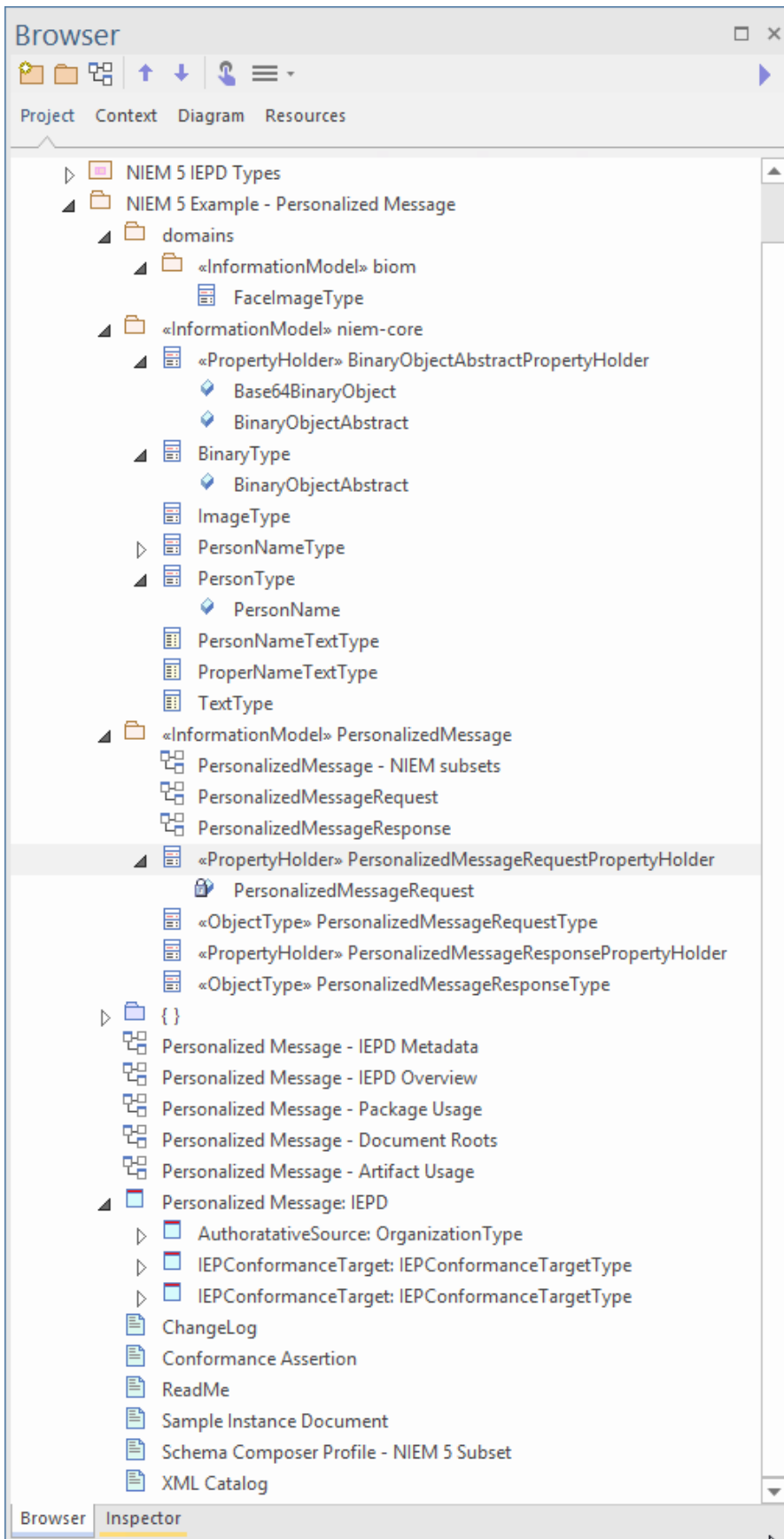
工件模式包括一个Schema 编辑器模型，用于指定一个子集。双击它打开Schema 编辑器并开始子集化过程。

我们希望我们的请求消息发送用于面部识别的面部图像。为此，我们需要从 Biometrics包中子集适当的类型。开始通过在NIEM 5.0参考模型中的 Domains\Biom包中找到类型 FaceImageType。将此类型拖入Schema 编辑器。该类型所继承的超类型会自动添加到Schema 编辑器中。

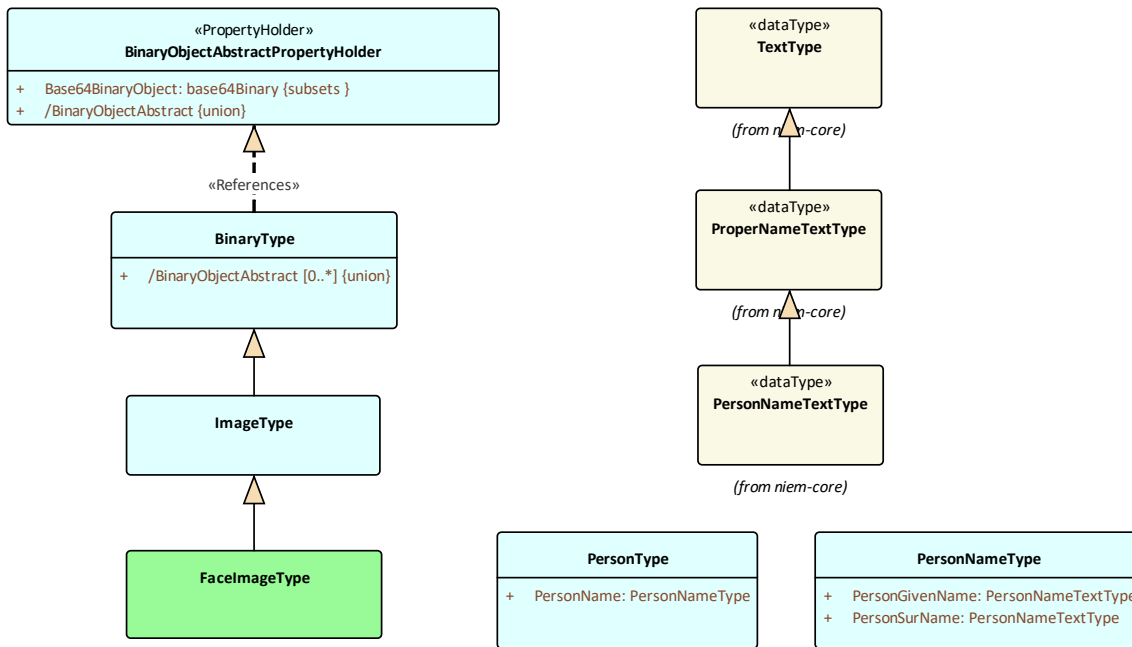
我们的响应消息需要来自 'niem-core'包的 PersonType 。将此类型也拖到Schema 编辑器上。
此图显示了在NIEM 5.0参考模型中跨多个命名空间选择类型和属性的子集：



一旦选择了所需的类型，您就可以生成子集。出现提示时，选择将生成子命名空间包的父包。生成后，子集包中的类应如下所示：

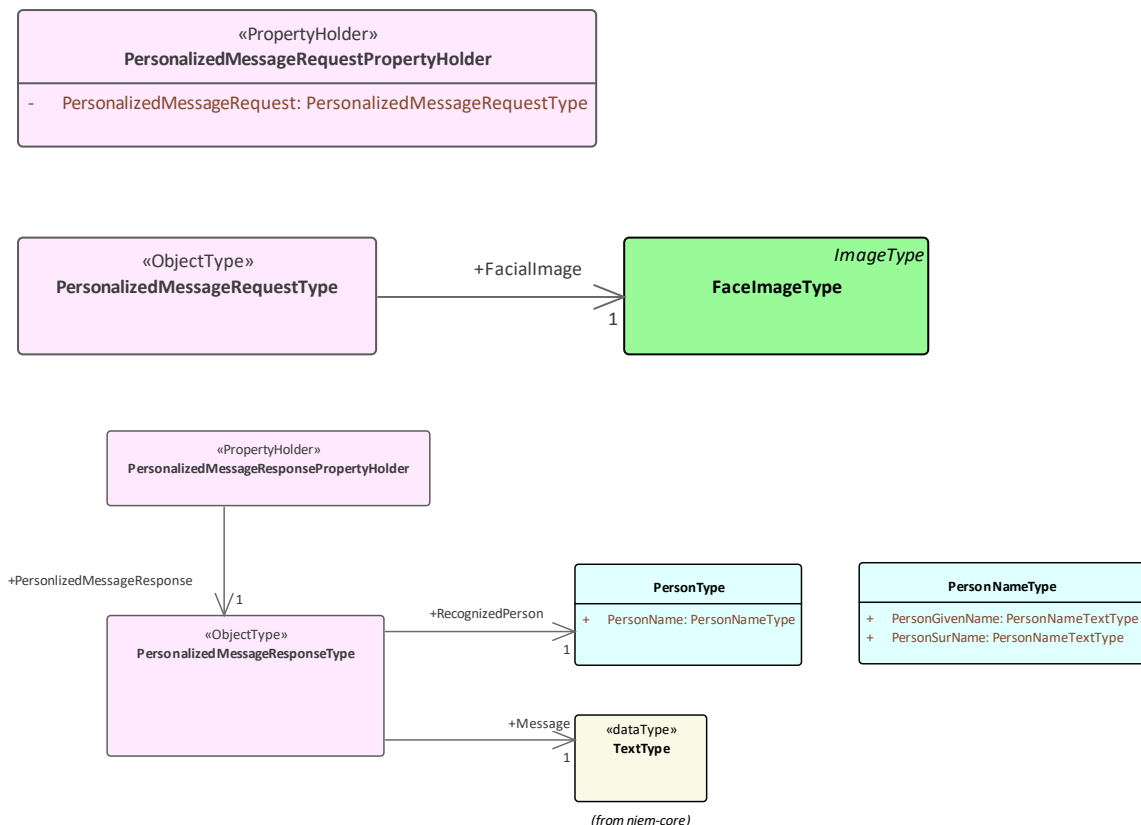


我们现在可以创建一个现在PIM 图NIEM我们所有的子类类放在该图上，以生成类似于以下内容的内容：



创建扩展类型

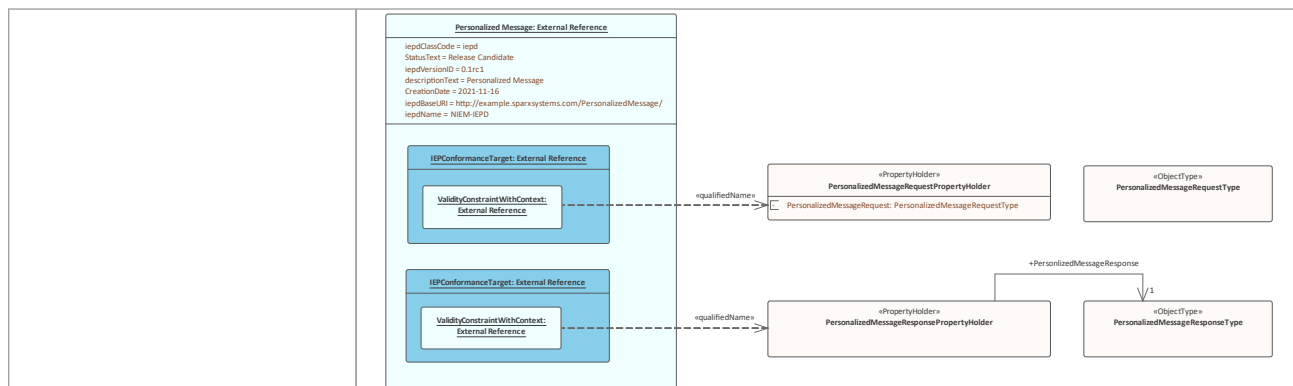
我们将定义两条消息，一条请求和一条响应。对于这些消息中的每一个，我们都需要定义文档根元素。这些将被建模为NIEM模式的扩展。现在我们已经定义了我们的子集包，我们可以定义这些文档根。因为我们只创建了两种简单的文档类型，所以只需要为每条消息创建一个 PropertyHolder 和ObjectType。 ObjectTypes 链接到我们NIEM框架中选择的类型，用于描述每条消息的内容，如下所示：



自定义 IEPD

IEPD类的实例携带标识 IEPD 的信息，它与各种其他模型工件之间的链接决定了在生成模式文件和目录文件时生成什么（以及生成到哪里）。这里描述了要点。

部件	描述
IEPD 元数据	<p>模式中的顶级object是 IEPD 类的一个实例。IEPD 的名称是物件本身的名 称。所有其他属性都处于object的运行状态。 该图显示了 IEPD 在提供真实信息后的样子。</p> <div data-bbox="528 622 1291 1312" style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <p style="text-align: center;">Personalized Message: External Reference</p> <pre>iepdClassCode = iepd StatusText = Release Candidate iepdVersionID = 0.1rc1 descriptionText = Personalized Message CreationDate = 2021-11-16 iepdBaseURI = http://example.sparxsystems.com/PersonalizedMessage/ iepdName = NIEM-IEPD</pre> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p style="text-align: center;">AuthoritativeSource: External Reference</p> <pre>name = Sparx Systems</pre> <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> <p style="text-align: center;">OrganizationPrimaryContactInformation: External Reference</p> <pre>ContactEmailID = support@SparxSystems.com ContactWebsiteURI = https://www.SparxSystems.com</pre> </div> </div> </div> <p>NIEM-UML 建议iepdBaseURI 的最后部分与IEPD 的名称相匹配，并指定 iepdVersionID 将附加到iepdBaseURI 以生成生成的iepdURI。此示例遵循该约 定。 该模式将iepdClassCode 的值默认为'iepd'。这意味着 IEPD 旨在表示信息交换 包文档 (IEPD)。这是最常见的 IEPD 类型，也是我们想要创建的，所以一直 保留默认值。</p>
定义的文档类型	IEPD 应定义一种或多种文档类型。每一个都是名为 “IEPConformanceTarget”的 IEPConformanceTargetType 实例。提供的模型模式 已经包括其中之一，但我们需要第二个，如下所示：

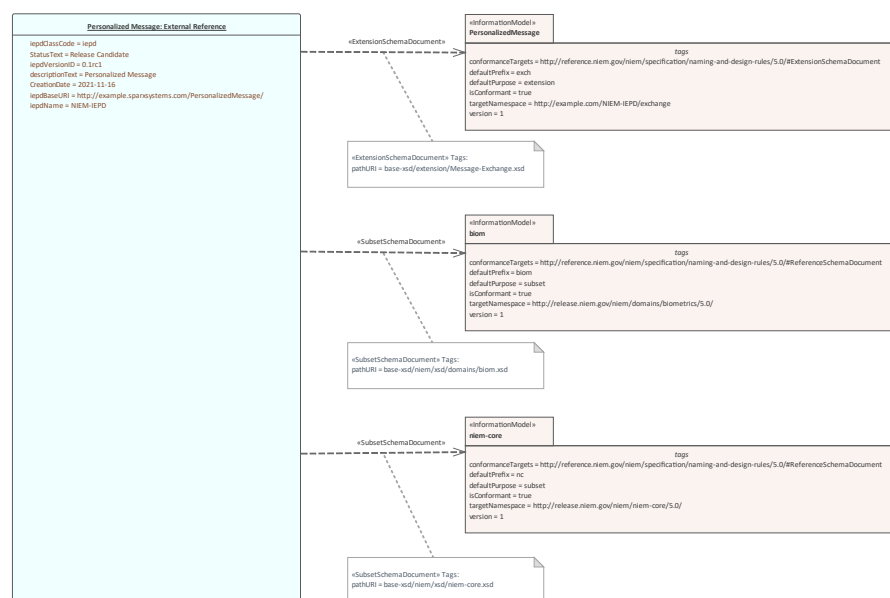


注册QualifiedNamesType 的实例，其qualifiedName关系到一个PropertyHolder。这指定正在描述的文档的顶层将是来自所包含属性之一的元素。主题创建NIEM数据模型中的创建扩展包部分描述了它是如何定义的。

用途包

将 IEPD 实例连接到信息模型的关系指定要使用该 IEPD 生成哪些模式文件。在此示例中，我们使用来自两个不同NIEM命名空间的类型。子设置过程为每个创建了一个 InformationModel包，其中命名空间标记值与原始匹配，目的设置为子集。我们还创建了一个扩展包，我们在其中定义了我们自己的类型以及如何使用NIEM类型。

此图显示了它的外观：

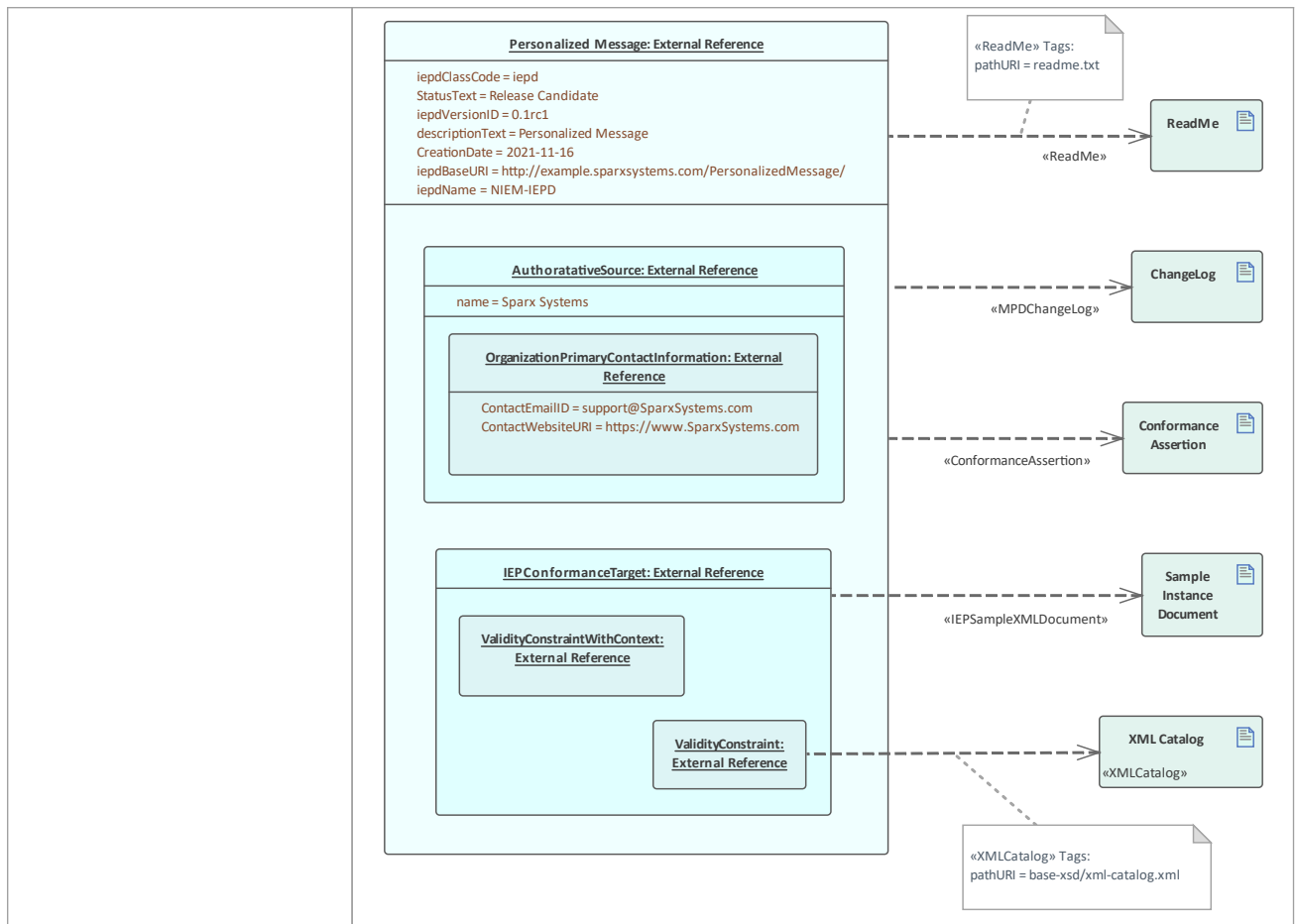


使用的关系还指定包使用方式以及该包定义的模式的路径。

附加文件

NI 预计所有NIEM包都至少包含更改log和自述文件，但也支持其他几种类型的工件包。在Enterprise Architect的工件关系。与包使用一样，关系指定文件的位置。

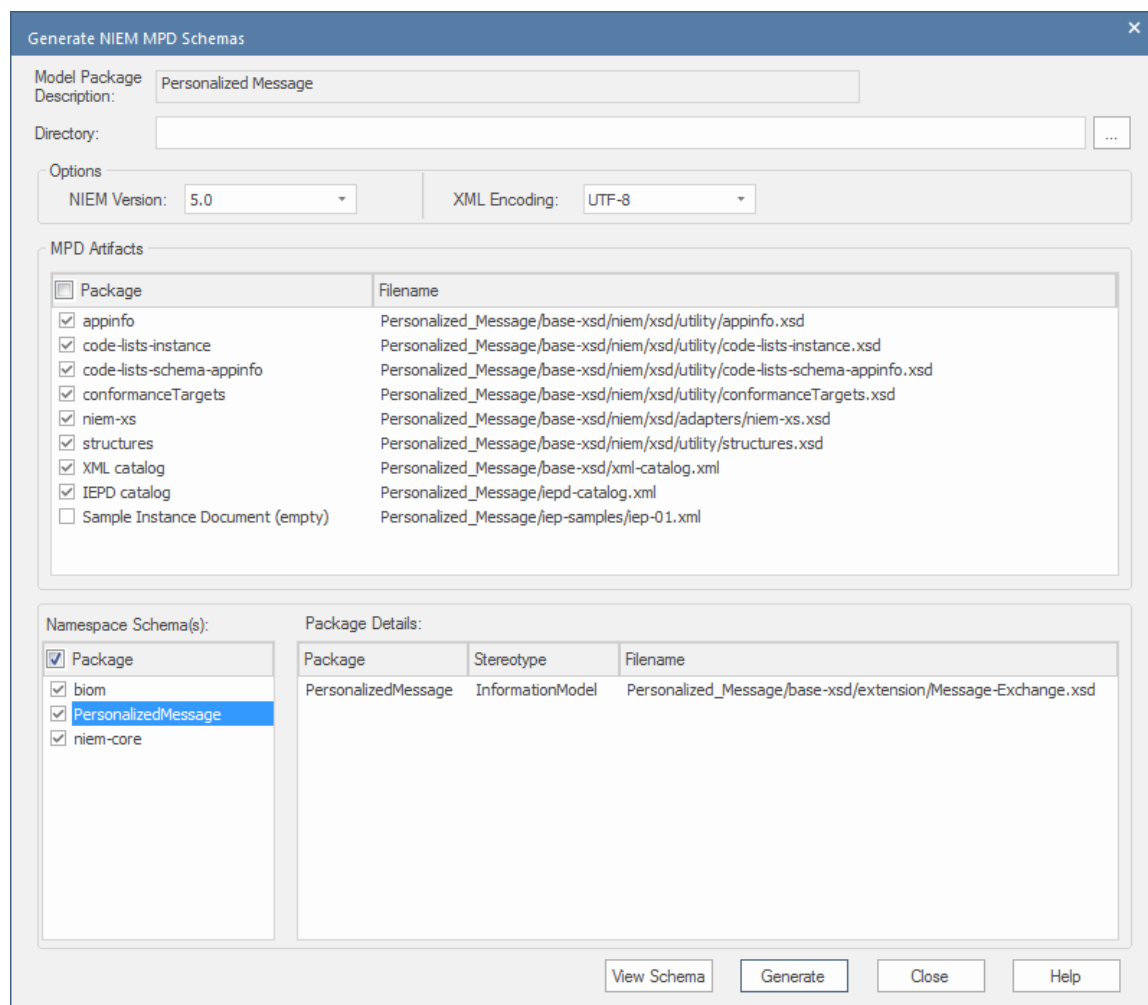
在此图像中，描述了每种文档类型的自述文件、变更日志和示例文档。这会将有关这些文件的信息添加到目标目录文件中。Enterprise Architect不会创建这些文件，它们的内容超出了本教程的范围。



生成IEPD

要生成您的 IEPD :

- 在图表或浏览器窗口中选择 IEPD 实例规范。
- 从 特定”功能区中，选择 技术> NIEM >生成NIEM架构”选项



该对话框显示标准NIEM工件和可作为模式生成的链接命名空间列表。设置目标目录并单击“生成”按钮以创建建模的 IEPD。

导入NIEM XML Schema

除了在Enterprise Architect中生成NIEM模式外，您还可以将外部 NIEM 特定XML Schema文件作为UML模型导入（反向工程）到Enterprise Architect项目中。

访问

功能区	特定>技术> NIEM >导入NIEM架构 特定>技术> NIEM 2.1 >导入NIEM 2.1架构
上下文菜单	在浏览器窗口中： 右键包 特定 NIEM 导入NIEM架构 右键包 特定 NIEM 2.1 导入NIEM 2.1架构

导入一个NIEM特定的XML Schema

选项	行动
包	在浏览器窗口中显示当前选定包的名称，作为导入NIEM模式的包。 您可以通过单击  按钮并检查“导航员”对话框来验证您使用的是适当的包；如有必要，请选择不同的包。
目录	单击  按钮并浏览包含源NIEM架构文件的目录。单击要导入的每个文件，然后单击浏览器打开按钮。
选定的文件	列出选择导入的XML Schema文件。
导入引用的XML Schema (s)	如果您要导入“选定文件”字段中列出的任何文件所引用的任何其他XML Schema，请选中此复选框。
跳过架构如果在模型中命名空间	如果模型中已存在 XML 模式，则如果要跳过导入XML Schema，请选中此复选框。 Enterprise Architect将使用架构命名空间和名称来确定它是否存在于模型中。
图表XML Schema (s)	选中此复选框以在每个导入的命名空间包下创建类图（NIEM PIM图）。
布局创建图表	（仅在选择XML Schema（S）'选项的“创建图表”时才启用。） 选中此复选框以自动布置创建的类图。
导入	单击此按钮开始导入过程。 导入进度报告在系统输出窗口的“NIEM Importer”选项卡中。当导入完成时，还会显示A消息框；单击确定按钮以清除消息。
关	单击此按钮可关闭“架构导入器”对话框。

帮助	单击此按钮可显示此帮助页面。
----	----------------

注记

- Enterprise Architect使用XML Schema的 XSD导入和 XSD Include 元素中的`schemaLocation`属性来确定文件之间的依赖关系；此属性必须设置为有效的文件路径（而不是 URL），才能正确导入依赖的XML Schema
- 'Create图表for XML Schema (s)' 选项为每个导入的模式文件生成图表，但仅显示用户特别选择的模式文件的图表；它不显示引用的架构文件的图表
- 如果您导入大型架构文件，建议您取消选择 为XML Schema (s) 创建图表”选项，因为这会大大增加导入所花费的时间

