



**ENTERPRISE ARCHITECT**

用户指南系列

# Unified Modeling Language (UML)

Author: Sparx Systems

Date: 13/11/2024

Version: 17.0

创建于  **ENTERPRISE  
ARCHITECT**

# 目录

Unified Modeling Language (UML)	8
UML图表	10
UML结构模型	11
类图表	12
复合结构图表	15
属性	17
部件图表	19
部署图表	21
物件图	25
包图	27
配置文件图表	29
UML行为模型	31
活动图表	32
用例图表	35
示例用例图表	37
状态机	38
伪状态	44
区域	45
创建连接点参考	46
状态机库表	48
状态机库表Options	49
状态机库表操作	51
更改状态机库表位置	52
更改状态机库表大小	53
插入触发器	54
插入/更改转移	55
插入新状态	57
重新状态或状态触发器细胞	58
添加图例	59
在状态机图表中查找 Cell	60
状态机库表约定	61
导出状态库表转csv文件	62
示例状态-触发器库表	63
示例状态-下个状态库表	64
状态机库表仿真	65
图表	67
创建一个图表	69
设置时间范围	70
编辑图表	71
添加和编辑状态生命线	72
将状态添加到状态生命线	74
在状态生命线中编辑状态	75
删除状态生命线中的状态	76
在状态生命线编辑转场	77
添加和移动过渡	78
添加和编辑值生命线	80
在价值中添加国家生命线	81

编辑价值生命线过渡	82
配置时间线 - 状态	84
数值范围生成器	86
配置时间线 - 过渡	87
时间间隔	89
创建时间间隔	90
压缩时间间隔	92
选择时间间隔	94
时间间隔操作	95
消息 ( 图表 )	98
创建时间信息	99
序列图表	101
表示元素的生命周期	104
布局序列图表	105
序列元素	106
消息 ( 序列图表 )	107
自我留言	110
调用	112
信息示例	113
更改时序细节	115
一般订购	117
异步信号信息	118
共同区域表示法	119
序列和图表版本控制	120
序列元素Activations	121
生命线激活水平	123
序列信息标签可见性	124
更改顶部边距	125
内联序列元素	126
通讯图表	127
通讯图表颜色	129
消息 ( 通讯图表 )	130
创建通讯信息	131
重新排序消息	132
交互图表	133
UML元素	136
行为图表元素	137
行动	138
行动类型	140
可变行动	144
当地的前置/后置条件	146
类操作在图表中	147
行动销	149
分配行动销	150
活动	151
活动符号	153
活动参数Nodes	154
活动分区	156
参与者	157
Central Buffer节点	158
选择	159

组合片段	161
创建组合片段	164
交互	165
约束	168
数据存储	169
决策	170
图框	172
门	174
端点	175
入口	176
事件	177
例外	178
膨胀节点	179
拓展区域	180
出口	183
终点	184
流终点	186
分叉/汇合	187
分叉	189
汇合	190
历史	191
最初的	193
交互	195
交互发生	197
可中断活动区域	199
连接点	200
生命线	202
合并	203
信息端点	204
信息标签	205
注记	206
物件节点	207
分区	208
接收	210
区域	211
发送	212
状态	213
复合状态	214
状态/延续	216
延续	217
状态不变	219
状态生命线	220
状态机	222
结构活动	223
结构化节点	225
顺序节点	226
Loop节点	227
条件节点	230
同步	232
系统边界	233
系统边界属性	235

终止	240
触发器	241
用例	243
用例扩展点	245
价值生命线	246
结构图表元素	248
工件	249
工件文件	253
使用检查清单和审核清单工件	254
使用阅读工件	258
文档工件	259
定制库表工件	260
类	265
活动课程	267
参数化类 (模板)	268
协作	270
协作应用	271
部件	273
数据类型	274
部署规范	275
设备	276
枚举	277
执行环境	278
曝露接口	279
信息项	280
接口	281
节点	282
物件	283
运行时状态	284
物件状态	286
包	287
包装部件	288
部件	289
添加属性值	290
端口	291
添加一个端口到一个元素	292
继承并重新定义端口	293
端口as Owners of Parts	294
属性窗口 -属性 · 重新定义/子集	295
原始	296
信号	297
接待	298
属性信号接收窗口	300
UML连接器	302
抽象	303
聚合	304
更改聚合连接器Form	305
组装	306
关联	307
限定词	308
定义限定词对话框	309

关联类	311
将新类连接到现有关联	313
通讯路径	314
组合	315
N-Ary关联	316
连接器	317
控件	318
代表	319
依赖	320
应用构造型	321
部署	322
延长	323
概括	326
包括	327
信息流	328
使用信息流	330
传达有关流程的信息	333
实现信息流	334
中断流	336
显现	337
信息	338
嵌套	339
注释链接	340
物件流	341
活动图表中的物件流	342
发生	343
导入包	344
包合并	345
实现	346
递归	347
角色捆绑	348
代表	349
表示	350
替代	351
模板捆绑	352
参数替换	353
跟踪	355
转移	356
内部转移	358
用途	360
使用	361
UML构造型	362
应用构造型	363
构造型选择器	364
构造型能见度	366
标准构造型	368
带有选择的构造型图片	370
自定义构造型	371
扩展UML	373
使用UML Profiles	374
将配置文件对象添加到一个图表	375

Profiles中的标记值	376
同步标记值和约束	377
扩展构造型	379
边界	380
创建一个边界	381
控件	382
创建一个元素控件	383
实体	384
创建一个实体	385
超链接	386
图像	388
进程	389
风险	390
任务	391
测试元素	392
测试用例	393
设计模式	394
发布模式	395
工件一个模式	397
导入模型模式	399
使用模式	400
添加模式对话框	402

# Unified Modeling Language (UML)



Enterprise Architect提供了大量工具，建模者可以使用这些工具来创建符合各种正式和非正式建模语言的模型。其中一种语言是Unified Modeling Language ( UML )， Enterprise Architect全面支持该语言中指定的所有元素、关系和图表。UML由物件管理组 (OMG) 管理，其中Sparx Systems是管理和改进语言过程的积极成员和贡献者。

## 功能

功能	描述
Unified Modeling Language ( UML )	<p>UML标准定义了用于指定业务和软件系统的符号和规则；该符号为面向object系统建模提供了一组丰富的图形元素，以及如何连接和使用这些元素的状态。</p> <p>UML不是创建软件系统的工具；相反，它是一种用于交流、建模、指定和定义系统的视觉语言。</p> <p>UML不是对软件系统建模的规定过程；它不提供方法或过程，只提供语言。因此，您可以以多种方式使用UML来指定和开发您的软件工程项目。</p> <p>这种语言被设计为灵活、可扩展和全面，但又足够通用，可以作为所有系统建模要求的基础。有了它的规范，就有了范围广泛的元素，这些元素的特点是它们所服务的图表种类和它们提供的属性。所有这些都可以通过使用刻板印象、标记值和配置文件来进一步指定。</p> <p>Enterprise Architect支持许多不同类型的UML元素（以及一些自定义扩展）；连同元素之间的连接器，这些构成了模型的基础。</p>
广泛的应用	<p>虽然最初被设想为一种用于软件开发的语言，但UML可用于模型广泛的现实世界领域和流程（在商业、科学、工业、教育和其他地方）、组织层次结构、部署图等等。</p> <p>Enterprise Architect还提供了额外的自定义图表和元素，以解决进一步的建模兴趣。</p>
为新领域扩展UML	<p>使用UML Profiles、模式、语法、数据类型、约束、MDG 技术和其他扩展，可以定制UML和Enterprise Architect以解决原始UML规范中未明确定义的特定建模域。</p> <p>Enterprise Architect使扩展UML变得简单直接，而且最重要的是，扩展机制仍然是UML规范的一部分。</p>
推荐阅读	<p>除了 OMG 提供的UML规范之外，还有两本书对UML进行了出色的介绍：</p> <ul style="list-style-type: none"> <li>• Schaum 的大纲：Bennett、Skelton 和 Lunn 的UML（第二版） 由麦格劳希尔出版。 国际标准书号：0-07-710741-1 ISBN-13：978-0-07-710741-3</li> <li>• Bernd Oestereich 用UML开发软件 艾迪生卫斯理出版。 ISBN-10：0201398265 ISBN-13：978-0201398267</li> </ul>



# UML图表

UML图是模型一部分的图形表示，通常显示通过关系连接的许多元素。图是存储库中最具表现力和吸引力的视图之一；该类型通常是特定受众构建的，以传达一个想法或创建模型的叙述性描述。图也可用于生成有用的系统图，具有名称和工件模式等数据库模式、编程代码等等。

UML规范定义了 14 种类型的图并列出了可以包含在每个图上的元素和关系。Enterprise Architect默认工具箱中为每种图表类型方便地提供了这些元素。虽然这些工具箱可作为新手建模者的指南，但经验丰富的建模者可以通过在同一图表中包含各种元素类型来创建高度表达的图表。

图表在主工作区中创建和查看，并存储在包或存储库中的其他元素中。

## 图表分组

团体	细节
结构图表	结构图描绘了构成系统或功能的结构函数，反映了结构或运行时架构的静态关系。
行为图表	行为图显示模型的动态视图，描述系统或业务流程的行为特征。
扩展图表	Enterprise Architect提供了一组额外的图表类型，这些图表类型扩展了特定领域模型的核心UML图。
自定义图表	Enterprise Architect还支持特定于MDG 技术的图表类型，包括集成技术。

# UML结构模型

UML结构图描述了与时间无关的系统元素，它们传达了系统的概念以及它们如何相互关联。这些图中的元素类似于自然语言中的名词，连接它们的关系是结构关系或语义关系。例如，一个车辆预订系统的结构图可能包含诸如 Car、Reservation、驱动因素、License 和 Credit Card 等元素，以及链接这些元素的连接器。经验丰富的建模者还将在这些图表上显示与行为元素的关系。

UML定义了七种类型的UML结构图。

## 结构图表类型

图表类型	细节
类	类图捕获系统的逻辑结构、构成模型的类和对象，描述存在的内容以及它具有的属性和行为。
复合结构	复合结构图反映了类、接口和组件（及其属性）的内部协作以描述功能。
部件	部件图说明了组成系统的软件、嵌入式控制器等部件，以及它们的组织和依赖关系。
部署	部署图显示了系统的部署方式和位置；也就是它的执行架构。
物件	物件图描述了类的object实例及其在某个时间点的关系。
包	包图描述了模型元素到包中的组织以及它们之间的依赖关系。
配置文件	配置文件图是在«profile»包中创建的，用于扩展UML元素、连接器和组件。

# 类图表

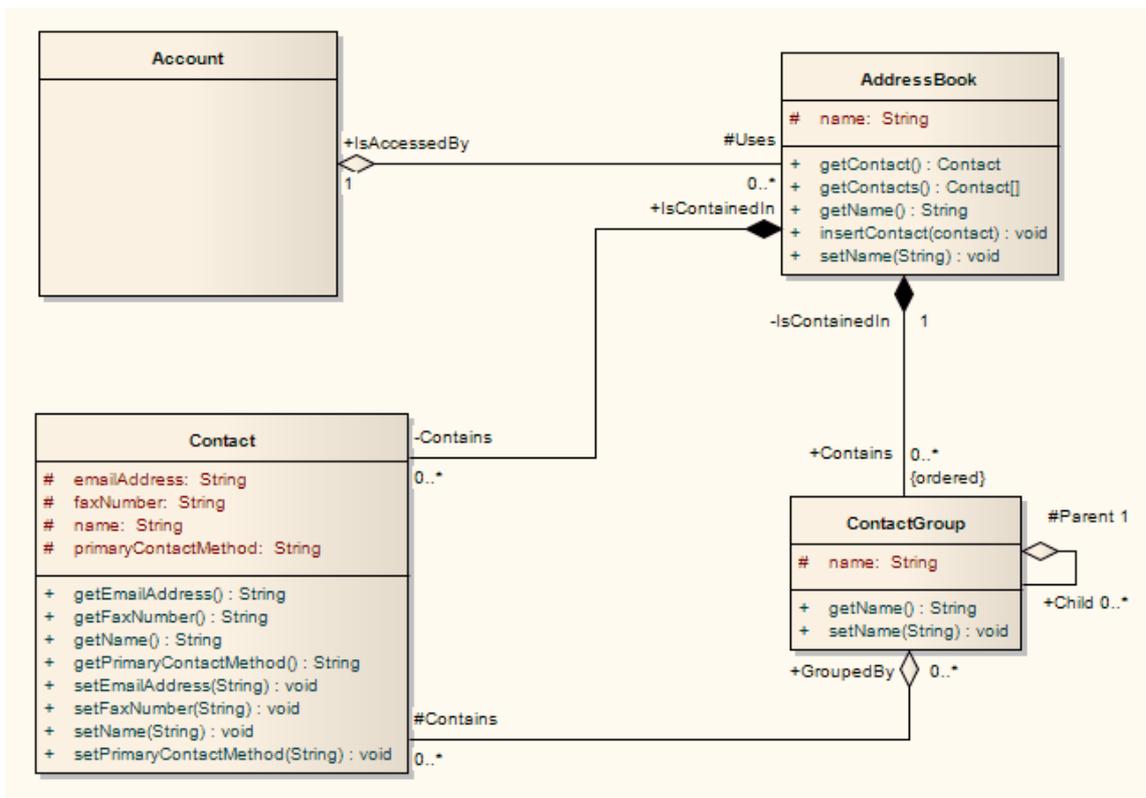
类图捕捉系统的逻辑结构——类——以及构成模型的事物。它是一个静态模型，描述的是存在什么以及它具有什么属性和行为，而不是描述某事是如何完成的。在类图中，您可以使用泛化、聚合和关联来说明类和接口之间的关系，它们分别在反映继承、组合或使用以及连接方面很有价值。

您可以从工具箱的“类”页面图表类图元素和连接器。

## 示例图表

在此示例类图中，聚合关系有两种形式：

- 淡色表示类账户使用通讯录，但不一定包含通讯录
- 深色复合聚合形式表示源类的目标类（在菱形端）的所有权或包含



## 类图表元素工具箱Icons

图标	描述
 Class	A类是一种object类型的表示，它反映了系统内此类对象的结构和行为。
 Interface	An接口是实现者同意满足的行为规范（或合同）。
 Data Type	数据类型是A特定的分类器，类似于类，只是数据类型不能拥有子数据类型，并且数据类型的实例仅由它们的值标识。

 Enumeration	枚举是一种数据类型，其实例可以是许多用户定义的枚举文字中的任何一种。
 Primitive	原始元素标识A预定义的数据类型，没有任何相关的子结构（也就是说，它在上下文的UML中没有部分）。
 Signal	信号是对象之间通信A发送请求实例的规范，通常在类或包图中。
 Association	n-模型关联元素用于模拟三个或更多元素之间的复杂关系，通常在类或物件图中。

## 类图表连接器工具箱Icons

图标	描述
 Associate	关联意味着两个模型元素具有关系，通常实现为一个或两个类中的实例变量。
 Generalize	A概括用来表示继承。
 Compose	组合A描述由较小组件组成的元素，通常在类或包图中。
 Aggregate	聚合连接器是一种显示元素包含其他元素或由其他元素组成的关联类型。
 Association Class	关联类是使关联具有属性和操作（特征）的UML构造。
 Realize	源object实现或实现其目标object A
 Template Binding	在绑定类和参数化类之间创建一个模板捆绑连接器。

## 类图表复合

图标	描述
 Part	部件是类或接口的运行时实例。
 Port	端口定义分类器与其环境之间的交互。
 Expose Interface	暴露接口元素是一种图形方法，用于在类、部件或复合结构图中描述部件、类或部件的所需或提供的接口。
 Assembly	组装连接器将组件所需的接口 (Component1) 与另一个组件 (Component2) 提供的接口连接起来，通常在部件图中。

 Connector	连接器说明了部件之间的通信链接以实现结构的目的，通常在类或复合结构图中。
 Delegate	代表连接器定义组件A外部组件的内部端口和接口，在类图或部件图上

## 类UML图表标准配置文件

UML标准配置文件是按照UML 2.5 规范（第 22 章，标准配置文件）作为建模工具提供的原型类、操作和关系的集合。

其中一些建模元素可以通过类或包图工具箱中的“UML标准配置文件”工具箱页面直接获得；其他的可以用作基本UML建模object的原型。

# 复合结构图表

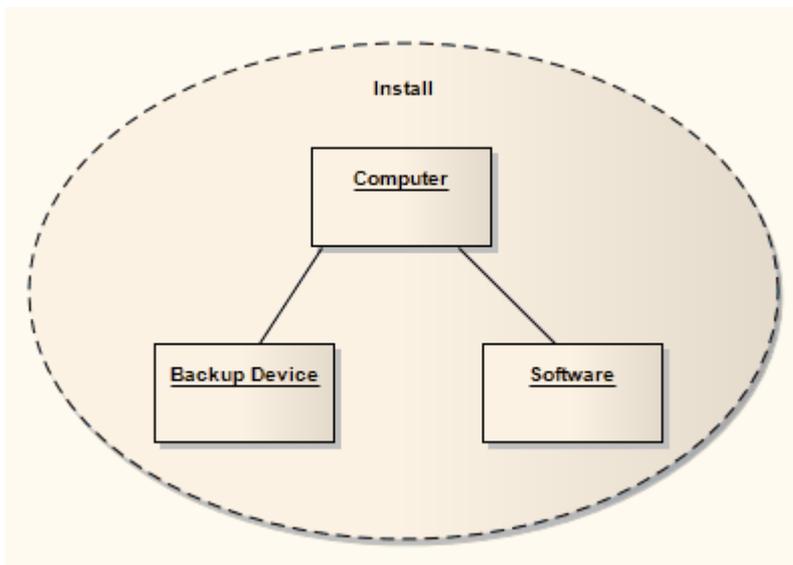
复合结构图反映了类、接口或组件（及其属性）A内部协作来描述功能。复合结构图类似于类图，但类图模型是类结构的静态视图，包括它们的属性和行为,复合结构图模型是结构的具体用法。您可以使用它们来表达运行时架构、使用模式和参与元素的关系，这些可能无法通过静态图反映出来。

在复合结构图中，类作为部件或履行特定角色的运行时实例来访问。如果由类填充的角色需要多个实例，则这些部件可以具有多重性。由一个部件的类定义的端口应该在复合结构中显示，以便所有连接部件都提供由端口指定的所需接口。对复合结构进行建模具有广泛的灵活性和随之而来的复杂性。要优化您的建模，请考虑构建协作以表示响应您的设计问题的可重用模式。

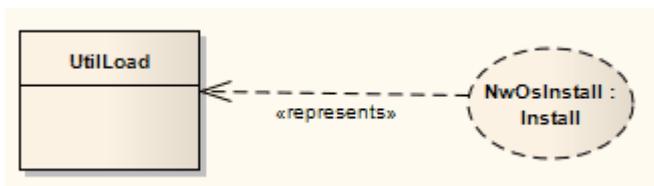
您可以从工具箱的“复合”页面图表复合结构图元素和连接器。

## 示例图表

此图显示了协作在复合结构图中显示执行安装的关系的图。协作通常用于模型常见的模式。



下一个图表通过协作应用程序使用此协作，并将其应用于协作类«表示»协作关系。这表明分类器协作在其实现中使用了协作模式。



## 复合结构图表元素工具箱Icons

图标	描述
 Class	A类是一种object类型的表示，它反映了系统内此类对象的结构和行为。
 Interface	An接口是实现者同意满足的行为规范（或合同）。

 Part	部件是类或接口的运行时实例。
 Port	端口定义分类器与其环境之间的交互。
 Collaboration	协作定义了A组协作角色及其连接器。
 Collaboration Use	使用协作应用将协作定义的模式应用于特定情况，在复合结构图中。
 Expose Interface	暴露接口元素是一种图形方法，用于在部件或复合结构图中描述部件、类或部件的所需或提供的接口。

### 复合结构图表连接器工具箱Icons

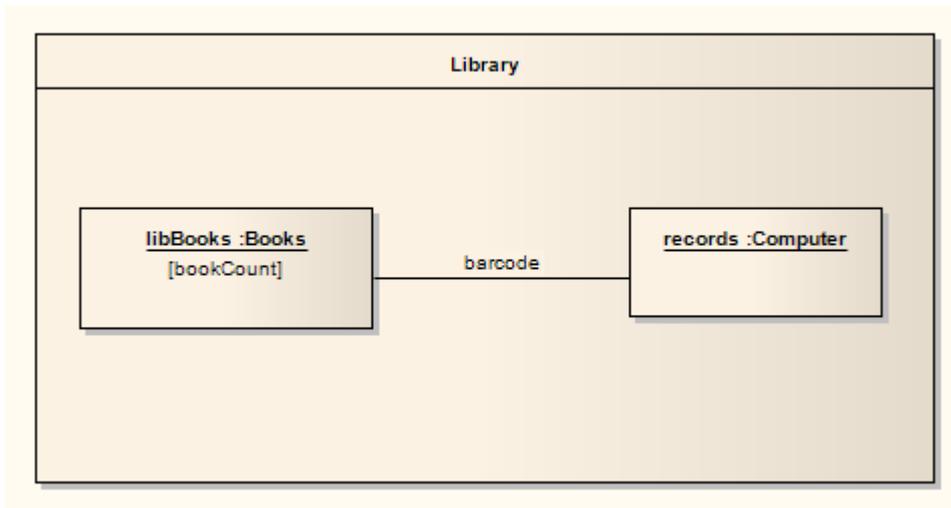
图标	描述
 Connector	连接器说明了部件之间的通信链接以实现结构的目的，通常在复合结构图中。
 Assembly	组装连接器将组件所需的接口 (Component1) 与另一个组件 (Component2) 提供的接口连接起来，通常在部件图中。
 Delegate	A委托连接器在一个部件上定义了一个组件的外部端口图和部件图。
 Role Binding	角色捆绑是协作应用内部角色与实现特定情况所需的各个部件之间的映射，通常在复合结构图中。
 Represents	代表连接器表示在协作分类器中使用了 a，通常在复合结构图中使用。
 Occurrence	一个协作关系表示一个类图代表一个分类器，在一个复合结构图中。

# 属性

属性是分类器中A嵌套结构，通常是复合结构图中的类或接口。包含的结构反映了包含分类器中反映的实例和关系。属性可以具有多重性，并且可以显示为：

- 零件（首选）或
- 关联角色

## 部分



在此图中，有两个部分，“libBooks”和“记录”，它们分别是对应于“书籍”和“电脑”类的实例。两个 Part 之间的关系由连接符表示，反映 Parts 之间的通信是通过条码进行的。这个包含的结构和它的部件是由图书馆类拥有的属性。

将零件从图表工具箱拖到类上后，右键单击部件并选择“高级”设置属性类型以连接到分类器。如果拖动到类上时部件消失，调整类的 Z 顺序以将其移动到部件后面（右键单击类并选择“Z-Order”选项）。

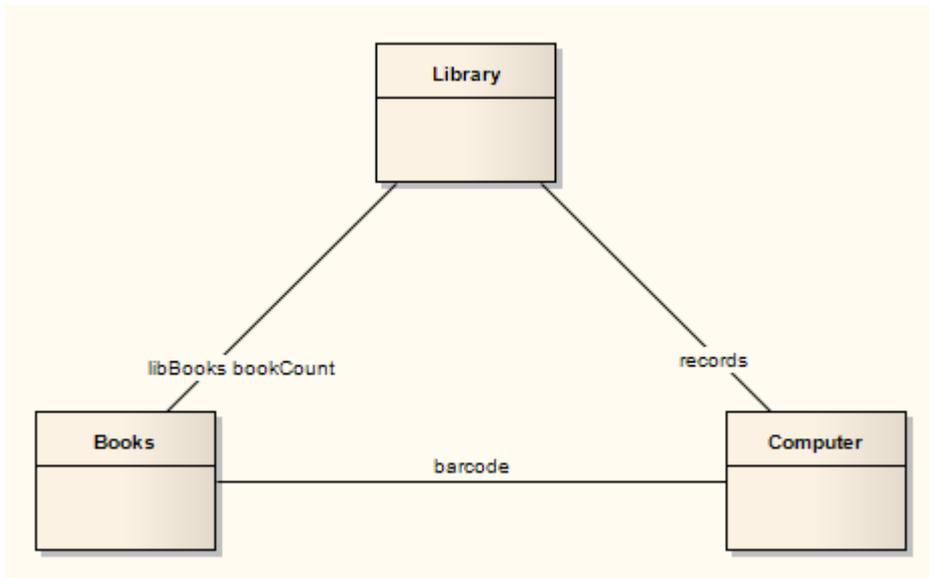
要向包含的分类器指示不属于组合的属性，请使用带有虚线轮廓的框符号，表示关联；去做这个：

1. 右键单击部件并选择“属性”选项。
2. 选择“属性”对话框的“高级”页面。
3. 将“IsReference”选项设置为 True。

## 关联角色

属性也可以使用普通的复合结构来反映（不包含在类中），通过与类的连接来指示适当的连接器、部件和关系。

此处显示了替代表示；然而，这种表示未能通过在分类器中包含属性来表达立即反映的所有权。



# 部件图表

部件图说明A构成系统的软件、嵌入式控制器等部分，以及它们的组织和依赖关系。

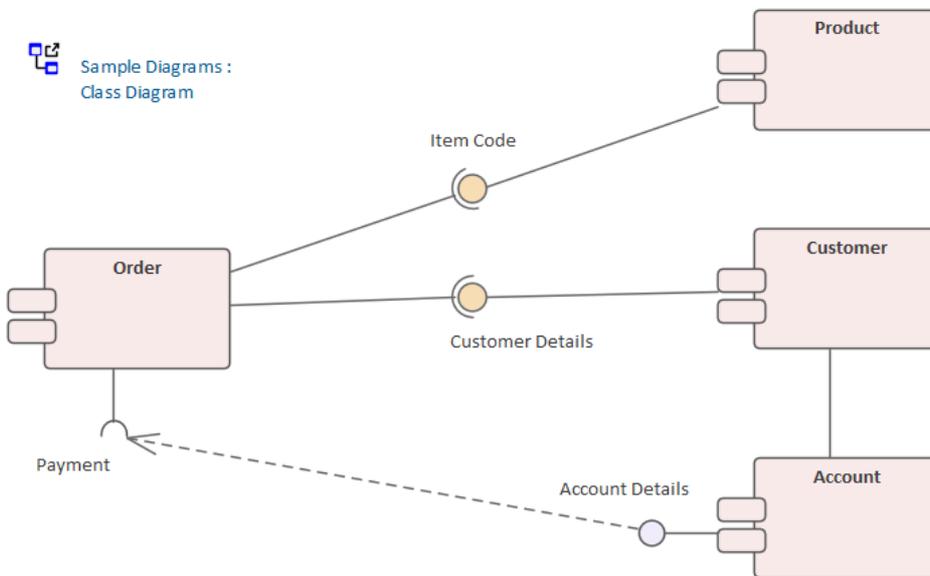
部件图比A图具有更高的抽象类；通常一个组件在运行时由一个或多个类（或对象）实现。它们是构建块，构建起来以便最终组件可以包含系统的大部分。

您可以从“工具箱”工具的“部件”页面图表部件图元素和连接器。

## 示例图表

该图演示了许多组件及其相互关系。

组装连接器将 Product 和顾客提供的接口连接到 Order 指定的所需接口。依赖关系A客户的关联帐户详细信息映射到所需的接口 Payment，也由 Order 指定。



## 部件图形元素工具箱图表

图标	描述
Packaging Component	Packaging 部件是A在图表中看起来与部件非常相似但在浏览器窗口中表现为包的元素。
Component	部件是系统A模块化部分，其行为由其提供和所需的接口定义。
Class	A类是一种object类型的表示，它反映了系统内此类对象的结构和行为。
Interface	An接口是实现者同意满足的行为规范（或合同）。
Object	物件是类在运行时的特定实例。
Port	端口定义分类器与其环境之间的交互。

 Expose Interface	<p>暴露接口元素是一种图形方法，用于在部件或复合结构图中描述部件、类或部件的所需或提供的接口。</p>
--	--

### 部件图形连接器工具箱图表

图标	描述
 Assembly	<p>组装连接器将组件所需的接口 (Component1) 与另一个组件 (Component2) 提供的接口连接起来，通常在部件图中。</p>
 Delegate	<p>A委托连接器在一个部件上定义了一个组件的外部端口图和部件图。</p>
 Associate	<p>关联意味着两个模型元素具有关系，通常实现为一个或两个类中的实例变量。</p>
 Realize	<p>源object实现或实现其目标object A实现连接器用于用例、部件或需求图中，以表示模型中的可追溯性和完整性。</p>
 Generalize	<p>A概括用来表示继承。</p>

## 部署图表

部署图显示了系统A部署方式和位置；也就是它的执行架构。

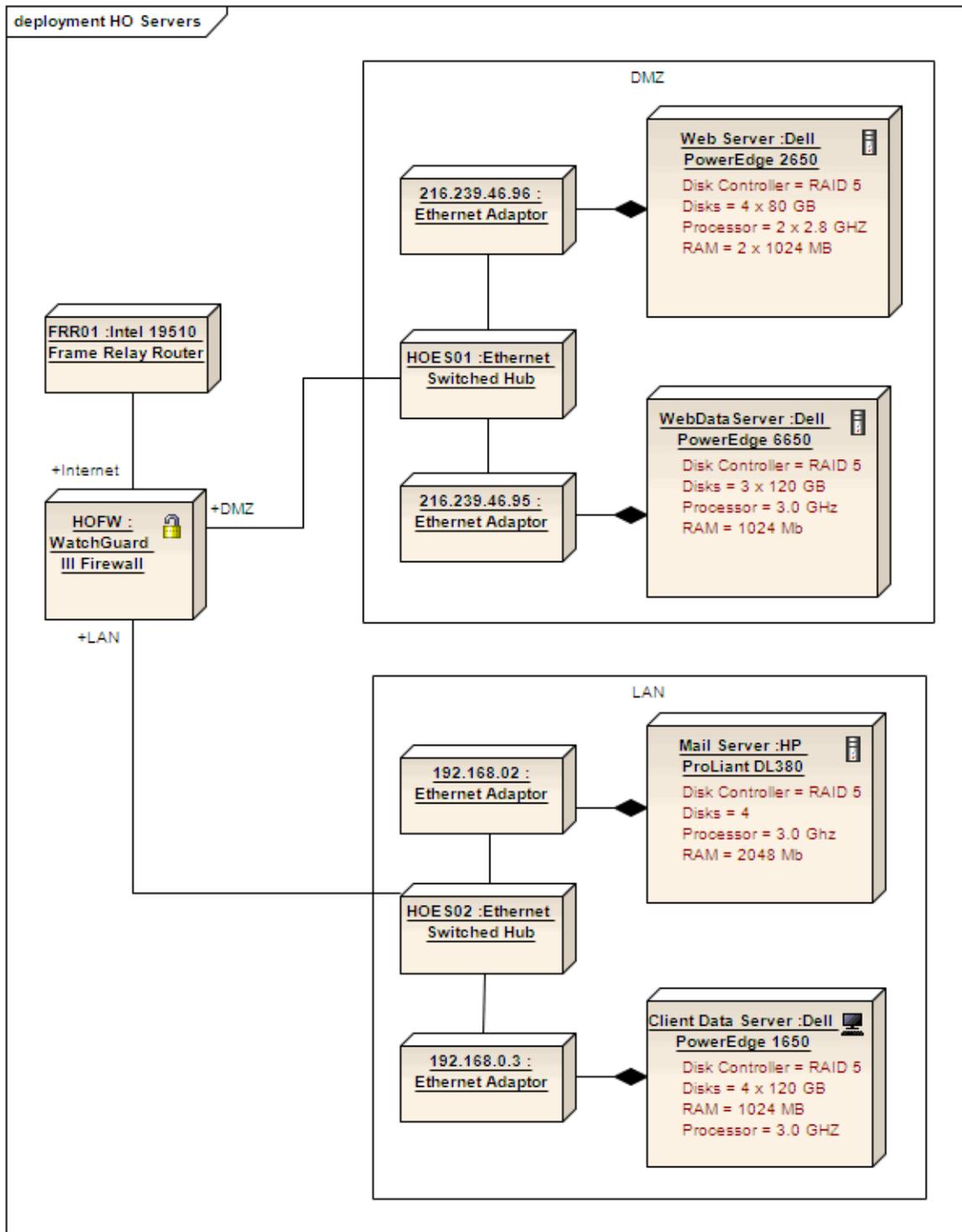
软件执行环境（系统）和软件执行环境（工件）和反映为节点可以通过嵌入的硬件设备或内部嵌套节点来描述。部署关系表示部署工件、组件和清单的部署关系。工件给节点的节点分配到模型系统部署时，以部署规范为指导。部署图也可以表明A节点有一个状态，或者显示一个节点的实例以及该状态的实际运行状态，代表一个特定的条件或场景。

您可以从图表工具箱的“部署”页面图表部署图元素和连接器。

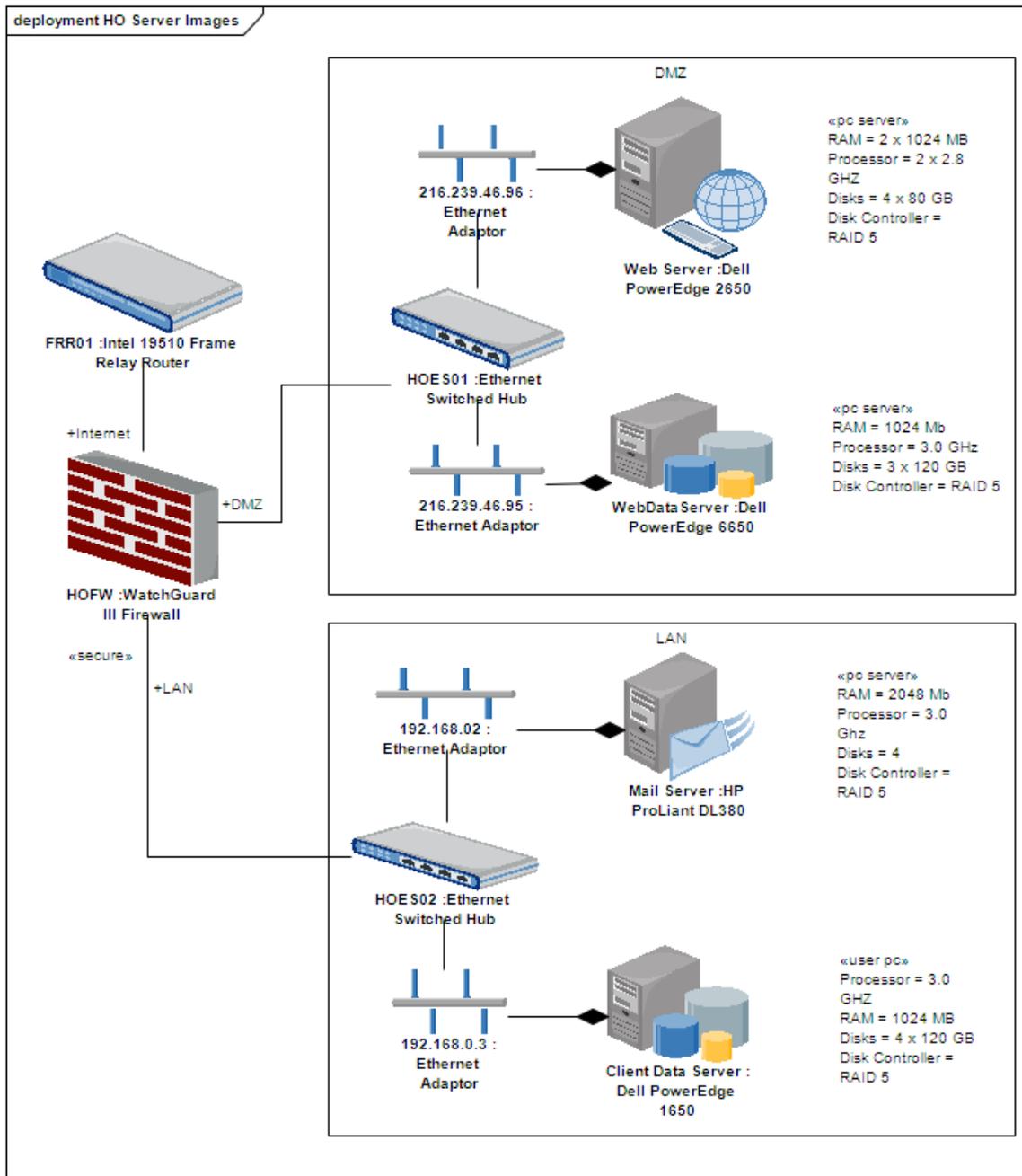
### 示例图表

这是一个简单的部署图，表示总部的服务器布置。这些元素是节点的实例并显示特定的运行时状态。

服务器由通过简单或聚合关联关系链接的节点表示。



部署图非常适合应用替代图像来描述元素所代表的对象。此类图像可以替换图中的元素，如下所示：



### Deployment 图表元素工具箱 Icons

图标	描述
 Node	节点是部署系统A物理设备，例如工作组服务器或工作站。
 Device	设备是具有处理能力的工件资源，可以部署以执行，如部署图中A电子资源。
 Execution Environment	一个提供特定类型的组件执行环境的执行环境是部署在其上的一个节点，该执行环境是以工件的形式执行的。

 <b>Component</b>	部件是系统A模块化部分，其行为由其提供和所需的接口定义。
 <b>Interface</b>	An接口是实现者同意满足的行为规范（或合同）。
 <b>Artifact</b>	工件是系统使用或产生的任何物理信息。
 <b>Deployment Specification</b>	部署规范指定A指导部署工件的参数（硬件规范），这是大多数软件技术所必需的。

## 部署图表连接器工具箱Icons

图标	描述
 <b>Associate</b>	关联意味着两个模型元素具有关系，通常实现为一个或两个类中的实例变量。
 <b>Communication Path</b>	通讯路径定义了两个 DeploymentTargets 能够交换信号和消息A路径。
 <b>Association Class</b>	关联类是使关联具有属性和操作（特征）的UML构造。
 <b>Generalize</b>	A概括用来表示继承。
 <b>Realize</b>	源object实现或实现其目标object A
 <b>Deployment</b>	A Deployment 类型的工件关系是一个部署关系到一个节点或一个可执行目标的部署，通常在一个 Deployment 图中。
 <b>Manifest</b>	A工件关系表明目标模型源能体现元素，通常在部件和部署图中。

# 物件图

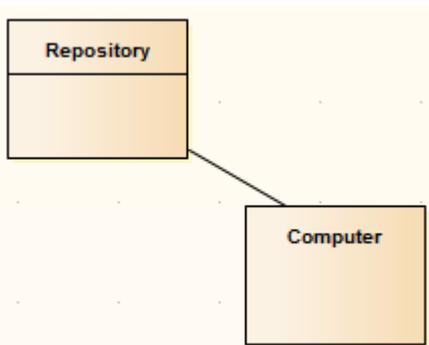
物件图与类图密切相关，区别在于它描述了类的object实例及其在某个时间点的关系。物件图不揭示与其对应的类图不同的体系结构，而是反映多样性和实例化的角色类可以服务。通过创建应用关系和类的不同案例，它们有助于理解复杂的类图

这可能看起来类似于复合结构图，它也模拟运行时行为；不同之处在于，物件图是静态类图的例证，而复合结构图反映了与其静态对应物不同的运行时架构。一个物件图也可以是一种通讯图（它也对对象之间的连接进行建模，但另外对每条路径上的事件进行排序）。

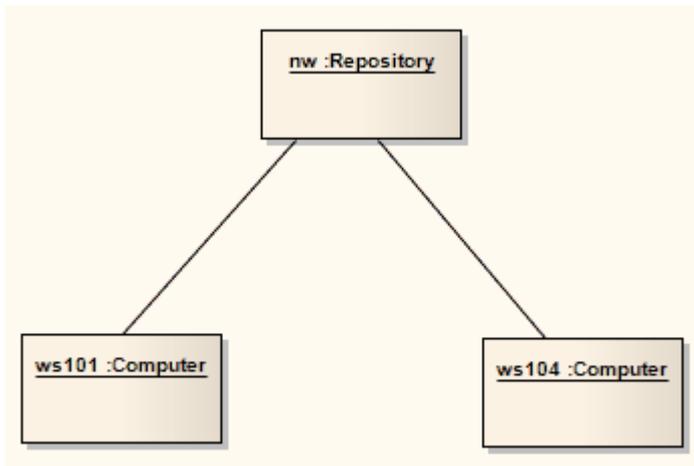
您可以从图形工具箱的“物件”页面图表物件图元素和连接工具箱。

## 示例图表

此示例显示了一个简单的类图，其中连接了两个类元素。



这些类被实例化为物件图中的对象。该模型中有两个电脑实例，证明了物件图在考虑类在实践中可能存在的关系和相互作用方面的有用性。



## 物件图元素工具箱图标

图标	描述
 Actor	参与者是系统的用户；用户可以指人类用户、机器，甚至是模型中的另一个系统或子系统。
 Object	物件是类在运行时的特定实例。

 Collaboration	协作定义了A组协作角色及其连接器。
 Collaboration Use	使用协作应用将协作定义的模式应用于特定情况，在复合结构图中。
 Boundary	A是一个边界的物件，它模拟一些系统边界，通常是一个用户界面屏幕。
 Control	A控件是一个原型化的物件，它对控制实体或管理器进行建模。
 Entity	实体是一种原型化的物件，它对捕获系统中的信息或知识的存储或持久性机制进行建模。
 Association	n-模型关联元素用于模拟三个或更多元素之间的复杂关系，通常在类或物件图中。

### 物件图连接器工具箱Icons

图标	描述
 Information Flow	信息流表示任何图表中两个元素之间的信息项（信息项元素或分类器）的流动。
 Associate	关联意味着两个模型元素具有关系，通常实现为一个或两个类中的实例变量。
 Dependency	依赖关系用于建模模型元素在用例、活动和结构图中，甚至模型本身之间的广泛依赖关系。

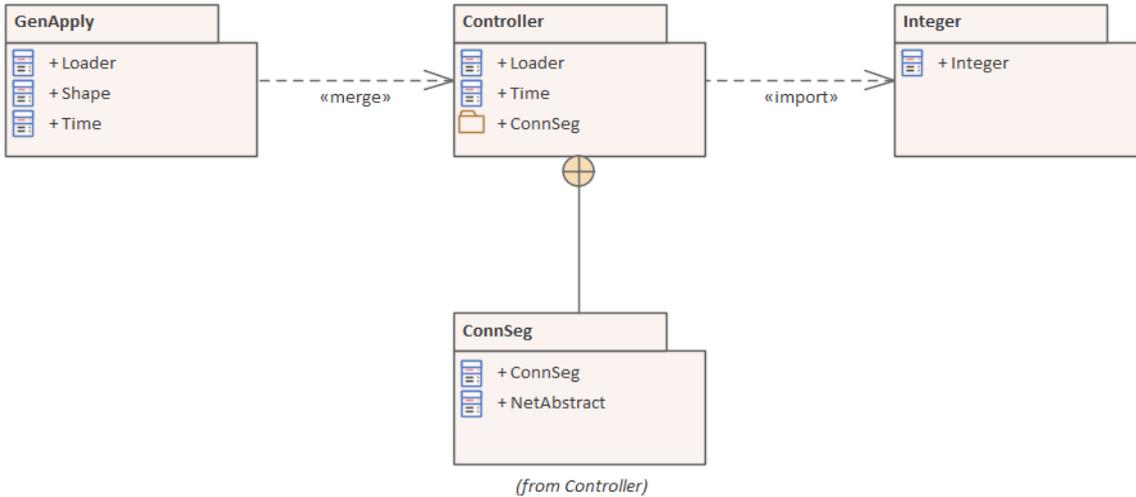
# 包图

包图描述了模型元素到包中的组织以及它们之间的依赖关系，包括包导入和包扩展。它们还提供了相应命名空间的可视化。

您可以从工具箱的“包”页面图表包图元素和连接器。

## 示例图表

这个例子说明了一个基本的包图。



## 包图元素工具箱Icons

图标	描述
 Package	包用于组织您的项目内容，但是当添加到图表时，它们可以用来描述您的模型的结构和关系。
 Profile	在您的技术开发模型的包图中生成具有构造型 <code>«profile»</code> 的配置文件包。配置 A 配置文件包用于在模型中定义新的结构类型。
 Model	生成具有原型 <code>«model»</code> 的模型包，以表示模型结构中的父节点。

## 包图关系工具箱Icons

连接器	描述
<code>c_nesting</code>	嵌套连接器的其他元素是一种替代图形，而不是表示包含或嵌套元素内的元素。 Conn 和控制器之间的嵌套连接或反映了包内容所揭示的内容可以通过单击图

	<p>表背景以显示图表的“属性”对话框，选择“元素”选项卡并在“显示分区”面板中选中“包内容”复选框来列出包内容。</p>
c_pkgmerge	<p>在包图中，包Merge表示两个包之间的关系，目标包的内容与源包的内容合并。</p> <p>在示例图中，«merge»连接器表示已将控制器包的元素导入到GenApply中，包括控制器的嵌套和导入内容。</p> <p>如果元素中已经存在元素，例如时间，则这些元素的定义将由包中包含的元素定义和控制器。由合并添加或更新的所有元素都由返回到该包的泛化关系记录。</p>
c_pkgimport	<p>导入包关系是A源包到内容已经被导入的包。</p> <p>«import»表示目标整数器包中的元素是本例中的整数类，已经被导入到了驱动包控制器。</p> <p>控制器获得了整数类的命名空间的访问权；整数空间不受影响。</p>
c_profileapplication	<p>配置文件A关系表示源配置文件已经应用到目标包。</p>

## UML标准配置文件工具箱图标

图标	描述
 Framework	生成具有原型«framework»的模型包，以表示框架结构中的父节点。
 Metamodel	生成具有原型«元模型»的模型包，以表示元模型结构中的父节点。
 Model Library	生成具有原型«modelLibrary»的模型包，以表示模型库结构中的父节点。
 System Model	生成具有原型«systemModel»的模型包，以表示系统模型结构中的父节点。

# 配置文件图表

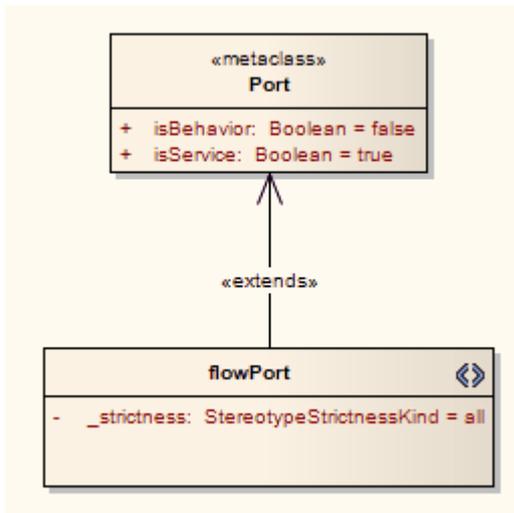
配置文件图是A «profile»包中创建的任何图。

Profiles提供了一种扩展UML的方法。它们基于应用于UML元素、连接器及其组件的附加构造型和标记值。配置文件是此类扩展A集合，它们共同描述某些特定的建模问题并促进该领域中的建模构造。

您可以从工具箱的“配置文件”页面图表配置文件图表元素和连接器。

## 示例图表

配置文件图上A典型单位类似于：



## 配置文件元素工具箱图表

图标	描述
Profile	创建UML配置文件的第一个阶段是创建一个配置文件包，该配置文件包在您的技术开发模型中具有构造型 «profile»。
Stereotype	构造型元素表示每个object的扩展方式。
Metaclass	元类元素表示您在配置文件包中扩展的object类型。
Enumeration	枚举是一种数据类型，其实例可以是许多用户定义的枚举文字中的任何一种。

## 配置文件图表连接器工具箱Icons

图标	描述

 Extension	扩展类型的连接器表示两个元素之间的“范围”关系。
 Generalize	A概括用来表示继承。
 Tagged Value	A标记值连接器定义了一个标注类型（即RefGUID）属于源元素的标记值；标记值名称是该连接器的目标角色的名称，并且标记值仅限于引用具有目标元素的构造型的元素。

# UML行为模型

UML行为图表描述了依赖于时间的系统元素，并传达了系统的动态概念以及它们如何相互关联。这些图中的元素类似于自然语言中的动词，连接它们的关系通常传达时间的流逝。例如，车辆预订系统的行为图可能包含诸如“预订”、“租车”和“提供信用卡详细信息”等元素。经验丰富的建模者将在这些图表上显示与结构元素的关系。

UML定义了七种类型的行为图。

## 图表类型

图表类型	细节
活动图表	活动图模型系统的行为，以及这些行为在系统整体流程中的关联方式。
用例图表	用例图捕获参与者和系统之间的使用案例和关系；它们描述了系统的功能要求、外部操作员在系统边界交互的方式以及系统的响应。
状态机图表	状态机图说明了元素如何在状态之间移动，根据转换触发器和约束守卫对其行为进行分类。
图表	时序图定义了不同对象在时间尺度内的行为，提供对象随时间变化状态和交互的可视化表示。
序列图表	序列图是行为的结构化表示，随着时间的推移一系列顺序步骤。它们用于描述工作流程、信息传递以及元素如何随着时间的一般进行合作以实现结果。
通讯图表	通讯图在运行时显示元素之间的交互，可视化对象间的关系。
交互图表	将交互关系图、交互、序列通讯和其他交互图之间的合作关系图可视化，以说明服务流程的目的。

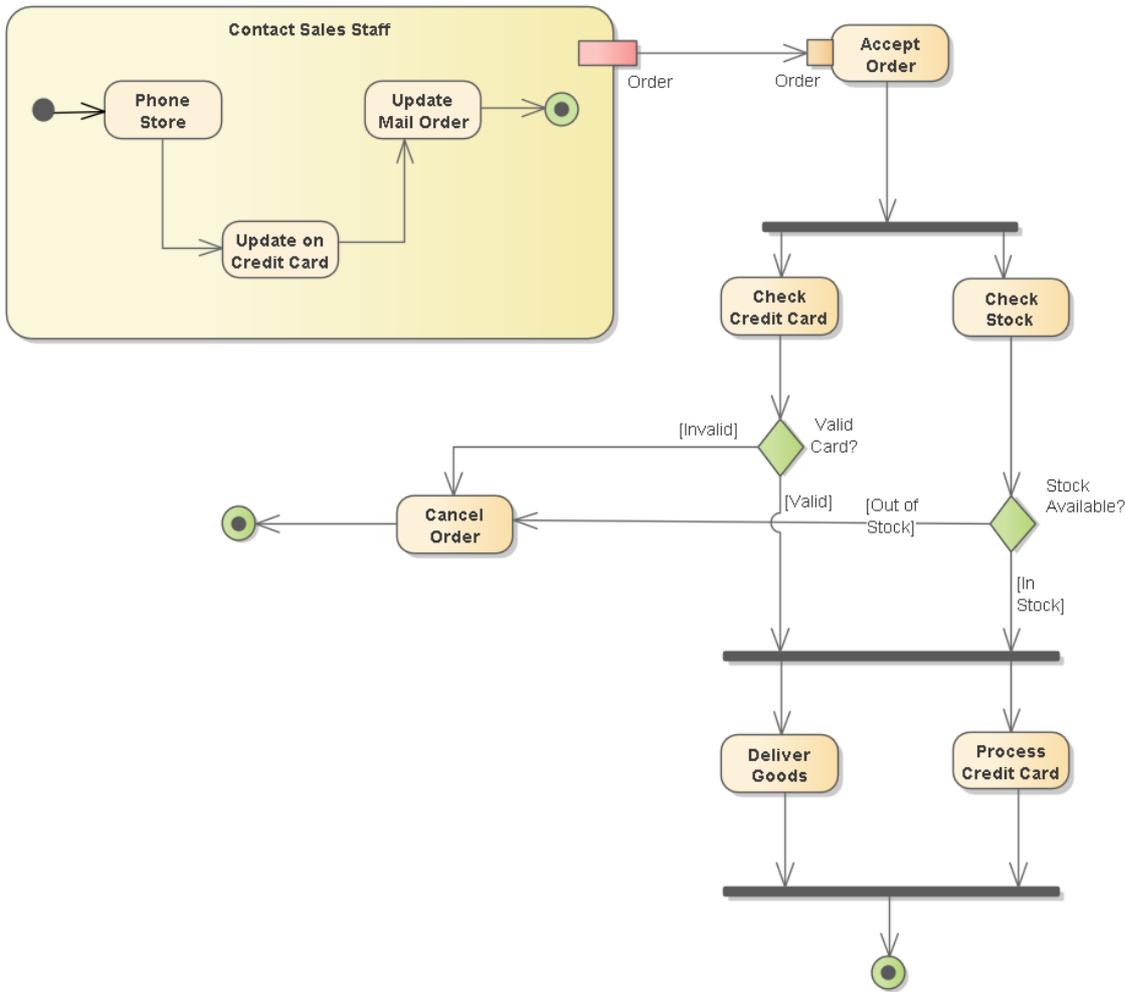
# 活动图表

活动图用于模型系统行为，以及这些行为在系统整体流程（即动态元素交互）中的关联方式。一个进程所遵循的逻辑路径，基于各种条件、并发处理、数据访问、中断等逻辑路径的区别，都用于构造一个进程、系统或过程。

您可以从图表工具箱的“活动”页面图表活动图表元素和连接器。

## 示例图表

该图说明了活动图的一些特征，包括活动、行动、开始节点、结束节点和决策点。



## 活动图表元素工具箱

图标	描述
 Activity	活动元素组织和指定从属行为的参与，例如子活动或行动，以反映流程的控制和数据流。
	行动元素描述了系统内发生的基本过程或转换，是活动图中的基本功能单

	元。
 Partition	分区元素用于A逻辑上组织活动的元素。
 Send	发送元素在活动图中描绘了发送信号的动作。
 Receive	接收元素A活动图中定义了请求的接受或接收。
 Structured Activity	A结构活动是一个活动节点，可以有从属节点作为一个独立的活动组。
 Expansion Region  Interruptible Region	Enterprise Architect支持两种类型的区域元素：扩展区域和可中断活动区域。 一个扩展区域围绕一个对输入数据多次施加的过程，对输入集中的每个元素一次。 一个可中断的活动区域围绕着一组活动元素，所有这些元素都受到某些中断的影响，这样在该区域内经过的所有令牌都会在中断被引发时终止。
 Exception	异常处理程序元素定义了发生异常时要执行的操作组。
 Activity Parameter	一个活动参数节点接受一个活动的输入或提供一个活动的输出。
 Object	物件是类在运行时的特定实例。
 Central Buffer Node	Central Buffer节点是A object节点，用于管理来自多个源和目的地的流，在活动图中表示。
 Datastore	数据存储定义永久存储A数据。
 Expansion Node	扩张节点是表示行动/活动由扩张区域组成的简写符号。
 Initial	Initial元素用于定义调用活动时流程的开始。
 Decision	在一个活动图决策或交互图中，一个决策一个点条件进展：如果一个条件为True，则处理单向；如果没有，那么另一个。
 Merge	A Merge节点a Merge together a flow paths in活动,分析和选择交互图表。
 Synch	同步状态对于指示状态机的并发路径是同步A很有用。它用于拆分和重新加入并行处理周期。
 Fork/Join  Fork/Join	A分叉/汇合元素可用于： <ol style="list-style-type: none"> <li>1) 将单个流拆分为多个并发流</li> <li>2) 汇合多个并发流或</li> <li>3) 将多个传入流加入和分叉到多个传出流</li> </ol>
 Flow Final	活动流终点元素表示系统的退出，与活动终点，表示系统的完成。
 Final	活动终点元素表示活动的完成；到达活动终点后，图中的所有执行都被中止。

## 活动图表连接器工具箱Icons

图标	描述
 Control Flow	控件连接活动图中的两个节点，对活动转换进行建模。
 Object Flow	一个物件流连接两个元素，通过它的特定数据，模拟一个主动过渡。
 Interrupt Flow	中断流定义了异常处理程序和可中断活动区域的连接器的两个UML概念。

## 图表导向

在活动图中，你可以设置流向为水平或者垂直，或者无（默认）。

要设置或清除方向，请右键单击图表背景，然后单击“设置图表流方向”。然后单击：

- 无（默认，未设置特定方向）
- 水平（图表横跨页面，泳池和泳道元素占据图表的整个宽度），或
- 垂直（图表沿页面向下流动，泳池和泳道元素占据图表的整个高度）

## 注记

- 您可以使用“新图表”对话框创建包含对业务流程建模最有用的元素的分析图（简化活动图）
- 您可以对活动模型进行模型模拟，您模拟的模型可以包含多个包中的元素；要在模拟中包含外部元素，您必须创建一个包含“父”包和包含外部元素的“外部”包的包图，然后创建一个从父包到每个外部包的包导入连接器

# 用例图表

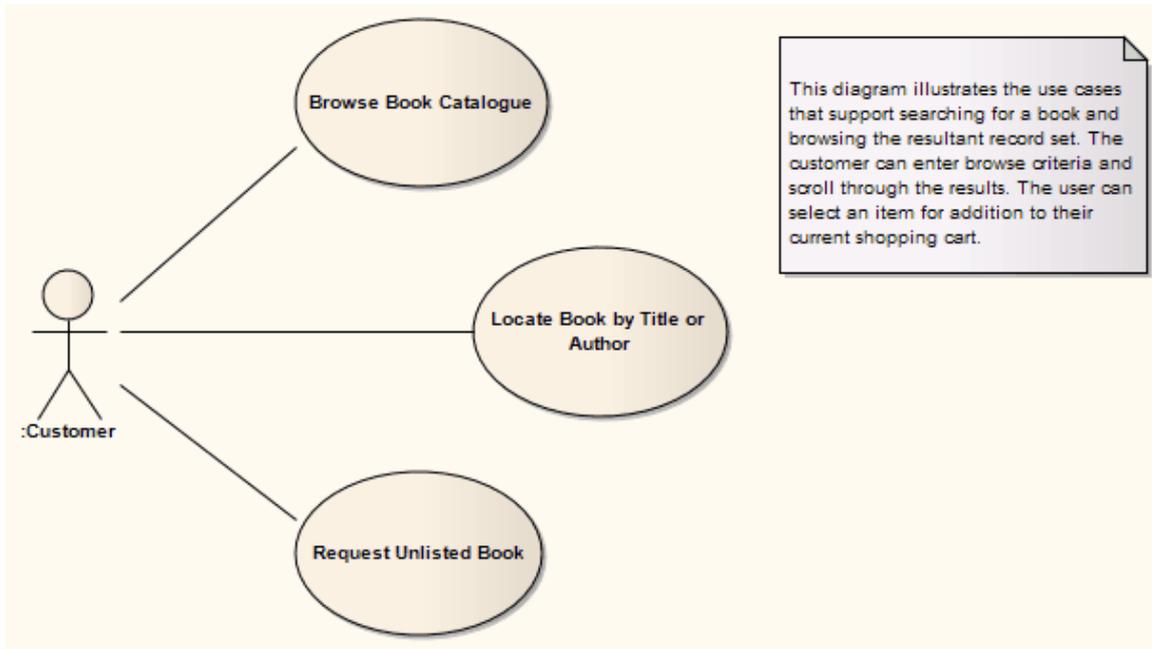
用例图捕获使用案例以及参与者与主体（系统）之间的关系。您可以使用它们：

- 描述系统的功能需求
- 描述外部事物（Actor）在系统边界交互的方式
- 描述系统的响应

您从工具箱的“用例”页面图表示用例图元素和连接器。

## 示例图表

此图说明了用例图的一些特征：



## 用例图元素工具箱图表

图标	描述
Actor	参与者是系统的用户；用户可以指人类用户、机器，甚至是模型中的另一个系统或子系统。
Use Case	用例是A UML建模元素，它描述了建议系统的用户如何与系统交互以执行离散的工作单元。
Test Case	测试用例是A刻板的用例元素，它使您能够更好地了解测试。
Collaboration	协作定义了A组协作角色及其连接器。
Collaboration Use	协作应用元素定义A模式协作应用于特定情况。

 <b>Boundary</b>	系统边界是用于定义概念边界A非元素元素
 <b>Package</b>	包用于组织您的项目内容，但是当添加到图表时，它们可以用来描述您的模型的结构和关系。

### 用例图连接器工具箱图表

图标	描述
 <b>Use</b>	A使用关系一个元素需要另一个元素来执行某些交互。
 <b>Associate</b>	关联意味着两个模型元素具有关系，通常实现为一个或两个类中的实例变量。
 <b>Generalize</b>	A概括用来表示继承。
 <b>Include</b>	Include 连接表明源元素包含目标元素的功能。
 <b>Extend</b>	Extend 连接器用于指示一个元素扩展了另一个元素的行为。
 <b>Realize</b>	A连接器表示源object实现或实现其目标object。
 <b>Invokes</b>	Invokes 连接器表明源object在某些时候会导致目标object发生。
 <b>Precedes</b>	A连接符表示源object必须在目标object开始之前完成。

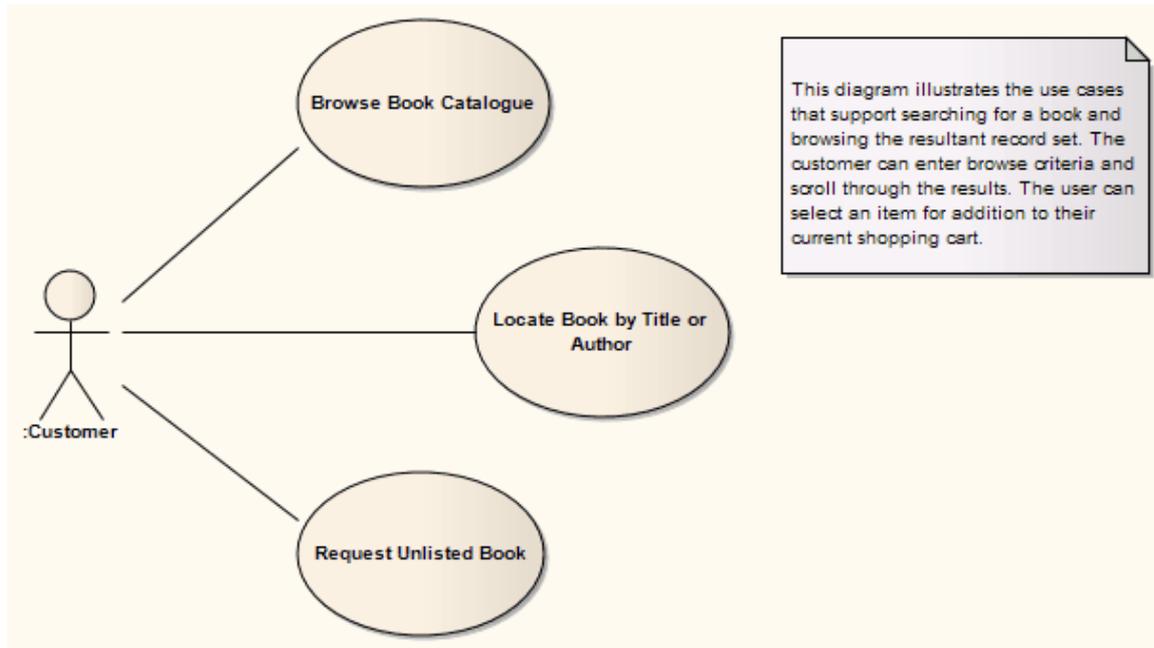
### 注记

Invokes and Precedes 依赖关系，由 OPEN建模语言定义 ( OML - 面向对象的建模过程环境，Notation进程- 是由 OPEN Consortium 开发和维护的国际事实上的标准面向对象开发方法 )。它们已被合并到用例建模元素中 )。

- Invokes 表示用例A在某些时候会导致用例B发生
- 完成表示必须先完成用例C，然后才能开始用例D

## 示例用例图表

此图说明了用例图的一些特征：



# 状态机

状态机说明元素（通常是类）如何在状态之间移动，根据转换触发器和约束守卫对其行为进行分类。您可以从工具箱的“状态”页面图表状态机元素和连接器。

## 命名

- 状态机以前称为状态图
- UML中的状态机表示基于状态状态图表，因此有时称为状态图表

## 状态表

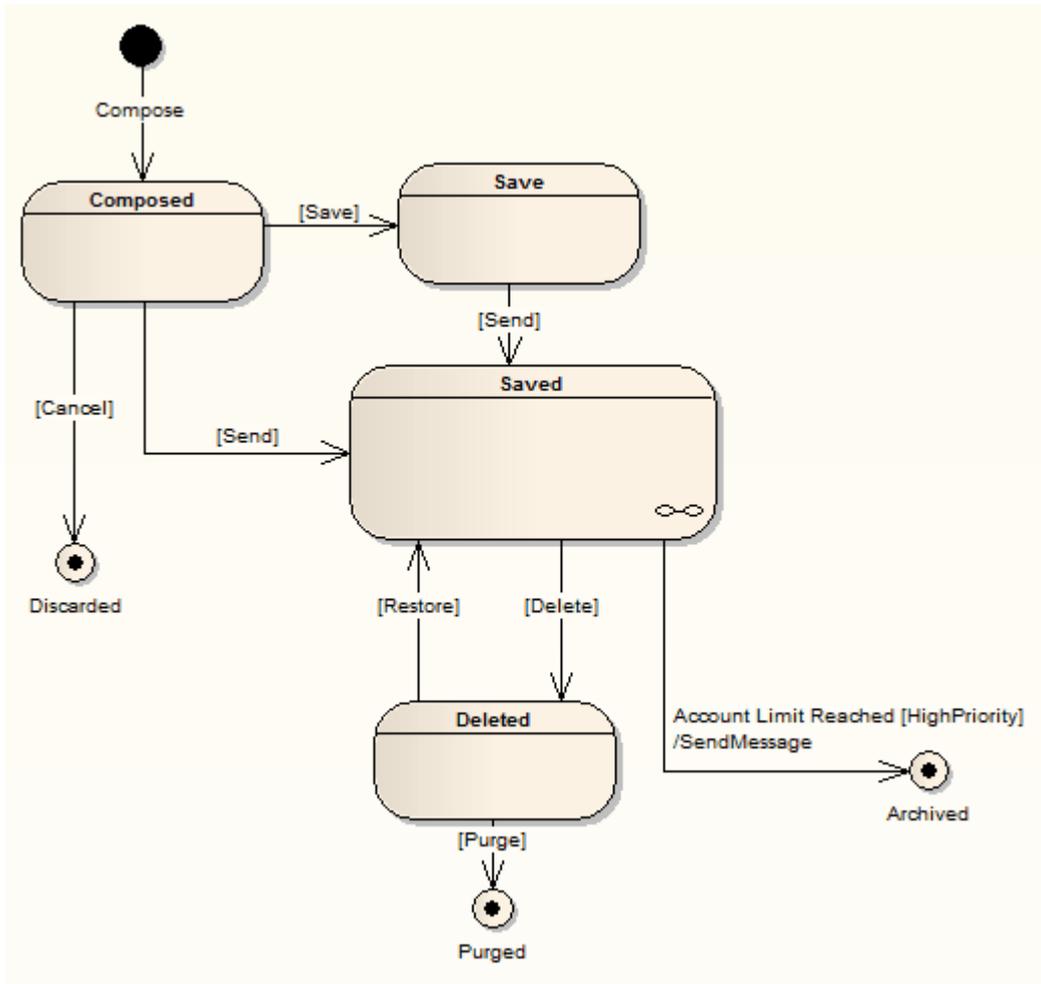
您可以将状态机显示为表或三种关系格式之一的表格。

## 选择显示格式

节	行动
1	右键单击图表背景并选择“状态图编辑器”选项。
2	选择适当的显示选项： <ul style="list-style-type: none"><li>• 图表</li><li>• 库表(状态-下个状态)</li><li>• 库表(状态-触发器)</li><li>• 库表(触发器-状态)</li></ul>

## 示例图表

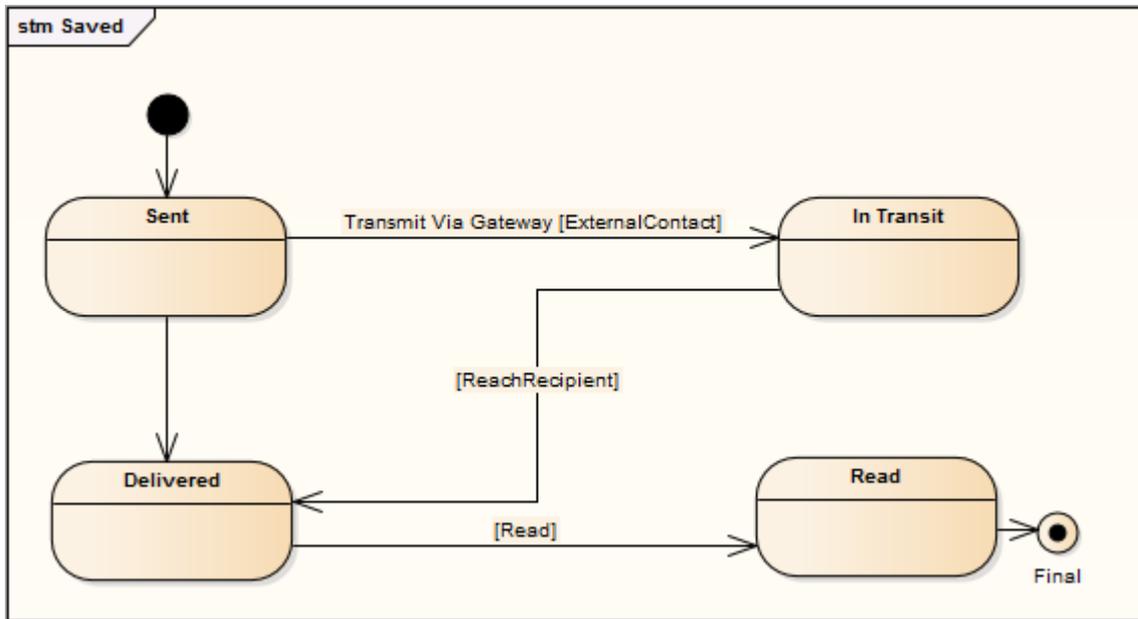
此图说明了状态机的一些特征。



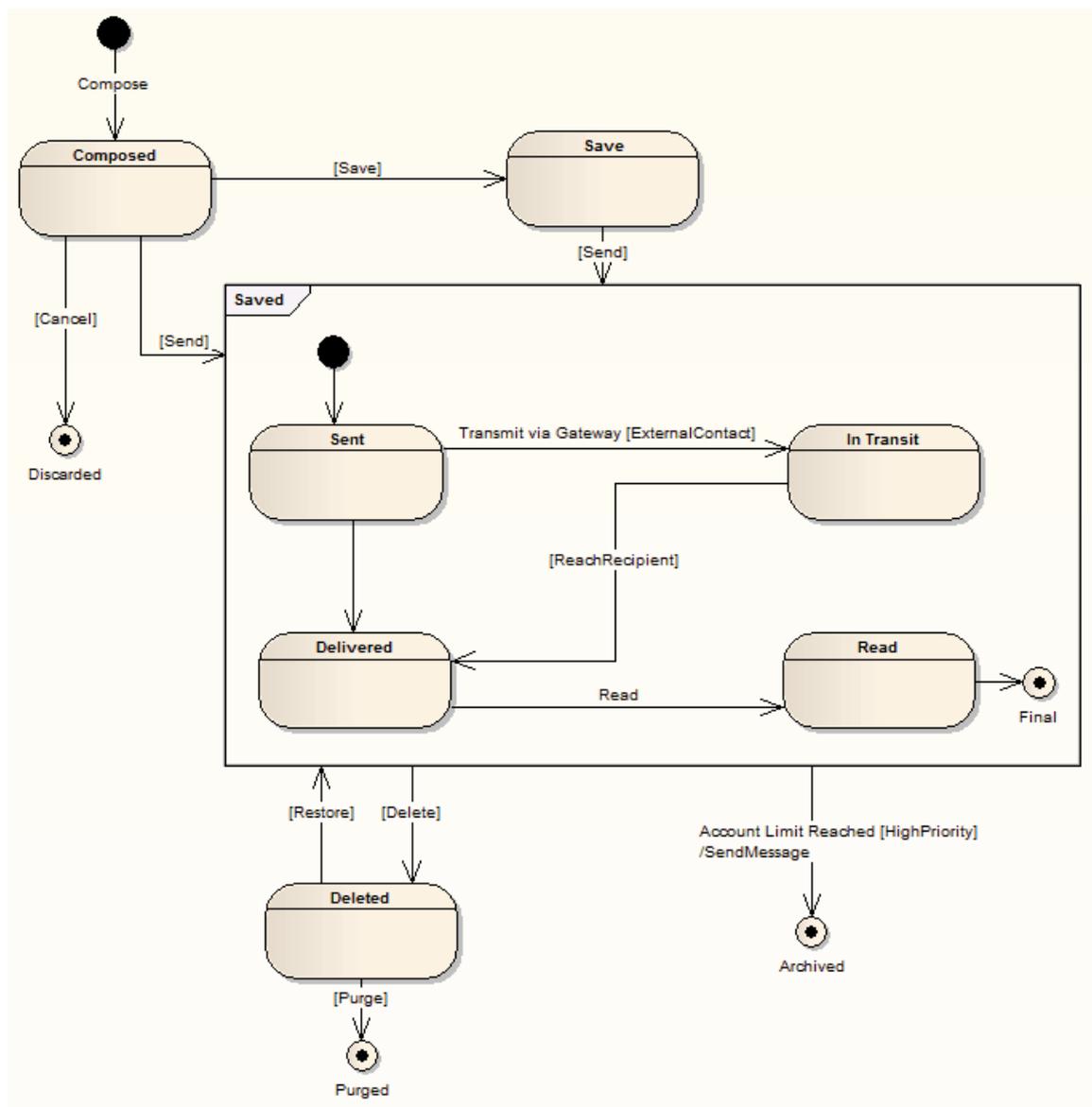
### 复合图表

已保存状态右下角的链节符号表示为状态，并带有复合图。

您有两个选项来显示状态的复合图的内容。首先，您可以双击父元素单独显示其子图，如下所示：



默认情况下，子图显示在一个带标签的框架内，该框架表示子图时间中的上下文object。您可以右键单击背景并选择“隐藏图框”选项以隐藏框架，并选择“显示图框”选项以再次显示框架。  
或者，您可以右键单击主图上的复合元素并选择“高级”。“Show复合图表”选项，它再次在一个带标签的框架中显示子图，但这次是在上下文图的时间范围内。



## 协议状态机

OMG UML规范 ( UML上层结构规范 · v2.5 · 第 14.4 节 ) 规定：

“ProtocolStateMachines 用于表达使用协议。ProtocolStateMachines 表达了事件发生的合法序列，关联的 BehavedClassifier 的行为必须遵守。状态机表示法是定义分类器的行为特征调用顺序的便捷方式。ProtocolStateMachines 可以与分类器、接口和端口相关联。”

要创建 ProtocolStateMachine，请创建状态机元素并打开该元素的属性窗口。选择 行为”选项卡，然后选择 协议状态机”复选框。图上的元素现在在元素名称的上方有一个<<protocol>>。

## 状态机图表元素工具箱Icons

图标	描述
	状态表示某种不变条件成立A情况；此条件可以是静态的（等待事件）或动态

 State	的 ( 执行一组活动 ) 。
 State Machine	状态机元素是A组相关状态元素的容器。
 Initial	起始元素的状态代表一个伪状态，表示一个复合状态；复合状态的每个区域可以有一个初始顶点。
 Final	活动终点元素表示活动的完成；到达活动终点后，图中的所有执行都被中止。
 History	UML中定义了两种历史伪状态：浅历史和深历史。
 Synch	同步状态对于指示状态机的并发路径是同步A很有用。它们用于拆分和重新加入并行处理周期。
 Object	物件是类在运行时的特定实例。
 Choice	选择伪状态用于组成复杂的过渡路径，其中传出过渡路径由动态的运行条件决定。
 Junction	连接点伪状态用于设计状态机图中复杂的过渡路径。A连接点可用于将多个路径组合或合并为共享转换路径。
 Entry	入口伪状态用于定义状态机的开始。每个区域都存在一个入口，指导初始并发状态配置。
 Exit	退出点用于状态机元素和状态机图中，表示机器退出的点以及源自该退出点的转换。
 Terminate	终止pseudostate表示一旦进入其pseudostate，状态机的执行就结束。
 Fork/Join	分叉/汇合元素A用于：1) 将单个流拆分为多个并发流，2) 连接多个并发流或3) 将多个传入流连接和分叉到多个传出流。
 Fork/Join	A分叉/汇合元素可用于： 1) 将单个流拆分为多个并发流 2) 汇合多个并发流或 3) 将多个传入流加入和分叉到多个传出流

### 状态机图表连接器工具箱Icons

图标	描述
 Transition	A转移 连接器表示状态机图中从一个状态到另一个状态的逻辑移动。
 Object Flow	一个物件流连接两个元素，通过它的特定数据，模拟一个主动过渡。

## 注记

- 状态元素可以显示有或没有横过它们的线；该行 - 如图所示 - 当元素具有诸如操作（可能被隐藏）之类的特征或在“首选项”对话框的“对象”页面中选中“显示状态分区”复选框时显示
- 可以在状态或状态机的边框上添加入口和出口元素——在图中的元素上单击鼠标右键，选择“新建子元素|入口”或“出口元素”选项；如果元素是复合元素，用框架表示，也可以在选中的框架上右击，添加入口或出口元素
- 如果您在作为另一个状态的分类器的状态机上有入口点和/或出口点，则可以从另一个状态创建对分类器的连接点状态
- 也可以将区域添加到状态元素或状态机元素框架；右键单击所选框架并选择“定义并发子状态”选项
- 您可以对状态机模型进行模型模拟，您模拟的模型可以包含多个包中的元素；要在模拟中包含外部元素，您必须创建一个包含“父”包和包含外部元素的“外部”包的包图，然后创建一个从父包到每个外部包的包导入连接器

# 伪状态

伪状态是状态机图中使用的各种类型的瞬态顶点的UML抽象。伪状态用于表示复杂的转换路径。您可以通过将这些元素图标之一拖到Enterprise Architect中的图表上来创建 Pseudostate。

## 图表工具箱

图标	描述
 Initial	起始元素的状态代表一个伪状态，表示一个复合状态；复合状态的每个区域可以有一个初始顶点。
 Entry	入口伪状态用于定义状态机的开始。每个区域都存在一个入口，指导初始并发状态配置。
 Exit	退出点用于状态机元素和状态机图中，以表示机器退出的点以及源自该退出点的转换。
 Choice	选择伪状态用于组成复杂的过渡路径，其中传出过渡路径由动态的运行条件决定。
 Junction	连接点伪状态用于设计状态机图中复杂的过渡路径。A连接点可用于将多个路径组合或合并为共享转换路径。
 History	UML中定义了两种历史伪状态：浅历史和深历史。
 Terminate	终止pseudostate表示一旦进入其pseudostate，状态机的执行就结束。
 Final	活动终点元素表示活动的完成；到达活动终点后，图中的所有执行都被中止。
 Fork/Join  Fork/Join	分叉/汇合元素A用于：1) 将单个流拆分为多个并发流，2) 连接多个并发流或3) 将多个传入流连接和分叉到多个传出流。

## 注记

- 所有列出的伪状态类型都可以在代码中表示，并且可以在Enterprise Architect 11 及更高版本的状态机代码生成模板下生成代码

# 区域

如果您正在对状态机图上的活动状态配置进行建模，并且需要将多个状态表示为同时处于活动状态，则可以通过首先创建状态机元素或复合状态元素然后将该元素细分为区域来实现此目的。您设置状态配置，以便每个区域只有一个同时活动的状态。多个转换可以从单个事件调度发生，只要类似触发的转换按区域划分。

区域显示在图表上的元素上，作为结构化隔间的细分，位于其他隔间（如标签、职责、属性和操作）下方。

## 访问

上下文菜单	右键单击元素 高级 定义并发子状态
-------	-------------------

## 在复合状态或状态机元素中创建区域

节	行动
1	在“状态”区域中，名称“”字段默认为“匿名”。
2	如果您要创建没有标题的区域，只需为每个区域单击一次保存按钮即可创建。 如果要创建命名区域，请输入名称并单击每个区域的保存按钮。
3	当您根据需要创建了尽可能多的区域后，单击关闭按钮。 您现在可以使用工具箱的“状态”页面中的图表填充区域。

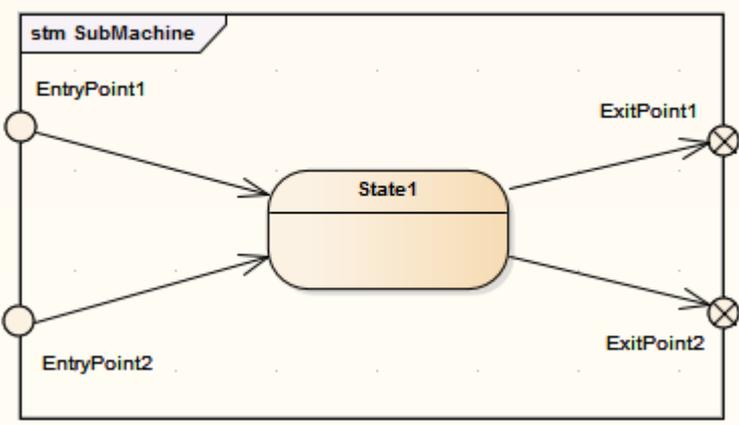
## 注记

- 保存图表时，对区域中的元素进行修改；如果要撤消更改，请重新加载图表而不保存
- 任何添加到区域的状态、状态节点（伪状态）或同步元素区域该区域所有，通常不能拖入另一个区域；但是，如果您尝试在区域之间拖动状态，则会显示“将嵌入的元素移动到区域”菜单选项 - 如果您选择它 - 允许转移到完成

# 创建连接点参考

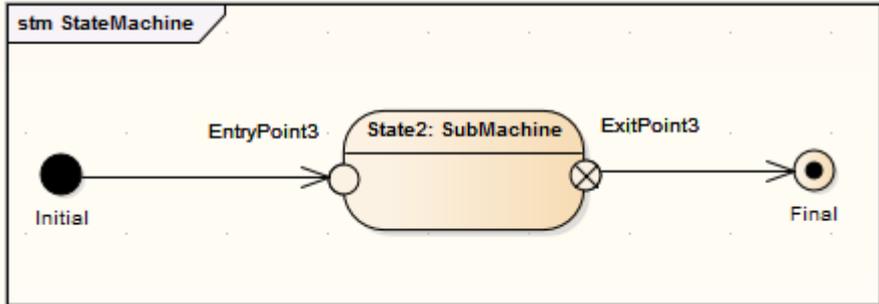
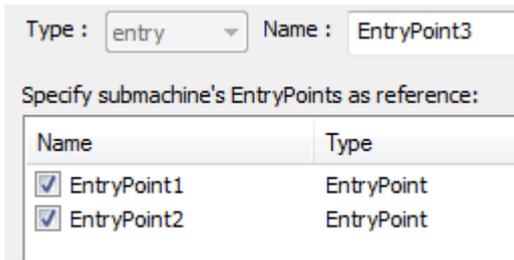
A点参考元素表示子机状态对分类器状态机中定义的入口或出口伪状态的使用。您最初创建连接点参考元素本身作为入口点或出口点。

## 创建入口点和/或出口点

节	行动
1	创建或打开分类器状态机（作为类元素的子图）。 状态机由带标签的框架表示。
2	如果入口点和/或出口点不存在，右键单击框架的内边缘并根据需要选择“新元素 入口”或“新元素 出口”选项。 相应的伪状态元素立即在帧的边缘创建。如果您愿意，可以双击元素并为其命名。
3	根据需要创建尽可能多的额外入口和/或出口元素。
4	如果相应的元素状态不存在，将状态图标从图表工具箱框架中。 在状态元素与入口和出口元素之间创建适当的连接器。
	
5	保存图表。

## 创建连接点参考

节	行动
1	创建或打开调用状态机（作为类元素的子图）。
2	如果元素尚不存在，请在图中创建适当的状态和伪状态元素和连接器。
3	单击调用状态元素并按 Ctrl+L 以显示“选择元素”对话框。

	<p>从 “创建入口点和/或出口点”阶段浏览并选择分类器状态机。</p>
<p>4</p>	<p>右键单击状态元素，并根据需要选择 “新元素 入口”或 “新元素 出口”选项。 相应的伪状态元素立即在元素的边界上创建。</p> 
<p>5</p>	<p>双击入口元素。 将显示 “编辑 ConnectionPointReference”对话框。</p>
<p>6</p>	<p>如果您愿意，在 “名称”字段中为所选入口输入一个新名称。 在 “指定子机的入口作为参考”面板中，选中每个分类器入口点的复选框以创建参考。您可以选择多个复选框。</p> 
<p>7</p>	<p>点击确定按钮。</p>
<p>8</p>	<p>如有必要，对状态元素的出口重复步骤 4 至 7。</p>

# 状态机库表

状态机表是状态状态机A两个变体之一（另一个是状态机图）。它以表的形式显示状态机的信息，是一种指定有限状态转移系统离散行为的方法；即状态机移动到什么状态以及发生转换的条件。

## 访问

上下文菜单	右键单击背景状态机图   状态图编辑器   库表 (选项)
-------	-------------------------------

## 状态机库表显示

您可以在表中将状态转换显示为两种不同类型的关系之一：

类型	描述
状态——触发器	行表示当前状态，列表示触发事件。 如果触发生，则行和列交叉处的单元标识转换中的目标状态，以及转换的条件（或保护），或者如果您愿意，可以触发器，以简单的状态格式。
-状态- 下一个状态	行和列都表示状态，行列交叉处的单元表示： <ul style="list-style-type: none"> <li>• 触发从当前（行）状态到下一个（列）状态转换的事件</li> <li>• 事件的条件（或守卫），以及</li> <li>• 过渡效果</li> </ul>

## 选择显示格式

节	行动
1	右键单击图表背景并选择“状态图编辑器”选项。
2	选择适当的显示选项： <ul style="list-style-type: none"> <li>• 图表</li> <li>• 库表(状态-下个状态)</li> <li>• 库表(状态-触发器)</li> <li>• 库表 ( 触发器-状态 )</li> </ul>

# 状态机库表Options

您可以从 状态机图表：选项”对话框中选择状态机表布局并设置其他选项，您可以通过以下任一方式显示该对话框：

- 双击状态机表背景或
- 右键单击背景并选择 状态库表选项”选项

## 选项

选项	行动
库表格式	<p>选择所需的表格式。</p> <p>状态——触发器：</p> <ul style="list-style-type: none"> <li>• 行代表状态，每个状态名称在一个左边缘单元</li> <li>• 列代表触发器，列标题中的每个触发器单元</li> <li>• 行和列的交点标识转移（如果有的话）</li> <li>• 转移单元显示转移的下一个状态和条件（守卫）的信息转移</li> </ul> <p>——状态：就状态触发器——触发器，只不过行代表触发器，列代表状态。</p> <p>状态下一个状态：</p> <ul style="list-style-type: none"> <li>• 行和列都代表状态</li> <li>• 行和列的交点定义了从行状态到列状态的过渡（如果有的话）</li> </ul>
单元格大小	完全接下来的四个领域。
转移 单元格宽度	指定过渡单元的宽度（即列宽）。
转移 单元格高度	指定过渡单元的高度（即行高）。
左边缘单元格宽度	指定左边缘（行标题）单元的宽度。
顶边单元格高度	指定顶部边缘的高度（列标题）单元。
单元格颜色	完全接下来的三个领域。
状态/触发器细胞	选择行列标题的单元。
状态枚举触发器	<p>选择枚举的颜色（行/列编号）单元。</p> <p>您必须至少选择 启用状态枚举”和 启用事件枚举”复选框之一来设置此颜色。</p>
转移 细胞	选择过渡的单元（在表的主体中）。

突出显示选项	
突出显示与选定的转移相关的区域转移	突出显示所选转换中涉及的所有元素的单元- 初始状态、目标状态和触发器。
高亮颜色	选择突出显示的颜色。
使用不同颜色的状态目标	在与源元素的单元不同的颜色中单元显示目标元素的单元。
目标区域颜色	选择突出显示的颜色。
显示选项	
始终显示空状态区	在表的末尾添加一个空行 ( 并且, 在状态下一个状态表上, 一个空列 ) 。 单元包含一个  按钮。您可以在按钮上单击两次 ( 不是双击 ) 来编辑单元并识别新的状态。在这种情况下, 会自动添加另一个空状态区。
启用状态枚举	为每个状态标题单元一个单元, 为状态状态。编号从 0 开始。
前缀	如果需要, 请键入状态编号的前缀或删除默认的 \$ " 以使其没有前缀。
启用事件枚举	为每个事件或触发器标题单元一个单元, 为事件编号。编号从 0 开始。
前缀	如果需要, 请键入事件编号的前缀或删除默认的 E 以使其没有前缀。
样品状态库表	显示您定义的表格式的预览。
高级	定义图表选项。显示状态机图表 属性"对话框。
恢复默认值	重新状态库表图默认值。
申请	将更改后的选项应用到状态库表图。

# 状态机库表操作

由于状态机表是状态机图的一种变体，所以操作数据的大部分操作与状态机图相同。这些主题中描述了特定于状态机库表的操作：

## 操作

手术
更改状态机库表位置
更改状态机库表大小
插入新状态
插入触发器
插入/更改转换
重新状态或状态触发器细胞
添加图例
在状态机图中定位 Cell
状态机库表约定
导出状态库表转CSV文件

## 更改状态机库表位置

如有必要，您可以在图表视图中移动状态机表。

### 更改状态机表的位置

节	行动
1	按Ctrl+A或双击左上角单元选择整个状态机表。
2	将状态机表拖放到需要的位置。 或者，使用 Shift+右箭头、左箭头、上箭头或下箭头移动状态机表。

## 更改状态机库表大小

改变状态机表大小的三种方式：

- 更改 状态机图表：选项”对话框上的单元大小
- 按Ctrl+A或双击左上角单元选择整个状态机表，然后按Ctrl+'Left'、'Up'、'Right'或'Down'改变大小
- 选择状态机表，然后拖动形状手柄改变大小

# 插入触发器

如果状态机表格式为状态-触发器或触发器-状态，您可以使用这些方法中的任何一种来插入新的触发器：

## 方法

节	行动
1	在状态机表单元，将光标移至“事件”一词，词尾显示+；单击+以创建一个新的触发器。
2	在状态机表左上角的单元中，右键单击并选择“添加触发器”选项以创建新的触发器。
3	在状态机表中选择一个现有的触发器，然后按 Insert 键在现有的触发器之前插入一个新的触发器。
4	单击状态机表中的现有触发器，右键单击并选择： <ul style="list-style-type: none"><li>• 'Insert New触发器' 选项可在当前简单之前插入一个新的触发器触发器，或</li><li>• 'Insert New触发器' 选项在当前触发器之后插入一个新触发器</li></ul>

# 插入/更改转移

本主题说明如何在两个状态元素之间插入或修改过渡链接。

## 选项

行动	描述
插入一个新的转移	<p>您可以插入一个新的转移使用其中一种方法。</p> <p>右键单击要在其中创建转移的单元转移</p> <p>:</p> <ul style="list-style-type: none"> <li>如果状态机表格式为状态-触发器或触发器-状态，则上下文菜单列出您可以选择作为转移目标的状态转移;单击所需的名称以创建转移</li> <li>如果状态机表格式为状态-下个状态，单击 转移 "转移" 上下文菜单选项来创建转移</li> </ul> <p>或者，在工具箱的 "状态关系" 页面中，选择转移元素，然后单击在其中的单元转移状态机表中创建转移;双击转移 在'转移' 中定义它转移 属性"对话框。</p>
更改转移	<p>图表状态图，改变一个转移的属性转移</p> <p>双击 转移 "转移" 单元和编辑'转移' 的细节转移 属性"对话框。</p>
更改转移状态	<p>您可以更改转移的源和目标转移</p> <p>通过右键单击转移并选择'高级 设置源和目标'选项。</p> <p>或者，您可以更改转移源、目标或触发器单击转移并将其拖到不同的单元中。</p>

	<p>如果状态机表格式为状态-触发器或触发器-状态，则可以更改状态的转移经过：</p> <ol style="list-style-type: none"> <li>1. 在转移中突出显示目标状态名称转移单元并单击它以在表中显示状态列表。</li> <li>2. 单击首选目标状态名称。</li> </ol>
<p>高亮状态和与触发器相关的转移</p>	<p>您可以选择选项以突出显示与单元相关转移的状态源、目标状态和触发器转移。</p> <ul style="list-style-type: none"> <li>· 使用“状态机图表：选项”对话框上的“突出显示选项”面板。</li> </ul> <p>当你点击转移单元与其相关的状态和触发器的单元被突出显示。</p> <p>或者，点击转移单元并按住L键。</p>

# 插入新状态

## 选项

行动	描述
<p>在状态机表中插入新状态</p>	<p>您可以使用以下方法之一在状态机表中插入新状态：</p> <ol style="list-style-type: none"> <li>1. 在状态机表单元，将光标移动到单词状态，单词末尾会显示一个+；点击+创建新状态</li> <li>2. 右键单击单元状态机表左上角并选择“添加状态”</li> <li>3. 右键单击单元状态机表中的现有状态并选择：                     <ul style="list-style-type: none"> <li>- 'Insert New状态' 在之前插入新状态</li> <li>当前状态，或</li> <li>- 'Insert New状态' 在当前状态之后插入新状态</li> </ul>                     状态                 </li> <li>4. 在状态机表中点击一个已有状态，按单元键在选中的单元状态创建并插入一个新状态</li> <li>5. 在工具箱的“状态元素”页面，点击一个元素，然后点击：                     <ul style="list-style-type: none"> <li>- 图表背景添加新状态到末尾</li> <li>表，或</li> <li>- 一个现有的状态单元新的状态就在它上面</li> </ul>                     在工具箱的“状态元素”页面中，您可以插入状态、初始、终点、进入、退出和终止元素。                 </li> </ol>
<p>将子态状态添加到选定的状态</p>	<p>为选中的状态单元添加子态状态，在状态机表中右击需要的状态单元，选择“添加子态状态”；Enterprise Architect将子态状态添加到状态。</p> <p>如果所选状态不允许子态状态，则“添加子态状态”选项将显示为灰色。</p> <p>您还可以将一个现有状态拖到另一个上；如果第二个状态允许子状态，则被拖拽的状态就变成了它的子态状态。</p> <p>同样，您可以通过将子态状态从原始父状态拖动到不同的状态来更改子态状态的父状态。</p>
<p>移除子态状态的父关系，使其成为一个独立的状态</p>	<p>删除子态状态的父关系，使其成为一个单独的状态，在状态机表中右击子态状态，选择“删除父关系”；子态单元状态成为一个状态单元。</p> <p>您也可以将子态状态拖放到状态机表左上方的单元上；被单元的子态又变成了一种状态。</p>

## 重新状态或状态触发器细胞

您可以通过以下方式之一更改选定状态或触发器单元的位置：

- 右键单击状态或触发器的单元并选择适当的 订单 |移动 xxx' 选项
- 单击单元并按 Shift+右箭头、左箭头、上箭头或下箭头

## 添加图例

您可以为任何没有过渡的状态机表单元格添加一个简单的图例。两个图例符号是：

- I - 忽略
- N - 永远不会发生

### 将图例符号分配给状态机表单元格

节	行动
1	单击要为其分配图例的单元，然后按： <ul style="list-style-type: none"> <li>• 插入“忽略”图例的I键，或</li> <li>• N键插入“Never Happen”图例</li> </ul> 所需的符号显示在单元的中心。

### 或者

节	行动
1	右键单击要为其分配图例的单元。
2	选择适当的上下文菜单选项： <ul style="list-style-type: none"> <li>• 图例 忽视</li> <li>• 图例 永远不会发生</li> </ul> 所需的符号显示在单元的中心。

### 注记

- 要从单元中删除单元符号，可以：
- 单击单元并按删除，或
- 右键单击单元并选择图例|删除图例

# 在状态机图表中查找 Cell

## 在状态图中图表

在状态机表上，要在状态机图中找到选定的状态或触发器元素：

- 选择 **查找 | 在状态图表中定位**

Enterprise Architect切换到状态机图并突出显示选定的元素。

你可以找到一个转移

以类似的方式，通过选择 **在状态图中图表**来建立关系。

触发器，状态机表上的A可能存在也可能不存在于相应的状态机图上；如果状态机触发器上不存在，则禁用 **在状态图表中图表**选项。

## 在状态库表中找到

在状态机图上，在相应的状态机表中找到选中的状态或触发器元素：

- 选择 **查找 | 定位在状态库表**

Enterprise Architect切换到状态机表并突出显示选定的元素。

你可以找到一个转移

以类似的方式，通过选择 **在状态库表中定位**来建立关系。

# 状态机库表约定

## 触发器

- 删除触发器模型会将其完全从模型中删除，因此您无法撤消触发器的删除
- 在事件标题行的末尾有一个 <None> 列；这是针对没有触发器信息的转换

## 状态

在工具箱，您只能插入这些状态元素类型（虽然状态机表可能会选择并显示其他类型，例如子机状态）：

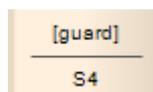
- 状态
- 最初的
- 终点
- 入口
- 出口
- 终止

## 转移

### 转移

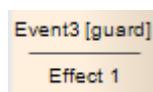
单元以两种方式之一显示其属性，具体取决于状态机表格式；如果机器状态机表的格式是状态-触发器或者触发器-状态，转移

单元显示守卫条件和目标，如图：



如果状态机表格式为状态-下一个状态，则转移

单元以这种格式显示触发器、守卫条件和影响：



在状态机表中，可以就地编辑守卫条件和影响。如果您选择的转移的守卫条件或影响为空转移

单元，该单元显示一个省略号（...）；在省略号上单击两次（不是双击）以输入守卫条件和影响名称。

## 导出状态库表转CSV文件

导出一个机器状态机库表到一个CSV文件

节	行动
1	打开所需的状态机库表。
2	右键单击图表背景并选择“导出Statechart to CSV file”选项。 将显示“另存为浏览器”对话框。
3	选择适当的目录位置并输入 .csv 文件名。
4	单击保存按钮。

## 示例状态-触发器库表

行表示当前状态，列表示触发事件（或者如果您愿意，也可以触发器，以简单的状态格式表示）。  
 如果触发发生，则行和列交叉处的单元标识转换中的目标状态，以及转换的条件（或保护）。

State \ Trigger		Event1	Event2	Event3	Event4	<None>
		E0	E1	E2	E3	E4
Initial	S0					S1
	S1				S2	
State2	S2	S6	$\frac{[\text{Guard}]}{S4}$			
	SubState1	S3	S4			
	SubState2	S4		$\frac{[\text{Cond}]}{S2}$		
	SubState3	S5				
State3		S6				S7
Final		S7				

# 示例状态-下个状态库表

行和列都表示状态，行列交叉处的单元表示：

- 触发从当前（行）状态到下一个（列）状态转换的事件
- 事件的条件（或守卫），以及
- 过渡效果

Next State		Initial	State1	State2			State3	Final	
				SubState1	SubState2	SubState3			
State		S0	S1	S2	S3	S4	S5	S6	S7
Initial	S0		_____						
State1	S1			Event4 _____					
State2	S2					Event2 [Guard] _____		Event1 _____	
	SubState1	S3				Event2 _____			
	SubState2	S4			Event3 [Cond] _____				
	SubState3	S5							
State3	S6							_____	
Final	S7								

# 状态机库表仿真

状态机A库表是状态状态机的表示，可以用与状态机图完全相同的方式进行模拟。

## 访问

使用库表中显示的状态机，使用此表中列出的任何方法开始模拟。

功能区	仿真>运行仿真>开始，或 仿真>动态仿真仿真> 模拟器> 打开模拟器窗口>  (开始icon)
上下文菜单	右键单击查看背景  执行仿真  <口译或手动>

## 突出显示活动单元

随着模拟的执行，表单元改变颜色以指示：

- 当前活动状态- 在 “状态机选项”对话框的 “突出显示颜色”字段中设置的颜色，以及深色边框
- 潜在的下一个状态- 突出显示颜色”字段中颜色A变体，或者，如果在 “状态机选项”对话框中选中 “使用不同颜色用于目标状态”复选框，则在 “目标”中设置颜色区域颜色'字段
- 主动转移  
(s) - '转移  
中设置的颜色转移  
“状态机选项”对话框的 “单元格”字段
- 触发器 (s) - 在 “状态机选项”对话框的 “突出显示颜色”字段中设置的颜色
- 非活动状态 - 灰色

例如：

		Trigger		CardInserted	Finished
		State		E0	E1
Initial		S0			
Idle		S1	S2		
		S2			S1

## 触发器信号

与以图表形式运行模拟时一样，模拟将自动遍历没有保护或经过验证的保护的转换。除非那个触发器已经被触

发，否则不会跟随带有触发器的转换。它们可以从仿真事件窗口自动触发，也可以通过右键单击转移来手动触发器转移或触发器单元选择 信号触发器仿真。

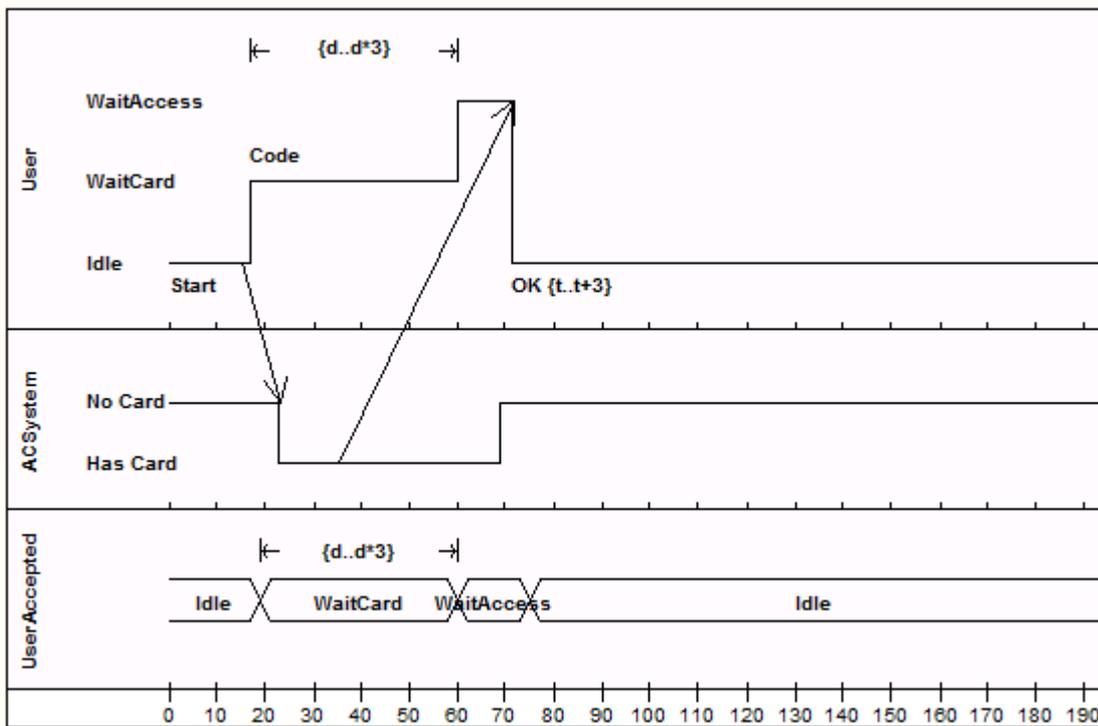
## 图表

时序图定义了不同对象在时间范围内A行为。它提供对象随时间变化状态和交互的可视化表示。您可以使用它来：

- 定义硬件驱动或嵌入式软件组件；例如，用于燃料喷射系统或微波控制器的那些
- 指定时间驱动的业务流程

您可以从图表工具箱的“图表”页面生成时序图元素和连接器。

### 示例图表



### 时序图元素工具箱图表

图标	描述
State Lifeline	状态生命线元素表示object在一段时间内A状态，使用 y 轴的变化来表示状态之间的离散转换。
Value Lifeline	价值生命线元素表示object在一段时间内A状态，使用平行线表示稳定状态，沿 x 轴。
Message Label	A信息标签是表示生命线之间消息的另一种方式，这对于整理“散布着消息的时序图很有用。
Message Endpoint	A信息元素表示一个信息： <ul style="list-style-type: none"> <li>• 终止于状态或价值生命线之外的未定义点，开始于生命线内的确定点，</li> </ul>

	<p>或</p> <ul style="list-style-type: none"> <li>起源于状态或价值生命线之外的未定义点，终止于生命线内的确定点</li> </ul>
	<p>A图表门元素表示一个信息：</p> <ul style="list-style-type: none"> <li>终止于状态或价值生命线外的定义点，开始于生命线内的一个确定点，或</li> <li>起源于状态或价值生命线外的定义点，终止于生命线内的确定点</li> </ul> <p>定义的图表门是交互到该碎片信息的边界，表明该碎片是从该碎片发出或传递到该碎片的边界。</p>

### 图表图连接器工具箱图标

图标	描述
	<p>消息指示元素之间的信息流或控制转换。</p>

# 创建一个图表

## 创建时序图

节	行动
1	右键单击浏览器窗口中的一个包，然后选择“添加图表”。 将显示“新图表”对话框。
2	在“选择自”面板中，选择“UML行为”。
3	在“图表类型”面板中，选择“时间”。
4	点击确定按钮。 显示图表视图，您可以在其上为图表创建分类计时元素。

## 设置时间范围

### 在将生命线元素添加到时序图之前设置时间范围

节	行动
1	右键单击图表并选择“设置时间线范围”。 将显示“设置时间线范围”对话框。
2	在“开始时间”和“结束时间”字段中，输入时间线起点和终点的数值；例如，将范围设置为 0 到 100。 开始时间必须小于结束时间。
3	在“时间单位”字段中，输入测量时间的单位；例如，秒或分钟。
4	如果不需要在图表上显示时间范围，请选中“Suppress In 图表”复选框。
5	点击确定按钮。 如果您没有抑制它，时间范围将显示在您在图表上创建的生命线元素下方。

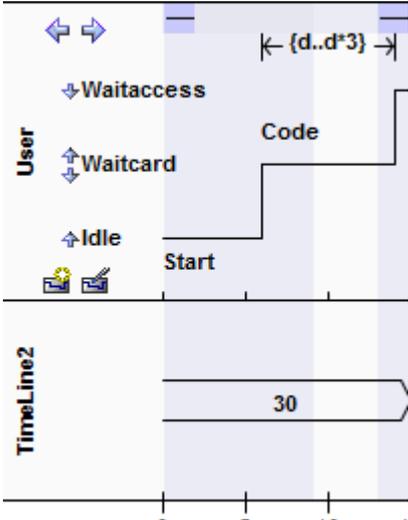
## 编辑图表

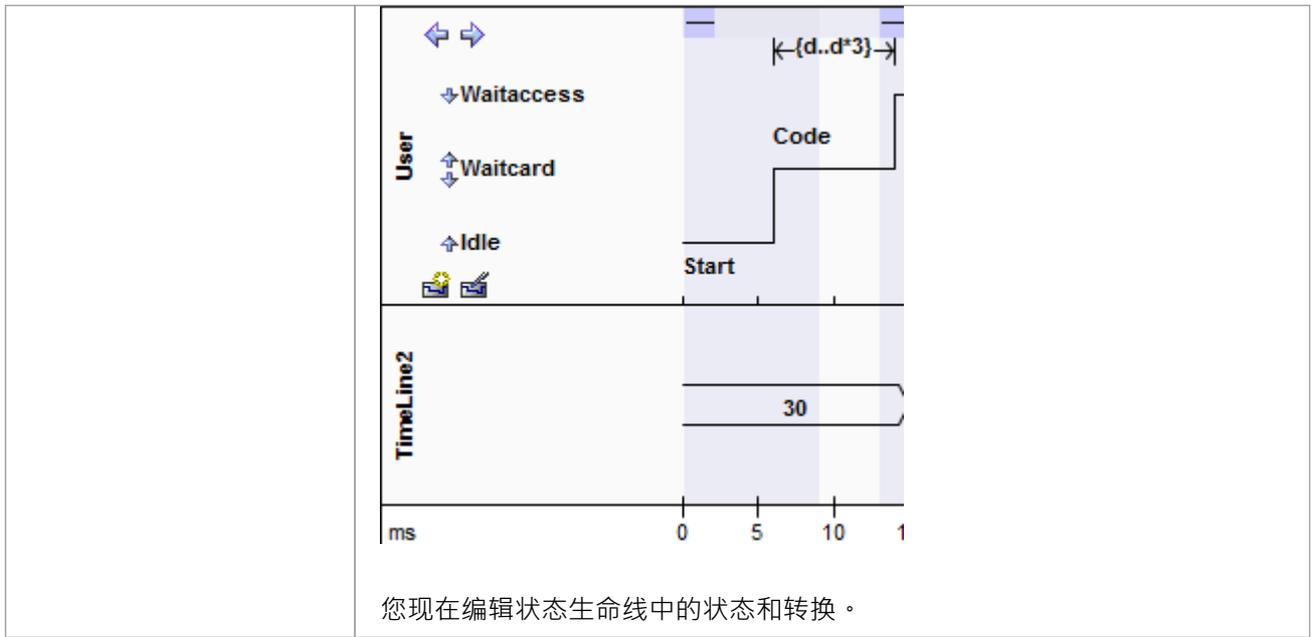
在时序图上，您可以添加状态生命线元素和值生命线元素。您可以在图表本身或通过“配置时间线”对话框维护这些生命线元素的状态和转换。

# 添加和编辑状态生命线

从计时元素的 组件”页面中，将状态生命线工具箱拖到图表上。元素显示在图表上。

## 编辑属性

任务	行动
定义状态生命线的名称	<ol style="list-style-type: none"> <li>1. 右键单击元素并选择 属性 属性”选项；'元素的属性窗口显示，显示'元素'选项卡。</li> <li>2. 改写 名称”字段。</li> <li>3. 单击关闭属性窗口。</li> </ol>
大小和规模	<p>在选定的生命线元素的左上角是左右快速调整大小按钮 (↔↔)。</p> <p>这些按钮增加或减少生命线元素的宽度，进而控制每个时间单元的刻度宽度；通过增加元素的宽度，您可以在添加过渡时提高分辨率，从而使它们更易于编辑。</p> <p>为了编辑状态生命线元素，您必须单击它来选择它。</p>
设置时间线开始位置	<p>您可能在时间线开始时需要更多空间；例如，使用长状态名称。</p> <p>要在图表的所有时间线中插入更多空间：</p> <ol style="list-style-type: none"> <li>1. 右键单击图表背景并选择 设置时间线开始位置”选项；将显示 设置时间线开始位置”对话框。</li> <li>2. 'Value 80 to 300' 字段默认为 80 作为时间线元素开始和时间线本身开始之间的最小距离（以像素为单位）；输入一个高达 300 像素的新值，然后单击确定按钮以增加时间线开头的空间。</li> </ol> <p>这两个图的起始位置分别为 80 像素和 150 像素。</p>  <p>ms      0      5      10      15</p> <p>线80.png 时间线</p>



# 将状态添加到状态生命线

## 将状态添加到状态生命线

节	描述
1	<p>单击状态生命线元素。</p> <p>新状态按钮 (newstatebutton.png) 和编辑状态按钮 (  ) 显示在元素的  。</p>
2	<p>单击新建状态按钮。</p> <p>将显示 新状态"对话框。</p>
3	<p>在 状态"字段中，输入状态的状态。</p>
4	<p>点击确定按钮。</p> <p>您必须添加至少两个状态；例如，开"和 关"。</p>
5	<p>添加状态时，通过拖动元素边缘的  图标之一来增加元素的高度。</p> <p>您还可以使用 配置时间线"对话框的 状态"选项卡添加状态。</p> <p>添加：</p> <ul style="list-style-type: none"><li>• 时间线的离散状态，或</li><li>• 连续A数字状态范围</li></ul>

# 在状态生命线上编辑状态

## 在状态生命线上编辑状态

节	描述
1	点击状态生命线元素，点击需要的状态。 将显示“编辑状态”对话框。
2	在“状态”字段中，根据需要更改名称。
3	点击确定按钮。
4	如有必要，通过以下任一方式更改状态顺序： <ul style="list-style-type: none"><li>单击每个状态名称旁边的向上或向下箭头 (↕)，或</li><li>右键单击状态名称并选择“上移”或“下移”选项</li></ul> 您还可以使用“配置时间线”对话框的“状态”选项卡编辑状态。

# 删除状态生命线中的状态

## 删除状态生命线中的状态

节	描述
1	右键单击状态名称并选择“删除”选项。

## 或者

节	描述
1	单击状态生命线元素。
2	按住 Ctrl 并将光标移到状态名称上。 光标改变形式 (  )。
3	单击鼠标按钮。 状态名称被删除。

## 在状态生命线编辑转场

在时序图中，您可以显示状态机在固定时间段内和特定时间点发生的转换（状态变化）。这类似于交互在很多方面的状态变化。当事件和变化发生在这个时序图表示的实例中时，状态变化就会发生并映射到这个时间轴上。在这方面，它是系统特定方面如何随时间变化的记录。

在构建时间线时，有必要首先定义状态，然后在特定时间点添加这些状态之间的显式转换。

### 编辑转场

任务	行动
添加和移动过渡	添加状态后，您可以使用鼠标直接在时间轴上添加状态之间的转换。
更改转移时间	将光标移到一条或另一条垂直过渡线上并向左或向右拖动该线以更改过渡时间。 在线时，光标形状变为水平移动光标 (  )。
合并过渡	如有必要，您可以“推送”一个转换，以将其与图表上任何生命线元素上的下一个或上一个转换点合并。 将光标定位在过渡线的适当一侧；光标改变形式 (  或  )。 单击鼠标按钮；系统在图上的任何元素上定位所需方向上最近的转换，并将当前转换与该转换合并。
删除转场	当您将过渡移动到与前一个过渡状态相同的状态并释放光标时，过渡会自动删除。 或者，右键单击过渡线并选择“删除”选项。

## 添加和移动过渡

添加状态后，您可以使用鼠标直接在时间轴上配置状态更改（转换）。这是建立状态随时间变化的详细模型的一种快速有效的方法。

要修改时间线，请将鼠标放在现有时间线上。当您光标移到时间轴上时，光标将变为三种形状之一，如下所述。

### 访问

上下文菜单	右击过渡线 编辑
其它	直接点击相应的过渡线，过渡开始后

### 修改时间线

当您光标移到转换的垂直线上时，转换发生的时间会显示在该行旁边。

任务	行动
移动光标 	当它直接位于时间轴上方时显示。 按住鼠标键并拖动线条，将时间线移动到状态位置上方或下方的状态；如有必要，您可以向上或向下移动多个状态的转换。
新的向上过渡光标 	当它刚好在时间线下方时显示，并且该线上方还有另一个状态。 按住 Alt 键；光标更改 (  )。 单击以创建到行上方状态的新过渡。 要向上推动多个状态的转换，请将光标移动到该线上并将其向上拖动。 过渡是针对一个间隔单位的；您可以通过更改过渡时间使其更长。 如果您不按住 Alt 键，则光标不会更改，并且从过渡开始的整个时间线向上移动。
新的向下过渡光标 	当它刚好在过渡线上方，并且在该线下方有另一个状态时显示。 按住 Alt 键；光标更改 (  )。 单击以创建到该行下方状态的新过渡。 要向下推动多个状态的转换，请将光标移动到该线上并向下拖动。 过渡是针对一个间隔单位的；您可以通过更改过渡时间使其更长。 如果您不按住 Alt 键，则光标不会更改，并且从过渡开始的整个时间线会向下移动。

### 编辑转移

根据需要编辑过渡，在 **编辑转移**  
“转移  
'对话框。

选项	行动
在时间	类型转换发生的时间尺度上的点。
转移 至	类型发生转换的状态的名称。
事件	类型转换所代表的事件的名称。 这显示在时间轴元素上，就在过渡线的上方。
期限约束	类型对过渡持续时间的任何约束。 这显示在时间轴元素上，沿着过渡元素的顶部。
时间约束	类型对转换开始的任何约束。 这会在过渡开始时显示在时间轴元素上。
确定	单击此按钮以保存更改。

## 注记

- 一旦事件、持续时间约束或时间约束显示在图表上，您可以通过单击它们直接编辑它们以显示它们的特定对话框
- 您可以通过在单击它们时按住 **Ctrl** 键来删除它们；按下 **Ctrl** 键时光标会改变形状
- 您还可以使用 **配置时间线**对话框的 **过渡**选项卡编辑过渡

## 添加和编辑值生命线

从工具箱“价值生命线”元素拖到图表上。元素显示在图表上。

### 编辑价值生命线名称

节	行动
1	右键单击元素并选择“属性 属性”选项。 时间轴元素的属性窗口显示“元素”选项卡。
2	改写“名称”字段。
3	单击关闭属性窗口。

### 大小和规模

在选定的生命线元素的左上角是左右快速调整大小按钮 ()。这些按钮增加或减少生命线元素的宽度，进而控制每个时间单元的刻度宽度。通过增加元素的宽度，您可以在添加过渡时提高分辨率，从而使它们更易于编辑。

## 在价值中添加国家生命线

向 Value 生命线添加状态类似于向状态生命线元素添加状态。

对于价值生命线，只有第一个状态显示在图表上。其他状态被添加到列表中，以便在创建转换时访问；它们仅在您创建到这些状态的转换时显示在生命线元素上。

您只能使用“配置时间线”对话框的“状态”选项卡编辑或删除值生命线元素中的状态。

# 编辑价值生命线过渡

## 通过图表将转换添加到 Value 生命线元素上的状态

节	行动
1	将光标移到过渡线上方。 光标改变形式 (  )。
2	单击鼠标按钮。 新的转移 显示 "事件" 对话框。
3	在 "转移 To" 字段，单击下拉箭头并从可用状态列表中选择一状态；这显示在转换框内的生命线元素上。 对话框中的其余字段是可选的。
4	在 "事件" 字段中，键入转换所代表的事件的名称；这显示在生命线元素的正下方和过渡线的开始处。
5	在 "持续时间约束" 字段中，输入对过渡持续时间的任何限制；这会显示在生命线元素上，沿着元素顶部的过渡。
6	在 "时间约束" 字段中，输入过渡开始时的任何限制条件。 这显示在过渡开始处的生命线元素上，就在事件名称之后。
7	单击确定按钮以创建新的过渡。

## 编辑一个转移

节	行动
1	单击过渡中的状态名称。 或者，右键单击状态名称并选择 "编辑" 选项。 '编辑转移 '对话框显示，这与 '新转移 相同转移 事件"对话框，但启用了 "时间" 字段。
2	如有必要，改写 "在时间" 字段以定义不同的起点。 您不能更改时间线中第一个状态的 "在时间" 字段；这始终是 "0"。

3	根据需要编辑其余字段。
4	单击确定按钮保存更改。

## 更改过渡时间

节	行动
1	要更改过渡的开始或结束时间，请单击过渡的开始或结束点并将其拖动到新位置。 在线时，光标形状变为水平移动光标 (  )。

## 删除转场

节	行动
1	要删除转换，请按住 Ctrl 并单击转换状态名称。 当您在转换状态名称上按住 Ctrl 时，光标会更改形式 (  )。 或者，右键单击状态名称并选择“删除”选项。

## 配置时间线 - 状态

您可以使用“配置时间线”对话框的“状态”选项卡来管理状态。要显示此对话框，请执行以下任一操作：

- 双击生命线元素
- 右键单击生命线元素并选择“属性”选项，或
- 在 Value 生命线上， 生命线

“配置时间线”对话框默认为“状态”选项卡。

当前为生命线元素定义的所有状态都列在“状态”面板中。

### 添加新的状态

节	行动
1	在“状态名称”字段中，输入生命线元素中第一个新状态的名称；例如，“等待状态”。
2	单击保存按钮。 状态被添加到“状态”面板和（对于状态生命线元素）到图表中。
3	单击新建按钮。
4	在“状态名称”字段中，输入生命线元素中下一个状态的名称。
5	重复步骤 2 到 5，直到您添加了所有需要的状态（您必须在生命线元素中添加至少三个）。
6	添加完所有需要的状态后，单击确定按钮关闭“配置时间线”对话框。

### 编辑现有状态

节	行动
1	单击“状态”列表中的状态。
2	在“状态名称”字段中，更改状态状态。
3	单击保存按钮。

### 删除现有状态

节	行动
1	单击“状态”列表中的状态。

2	点击删除按钮。
---	---------

## 更改状态的顺序

节	行动
1	单击 状态：“列表中的状态”。
2	点击  或  按钮，将状态向上或向下序列。

## 数值范围生成器

您还可以使用 **配置时间线**对话框来创建具有要应用于时间线的数值的状态范围。

**重要提示：**此操作会删除时间线元素的所有现有状态和转换。

### 创建具有数值的状态范围

节	行动
1	双击生命线元素。 将显示 <b>配置时间线</b> 对话框。
2	单击创建连续数值状态按钮。 将显示 <b>数字范围生成器</b> 对话框。
3	在 <b>高值</b> 和 <b>低值</b> 字段中，输入范围的上限值和下限值。
4	在 <b>节值</b> 字段中，输入增加间隔。 无意义的值不解析； <b>低值</b> 必须 于 <b>高值</b> ， <b>节值</b> 必须是 于总范围的正值。
5	在 <b>单位</b> 字段中，输入测量单位的名称；例如， <b>分钟</b> 。
6	点击确定按钮。 Enterprise Architect显示一个警告，指出现有状态和转换将被删除。
7	单击是按钮。 重新显示 <b>配置时间线</b> 对话框， <b>状态</b> 面板中列出了定义的状态范围。
8	点击确定按钮。 为一个： <ul style="list-style-type: none"> <li>• Value生命线，第一个状态显示在Timeline的整个时间范围内</li> <li>• 状态生命线，状态范围显示为Timeline的y轴</li> </ul>

## 配置时间线 - 过渡

您还可以使用 配置时间线”对话框的 转换”选项卡来管理转换。要显示此内容，请执行以下任一操作：

- 双击生命线元素
- 右键单击生命线元素并选择 属性”选项，或
- 在 Value 生命线上， 生命线

配置时间线”对话框默认为 状态”选项卡。单击 转换”选项卡。

为时间线元素定义的所有转换都列在 转移

”中转移

点面板。

### 添加新的过渡

节	行动
1	单击新建按钮。
2	在 新的转移 '面板，键入过渡的详细信息。
3	单击保存按钮。

### 编辑过渡

节	行动
1	单击列表中的转换。
2	在 编辑转移 ”中转移 '面板，根据需要编辑转换的字段。
3	单击保存按钮。

### 删除过渡

节	行动
1	单击列表中的转换。

---

2	点击删除按钮。 转换从对话框和生命线中移除。
3	点击确定按钮。

## 时间间隔

您可以使用间隔栏（沿每个选定生命线元素顶部的浅色线）创建和管理时间间隔。使用时间间隔，您可以对转换执行各种操作，例如复制和粘贴。您还可以压缩时间线的各个部分，使它们不可见。

每个时间间隔显示在所有时间线元素。

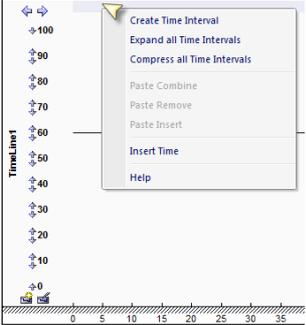
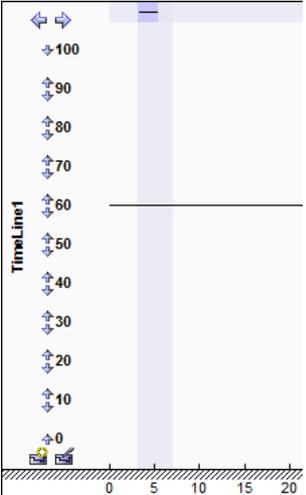
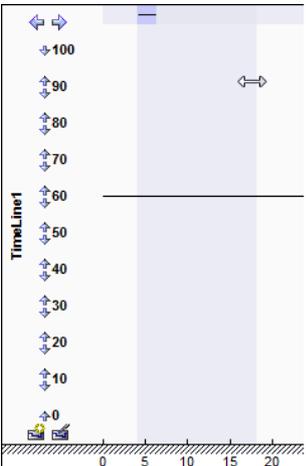
### 管理时间间隔

行动	描述
创建时间间隔	您可以使用以下方法创建时间间隔： <ul style="list-style-type: none"> <li>• 间隔酒吧 - 上下文菜单</li> <li>• 间隔条 - Shift 键，或</li> <li>• 时间线-上下文菜单</li> </ul>
压缩时间间隔	您可以压缩时间间隔以节省较长时间线的空间。
选择时间间隔	有多种方法可以选择时间间隔来执行其他操作。
移动时间间隔	要移动时间间隔，请将光标移到时间间隔内的间隔条上，按住鼠标按钮并向左或向右拖动间隔。 时间间隔可以相遇，但不能重叠。
调整时间间隔	要调整时间间隔的大小，请将光标移动到时间间隔开始或结束边缘的间隔条上，按住鼠标按钮并向左或向右移动边缘。 时间间隔可以相遇，但不能重叠。
删除时间间隔	要删除时间间隔，请选择要删除的每个时间，然后按删除键。 删除时间间隔不会删除该间隔内的转换。

# 创建时间间隔

您可以通过多种方式在计时元素上创建时间间隔。

## 使用间隔栏上下文菜单创建时间间隔

图片	节与行动
	<p>1. 右键单击间隔栏大约开始或结束时间间隔的点，然后选择“创建时间间隔”选项。</p>
	<p>2. TimeInterval 将所有时间线元素向下显示，作为顶部带有蓝色压缩框的窄淡色带。</p>
	<p>3. 将光标移动到间隔条中时间间隔的边缘，使光标变为拖动形式，将边缘拖动到正确的起点或终点。</p>

## 使用间隔条和 **Shift** 键创建时间间隔

节	描述
1	将光标移到间隔条上并按 <b>Shift</b> 。 光标改变形状 (  )。
2	单击以创建时间间隔。
3	将光标移动到间隔栏中的时间间隔的边缘，使光标变为拖动形式 (  ) 并将边缘拖动到正确的起点或终点。

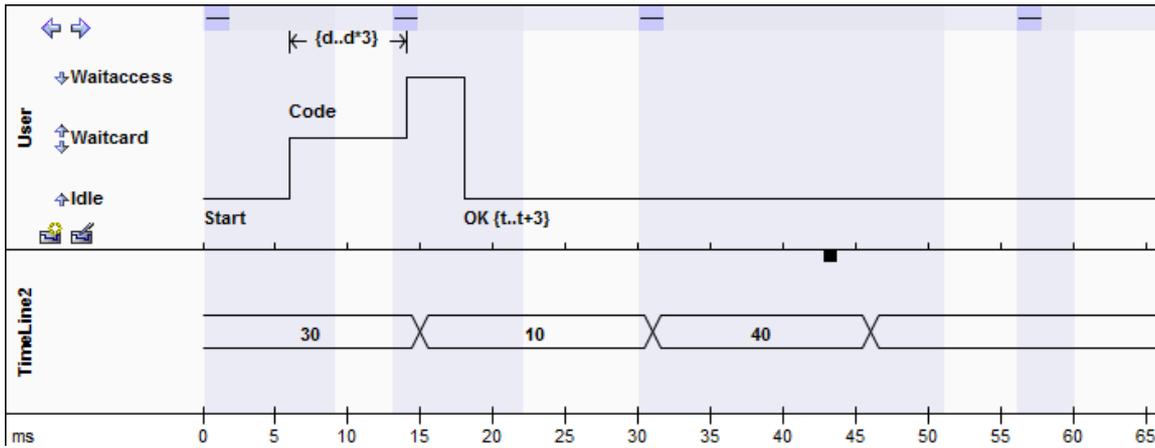
## 使用时间线上下文菜单创建时间间隔

节	描述
1	在过渡后右键单击时间轴。 上下文菜单显示。
2	单击 <b>选择</b> 选项。 Enterprise Architect 创建一个时间间隔，定义从所选转换到下一个转换的时间段。 如果在此期间还有其他时间间隔，Enterprise Architect 将其替换为单个时间间隔，用于过渡状态；您应该在创建时间间隔时考虑这一点，因为它延伸到图表中的其他时间线元素。 此方法的 A 价值是它为不发生转换的时间段创建一个时间间隔，这可能很长；然后你可以压缩这个时间间隔来隐藏不活动的时间。

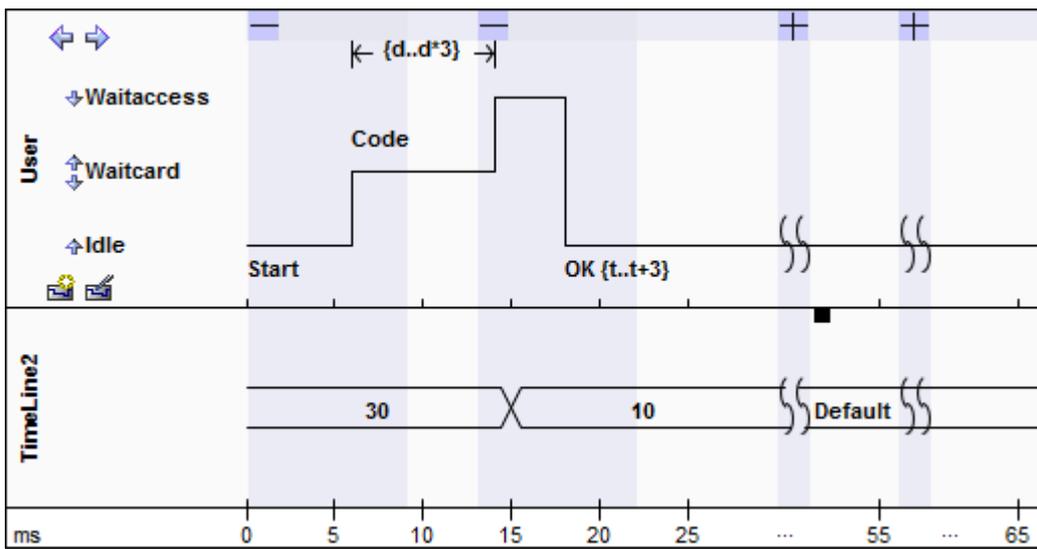
# 压缩时间间隔

您可以压缩时间间隔以节省较长时间线的空间。

## 未压缩时间间隔



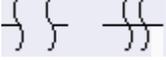
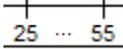
## 压缩时间间隔



注记：

您还可以使用上下文菜单选项压缩和扩展时间间隔。

物品	描述
	压缩切换框： <ul style="list-style-type: none"> <li>展开，点击这个压缩选择的时间间隔</li> <li>被压缩，点击这里再次展开选中的时间间隔</li> </ul>
	时间线本身的压缩部分，在所有元素中。

	<p>如果成对符号之间有空格，则压缩部分中有过渡。                  如果时间线继续通过配对符号，则压缩部分中没有过渡。</p>
	<p>元素下方时间范围内的压缩部分。</p>

# 选择时间间隔

## 选择间隔

任务	行动
选择图表上所有元素的时间间隔	单击时间间隔内的间隔条。
选择多个单独的时间间隔	在每个时间间隔内单击间隔条的同时按住 <b>Ctrl</b> 键。
选择范围内的所有时间间隔	单击范围内第一个时间间隔内的间隔条，然后按住 <b>Shift</b> 键并单击范围内最后一个时间间隔内的间隔条。 选择两者之间的所有时间间隔。

## 修改间隔

选择时间间隔后，您可以对其进行修改。

任务	行动
从选择中排除生命线元素	按住 <b>Ctrl</b> 键并单击该元素中选择的任何部分。 重复该步骤以切换选择并重新包含元素。
只选择一个生命线元素，排除所有其他生命线元素	按住 <b>Shift</b> 键并单击该元素中选择的任何部分。

# 时间间隔操作

您可以选择和更新特定的时间间隔。

右键单击间隔内的间隔条。上下文菜单显示A这些选项。

## 压缩时间线

压缩“切换框和 压缩间隔”菜单选项在时间间隔上运行，并压缩时间线和间隔内的所有过渡。您有一个替代选项，它在时间线上运行并压缩单个过渡状态。

1. 在转换后右键单击时间线（而不是间隔栏），然后选择 压缩”选项。
2. Enterprise Architect创建一个新的时间间隔，跨越从选定过渡到下一个过渡的时间段，然后压缩该时间间隔。

如果在此期间还有其他时间间隔，Enterprise Architect将其替换为单个时间间隔，用于过渡状态。您应该在创建和压缩时间间隔时考虑这一点，因为它延伸到图表中的其他时间线元素。

这个方法的价值是它为没有发生转换的时间段创建一个时间间隔，这个时间段可能很长，然后压缩这个时间间隔来隐藏不活动的时间段。

## 上下文菜单选项

选项	行动
选择间隔 取消选择间隔	选择时间间隔，或者，如果已选择间隔，则取消选择。 您可以通过这种方式选择多个时间间隔，在每个时间间隔单独访问菜单。
切换间隔选择	在选定的时间线元素内切换时间间隔的选择或取消选择。 您在所有时间轴元素中选择或取消选择时间间隔，但 切换”选项仅对您访问菜单的元素起作用。
压缩间隔	压缩时间间隔，并隐藏该时间间隔内的所有过渡。 这对于隐藏时间线上的长时间不活动也很有用。
删除间隔	删除时间间隔。
复制	复制所有选定时间间隔的过渡。
切	从图中复制并删除选定的转换。
剪切和移除时间	从图中复制并删除位于所选时间间隔内的转换。 此选项还会从时间线中删除时间，该量是时间间隔的持续时间。 所选时间间隔右侧的所有转换和时间间隔都向左移动。
删除	从图中删除选定的转换。
删除和删除时间	删除图中选定时间间隔内的转换。 此选项还会从时间线中删除时间，该量是时间间隔的持续时间。 当前时间间隔右侧的所有转换和时间间隔都向左移动。

插入时间	将时间添加到时间线并将所有过渡和时间间隔向右移动。 同时扩展当前时间间隔的持续时间。
------	---

## 图表中的所有时间图表

要创建新的时间间隔或跨图表中的所有时间间隔工作，请右键单击时间间隔之间的间隔栏。A 上下文菜单，提供许多选项（“粘贴...”菜单选项在复制过渡 变为活动状态）。

菜单选项	行动
创建时间间隔	创建一个时间间隔。
展开所有时间间隔	在整个图表上展开所有时间间隔。
压缩所有时间间隔	压缩整个图表上的所有时间间隔。
粘贴组合	粘贴复制的过渡在复制的时间范围内的任何现有过渡。 该图不允许两个连续转换到相同的状态，并自动删除第二个转换。
粘贴移除	删除所有过渡，然后在复制的时间范围内粘贴复制的过渡。
粘贴插入	插入时间，将所有过渡和时间间隔向右移动，以便为粘贴复制的过渡腾出空间。
插入时间	将时间添加到时间线并将所有过渡和时间间隔向右移动。 此选项不会更改任何时间间隔的持续时间。

## 将过渡从一个时间线元素复制并粘贴到另一个时间线

节	行动
1	按住 Shift 键并在时间间隔内选择时间线元素以进行复制或剪切。
2	右键单击间隔栏（选择哪个元素都没有关系）。 上下文菜单显示。
3	复制或剪切过渡。 您还可以削减和删除时间。
4	选择要粘贴过渡的时间线，然后右键单击间隔栏。 上下文菜单显示。
5	选择一种粘贴操作。 注记如果时间轴中不存在状态，则创建状态。

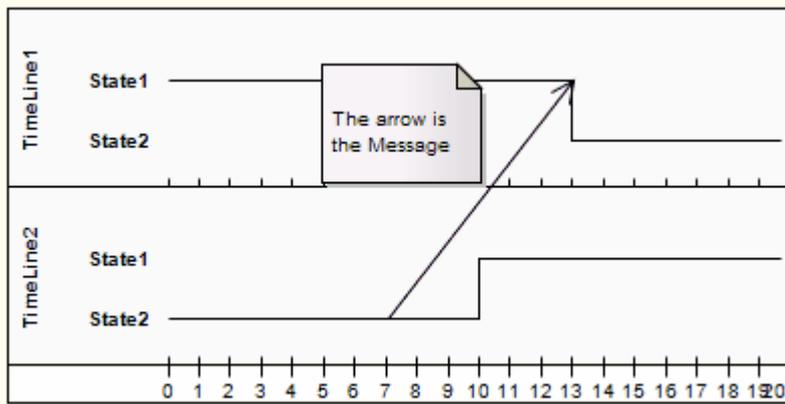
将创建您要粘贴到的时间线元素中不存在的任何状态。 创建的任何新状态都可能顺序错误；您可以通过图表中的“快速”按钮更改顺序。
--

## 在选定的时间间隔或多个选定的时间间隔内转换过渡

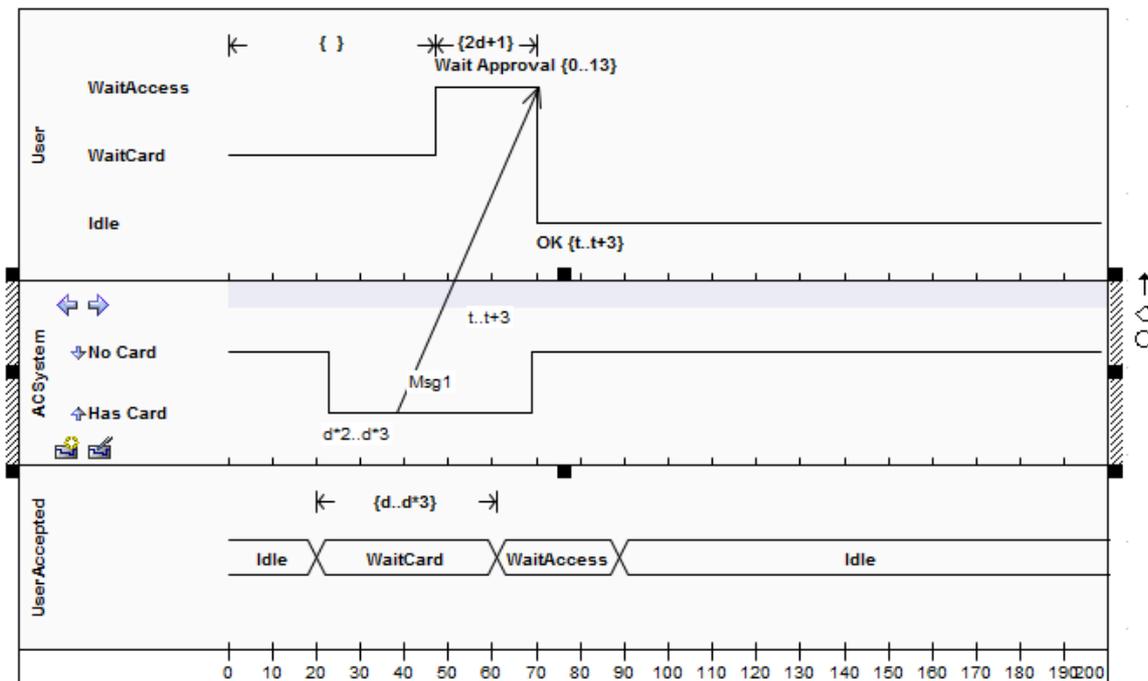
节	行动
1	选择所有包含要移动的过渡的时间间隔。
2	按住 <b>Shift</b> 并单击间隔栏（无论您选择哪个时间线元素），然后向左或向右移动过渡。 您不能将转场拖到其他转场上；当移动的过渡与静止的过渡碰撞时，移动停止。 如果您有碰撞问题，请使用 <b>Shift+select</b> 来移动单个时间轴元素的过渡。

# 消息 ( 图表 )

消息是时序图中生命线之间的通信链接。在时间线的情况下，信息是两个时间线对象之间的连接。

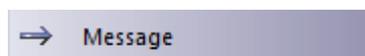


例如：



OMG Unified Modeling Language规范 · ( 1 · 图 14.30 和 14.31 · p.520 。 )

## 工具箱icon



# 创建时间信息

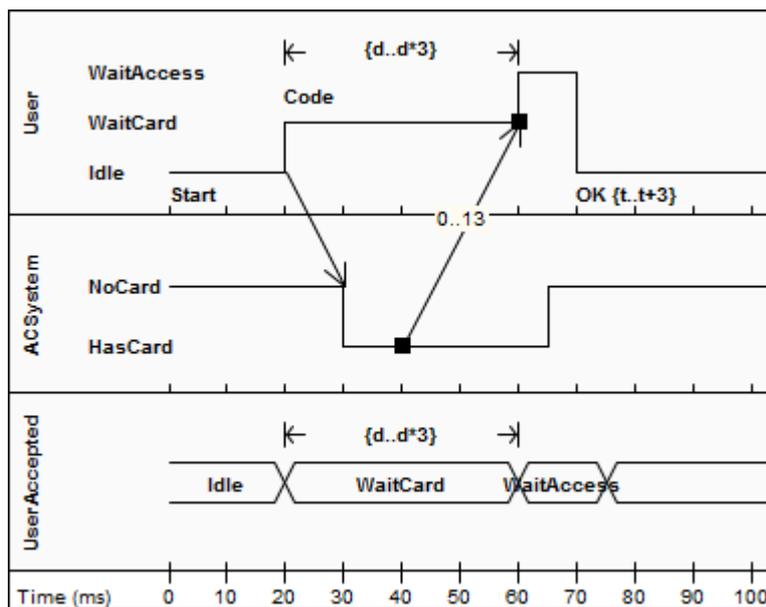
您可以在时序图上的两个生命线对象（状态或值）之间创建时序信息，每个对象都有现有的转换点。

## 在生命线之间创建信息

节	行动
1	点击  工具箱关系信息图表工具箱“对话框并指定“时间”）。
2	在信息开始的地方点击源生命线，将光标拖到目的地生命线上信息结束的过渡点。在这两个点之间创建一个新的计时信息。
3	双击新信息以打开“时间信息”对话框。审阅或完成对话框，如“对话框字段”表中所示。

## 对话框字段

此图显示了配置信息的示例：



OMG Unified Modeling Language规范 · ( 1 · 图 14.30 和 14.31 · p.520 )

字段/按钮	行动
开始	标识信息来源的生命线。
结尾	标识信息终止的生命线。
开始时间	显示时间线开始后信息开始的时间。如果需要，您可以更改此设置。

结束时间	显示时间线开始后信息结束的时间。如果需要，您可以更改此设置，但时间必须对应于目标生命线上的过渡点。
名称	(可选) 信息名称中的类型。
时间观察	(可选) 类型作为标签的任何文本，提供有关何时发送信息。
持续时间观察	(可选) 类型作为标签的任何文本，提供有关特定状态下生命线间隔的信息，从收到信息开始。
转移至	信息终止的目标生命线上的状态。如有必要，您可以单击下拉箭头并选择要转换到的其他状态。信息的头部相应地移动。
事件	(可选) 触发转换的任何事件的名称中的类型。
时间约束	(可选) 传输信息所需的最长时间类型。
期限约束	(可选) 类型生命线在收到信息后能保持变化状态的最长时间。

## 注记

- 您可以沿源时间线自由移动信息的源端；但是，目标端（箭头）必须附加到过渡
- 如果您创建一个新信息并且没有给它一个目标转换，它会自动找到并附加到最近的转换；如果您移动目标端，它会拖动过渡

# 序列图表

序列图是行为A结构化表示，随着时间的推移一系列顺序步骤。您可以使用它来：

- 描述工作流程、信息传递以及一般元素如何随着时间的推移进行合作以实现结果
- 在分析早期捕获整个系统的信息流和责任流；元素之间的消息最终成为类模型中的方法调用
- 为用例场景制作解释模型；通过使用参与者和参与的用例创建一个序列图，您可以模型用户和系统为完成所需任务而序列的步骤

## 建造

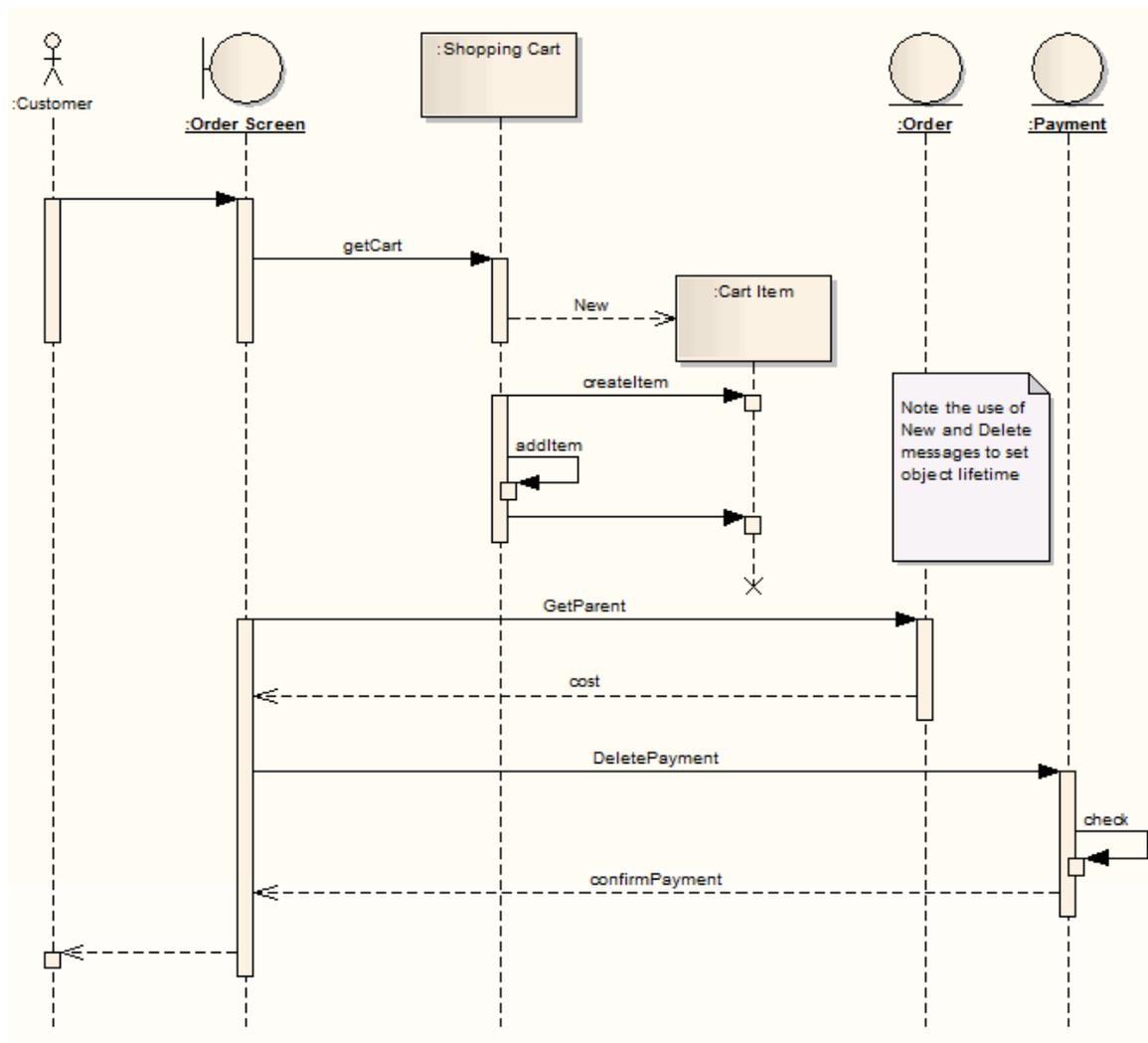
- 序列元素按水平顺序序列，消息在元素之间来回传递
- 序列图上的消息可以有多种类型；消息还可以配置为反映源元素和目标元素的操作和属性（参见信息帮助主题中的注记）
- 一个参与者元素可以用来表示用户发起的事件
- 边界、控件和实体等元素可以分别用于说明屏幕、控制器和数据库项
- 每个元素都有一个名为生命线的虚线，该元素存在并可能参与交互

要切换序列图中的消息编号，请选中或取消选中“首选项”对话框中的“显示序列编号”复选框。

您从工具箱“交互”的元素和序列的页面中生成连接器。

## 示例图表

此示例序列图演示了几个不同的元素。



### 图形图表元素工具箱序列

图标	描述
Actor	参与者是系统的用户；用户可以指人类用户、机器，甚至是模型中的另一个系统或子系统。
Lifeline	A生命线代表了一个独特的可连接元素，是交互中的个体参与者。
Boundary	在分析中使用边界来捕获用户交互、屏幕流和元素交互。
Control	A控件组织和安排其他活动和元素。
Entity	实体是一种原型化的物件，它对捕获系统中的信息或知识的存储或持久性机制进行建模。
Fragment	Fragment元素可以表示序列图中A迭代或替代过程。

 Endpoint	Endpoint信息用于在交互中反映丢失或找到的序列。
 Diagram Gate	图表门是A简单的图形方式，用于指示消息可以传入和传出交互片段的点。
 State/Continuation	状态/延续对于序列图元素两个不同的用途，作为状态不变量和继续。
 Interaction	您可以使用一个交互来插入一个元素交互作为一个类元素。

### 图表连接器工具箱序列

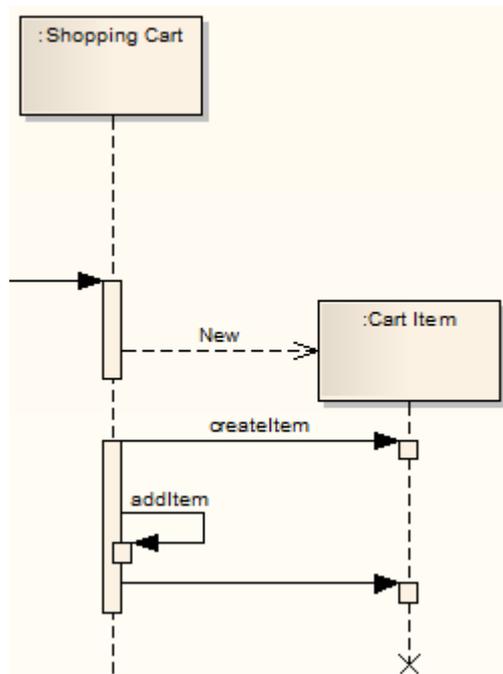
图标	描述
 Message	A信息表示元素之间的信息流或控制转移。
 Self-Message	Self-Message 反映在调用生命线的操作中调用A新流程或方法。
 Recursion	递归是序列图中用于指示递归函数的A信息。
 Call	A调用是一种信息连接器，它扩展了先前信息的激活级别。

# 表示元素的生命周期

使用标记为新消息或删除消息类型的消息捕获元素生命周期

节	行动
1	双击序列图中的消息以显示该信息的属性窗口。
2	在“生命周期”字段中，单击下拉箭头并选择“新建”或“删除”。
3	单击确定按钮保存更改。

## 示例图表



此示例显示了具有特定创建和删除时间的两个元素。

## 注记

- 要在示例图中显示生命线上的终止 X，您必须打开垃圾收集：'开始>外观>首选项>首选项>图表>序列：垃圾收集'

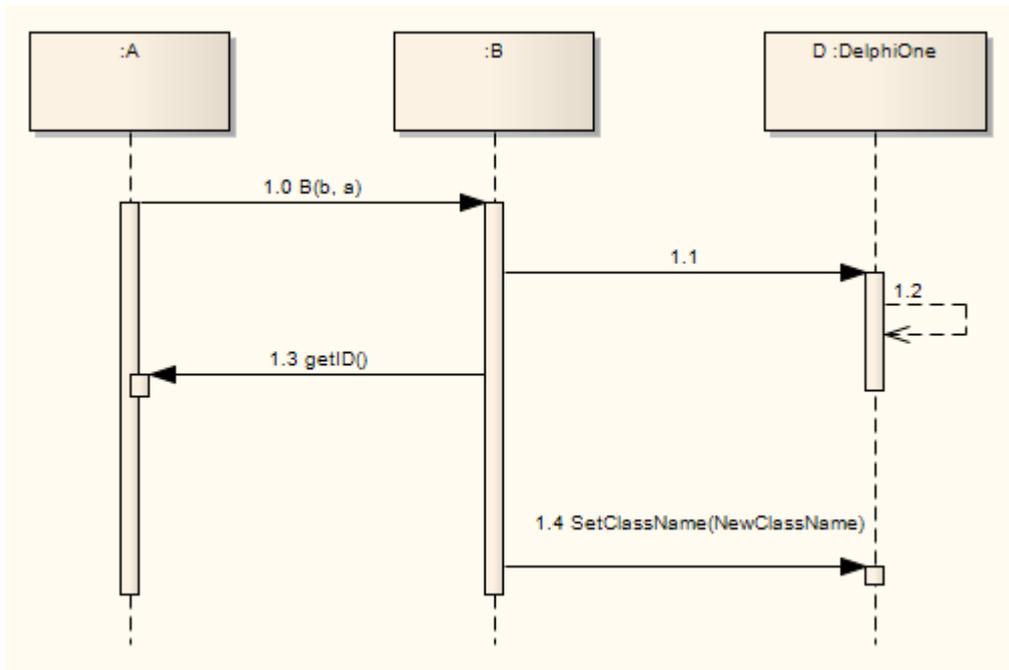
# 布局序列图表

## 偏移序列消息的垂直分隔

节	行动
1	在序列图中选择适当的消息。
2	<p>使用鼠标根据需要向上或向下拖动消息。</p> <p>当您向上或向下拖动一条消息时，该消息下方的任何消息或片段都会向上或向下移动相同的量。</p> <p>如果启用了“重新排序消息”选项，当您消息向上或向下拖过下一条或上一条消息时，消息会交换位置，而不是简单地移动位置。或者，在移动消息时按住 Shift 键，以获得相同的结果。在窗口下（但在 Linux 或虚拟机下），您也可以以相同的方式使用 Alt 键。</p> <p>当您将一个信息移过另一个时，会显示一个工具提示，提醒您启用重新排序来自布局的消息重新排序消息的帮助程序，无论是否启用该选项。您可以通过取消选择“首选项”对话框的“图表&gt;序列”页面上的“重新排消息时启用工具提示”复选框来隐藏此工具提示。</p>

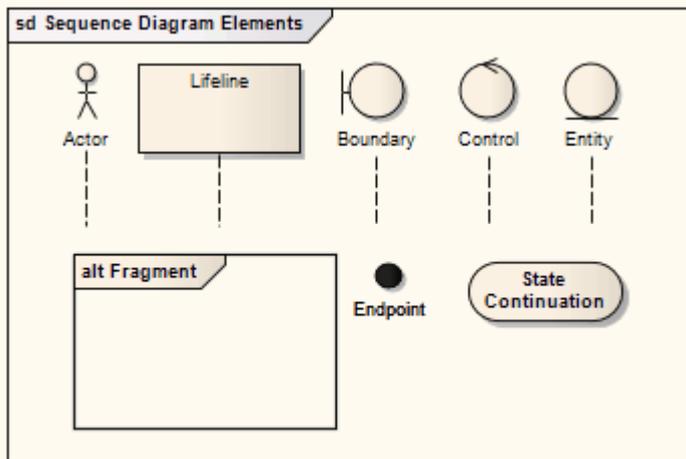
## 示例图表

此示例显示了序列图中空间的经济使用。



# 序列元素

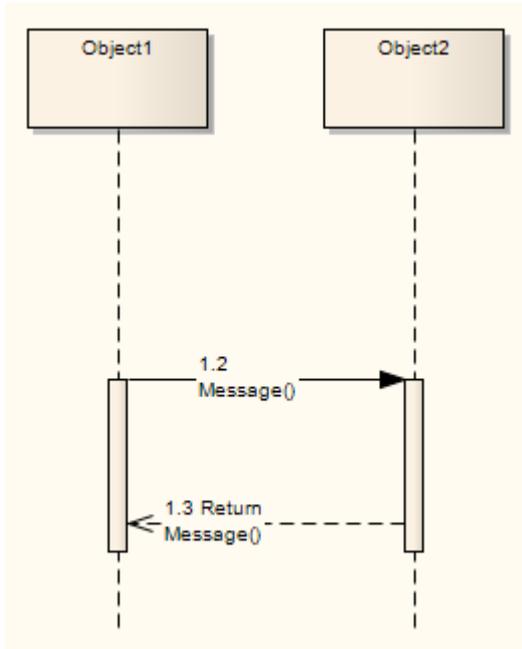
这个例子展示了序列图的一些可能元素及其原型显示。



## 元素描述

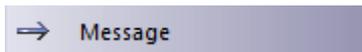
元素	描述
参与者	运行时actor的一个实例；这可以被描述为人形或矩形符号。
生命线	具有刻板印象生命线的物件元素。
边界	代表用户界面屏幕或输入/输出设备。
实体	A持久的元素——通常实现为一个数据库表或元素。
控件	控制完成什么工作、何时以及如何完成的活动组件。

# 消息 ( 序列图表 )



序列图使用从元素传递到元素的消息来描述随时间推移的工作流或活动。在软件模型中。这些消息对应于类操作和行为。当您显示序列时，图表工具箱自动切换到工具箱，其中包含 交互”信息的图表图标。

## 工具箱icon



## 访问

图表工具箱	单击 信息”图标，单击源object并将光标拖动到目标object ( 如果信息的属性窗口未显示，请右键单击信息和 属性”菜单选项 )
-------	--

## 在序列图上创建信息

选项	行动
信息	类型信息名称。 如果信息流是流向类元素 ( 从类图中下拉 ) 或具有分类器的生命线元素，并且目标类已经定义了操作，您可以单击下拉箭头并选择适当的操作名称；信息则反映目的地类操作。 您还可以在列表中包含元素的分类器继承的操作。为此，请选中 显示继承方法”复选框。

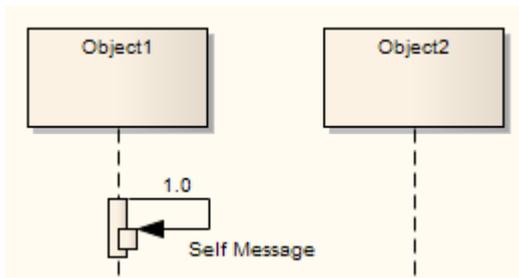
操作	<p>如果目标类上的可用操作不合适，请单击此按钮并使用“操作”对话框在目标元素中定义新操作。</p> <p>如果创建信息时不参考目标类操作，则不会向目标类添加新操作。</p>
参数	类型信息具有的任何参数，以逗号分隔的列表形式。
参数值	(可选) 类型每个参数对应的实际值，以逗号分隔的列表形式。
返回价值	如果信息有返回值或类型，请在此字段中指定。
显示继承的方法	<p>选中此复选框以将目标元素的分类器继承的操作包含在“信息”字段中可用操作的下拉列表中。</p> <p>清除复选框以仅显示来自分类器本身的操作。</p>
分配给	<p>如果信息流来自具有定义属性的类元素或生命线元素，请单击分类器箭头并选择适当的属性名称。</p> <p>信息反映了源类的属性；您不能在此处向源类添加更多属性 - 如果未列出适当的属性，请打开类元素的“属性”对话框并添加所需的属性。</p> <p>否则，可选择键入要分配消息流的object的名称。</p>
构造型	(可选) 类型或选择连接器的构造型 (如果输入，则会显示在图表上)。
别名	<p>(可选) 类型信息名称的别名。</p> <p>在图表上，如果在图表的“属性”对话框的“图表”选项卡上选择了“使用别名如果可用”复选框，则显示别名而不是信息名称。</p>
条件	类型为了发送信息而必须为真的任何条件。
约束	类型发送信息时可能存在的任何约束。
是迭代	<p>选中复选框以指示信息将迭代，直到指定的条件取值为false。图表上的条件语句以星号(*)为前缀。</p> <p>如果指定的条件为真，清除复选框以指示信息将仅在处理周期内发送一次。</p>
开始组	(用于通讯图消息)。选中此复选框可将信息(和所有后续消息)重置为具有新初始编号的单独组。
同步	<p>单击下拉箭头并根据需要选择“同步”或“异步”。</p> <p>值“同步”禁用“种类”字段；同步消息始终是调用。</p>
种类	<p>当“同步”字段设置为异步时，该字段被启用。</p> <p>单击下拉箭头并根据需要选择“调用”或“信号”。</p>
生命周期	<p>选择“新建”以在信息末尾创建新元素，或选择“删除”以在信息末尾终止消息流。</p> <p>如果两种情况都不适用，则将该字段设置为“&lt;none&gt;”。</p>
是返回	如果您创建的信息是回信，请选中此复选框。
节省	<p>单击此工具栏按钮以保存信息定义或对其进行的任何更改。</p> <ul style="list-style-type: none"> <li>您可以在“时序详细信息”对话框中更改消息的时序详细信息，并使用通用</li> </ul>

	<p>排序强调序列排序的消息</p> <ul style="list-style-type: none"><li>• 要切换序列图上的消息编号，请选中或取消选中“首选项”对话框中的“显示序列编号”复选框</li></ul>
--	---

## 注记

- 您还可以将信息连接器用作信息流，在信息上实现信息流

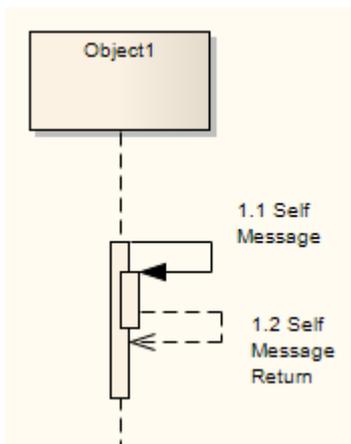
## 自我留言



Self-Message 反映在调用生命线的操作中调用A新流程或方法。它是信息的规范，通常在序列图中。自消息调用表示嵌套调用；每次调用添加新的激活级别。

## 自我留言作为返回

可以描述自信息调用的返回。



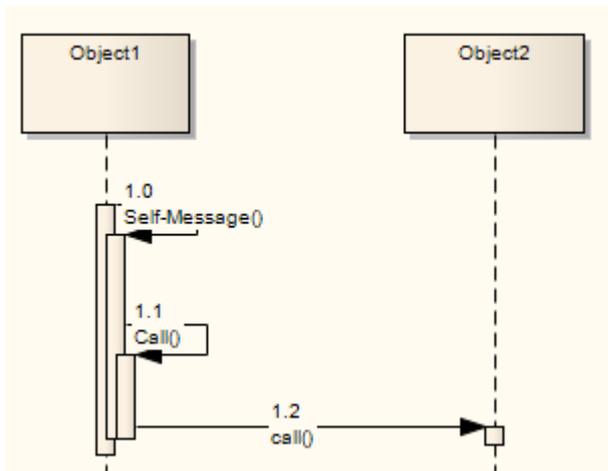
## 创建一个自我信息回报

节	行动
1	在执行结束时创建第二个自我信息。
2	双击信息名称以打开信息的属性窗口。
3	选中“返回”复选框。
4	提高回报的激活级别。

## 工具箱icon

 Self-Message

# 调用



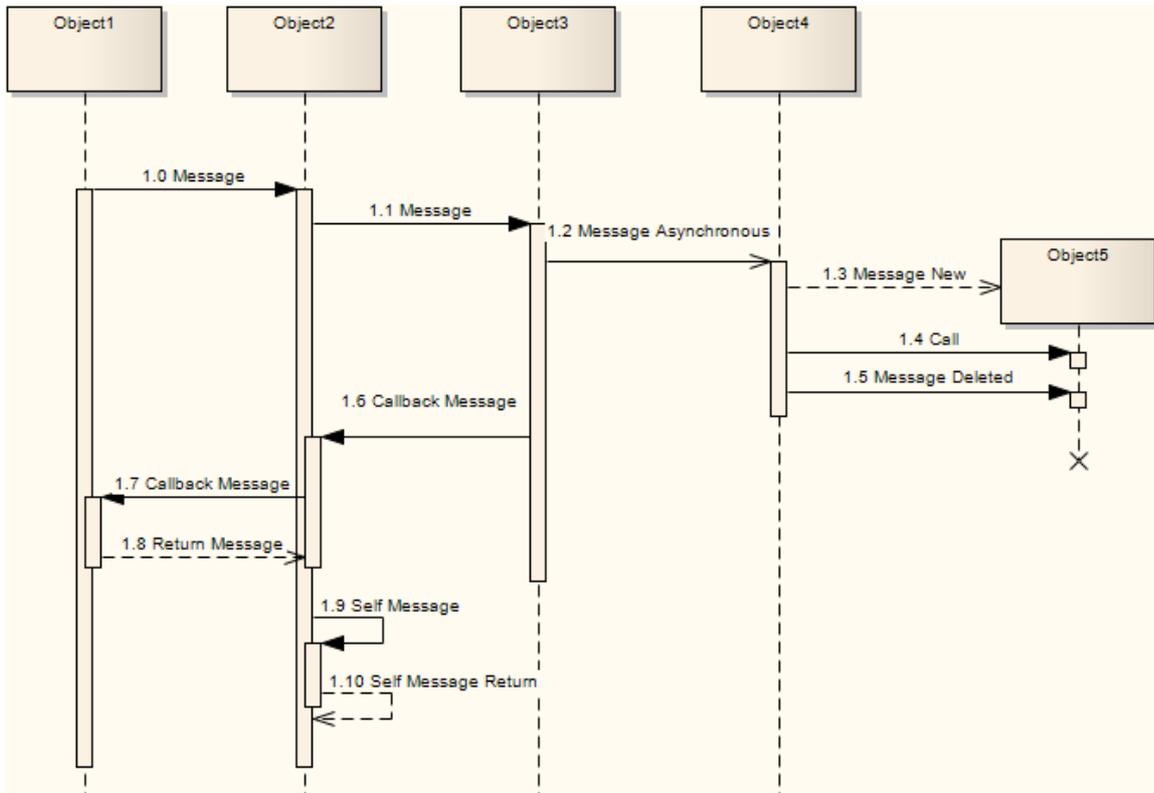
A调用是一种信息连接器，它扩展了先前信息的激活级别。所有的自我信息都会创建一个新的激活级别，但这种聚焦控制通常以下一个信息结束（除非手动调整激活级别）。自消息调用，如第一个调用的图像调用，表示嵌套调用；每次调用添加新的激活级别。与元素之间的常规信息不同，元素之间的调用继续源元素中的现有激活，这意味着调用是在先前信息的激活范围内启动的。

## 工具箱icon



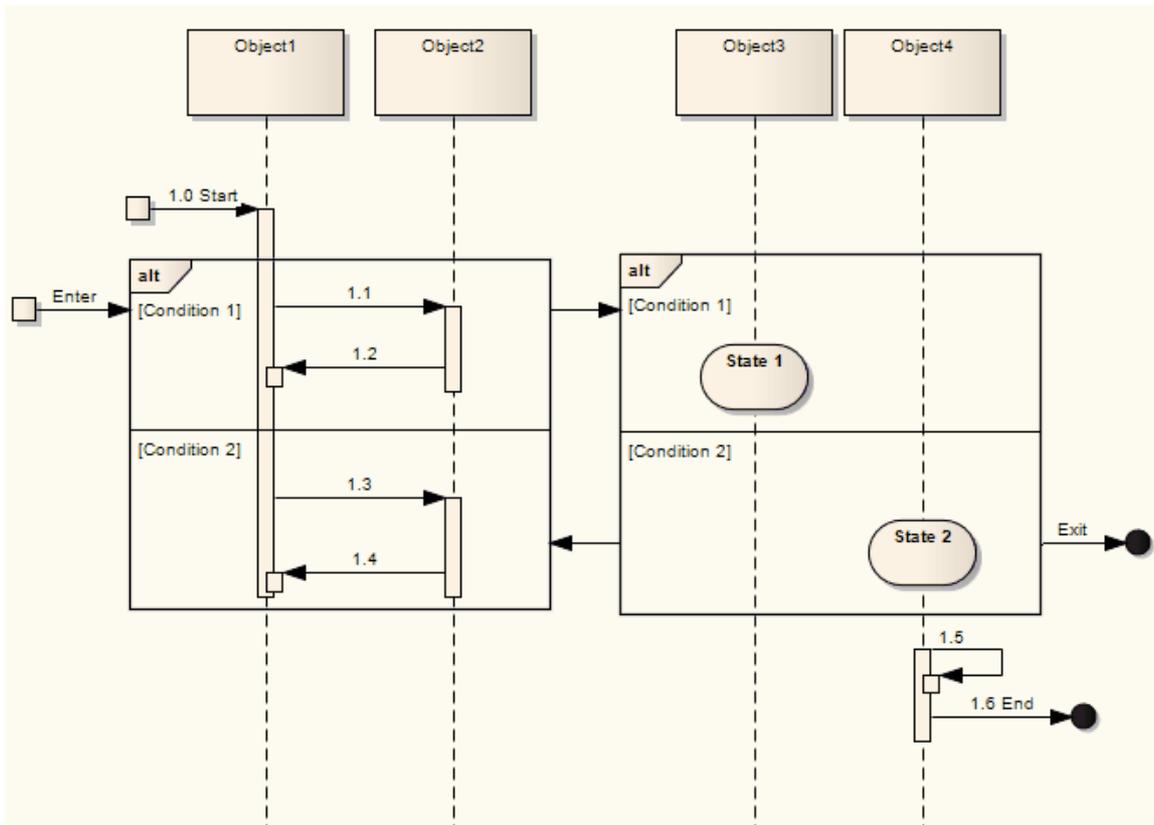
# 信息示例

这些是序列图中可用的不同类型的信息。请记住，序列图中的消息也可以使用形状脚本进行修改。



## 其它序列消息

这些是不属于图表描述的序列的一部分的消息示例。



# 更改时序细节

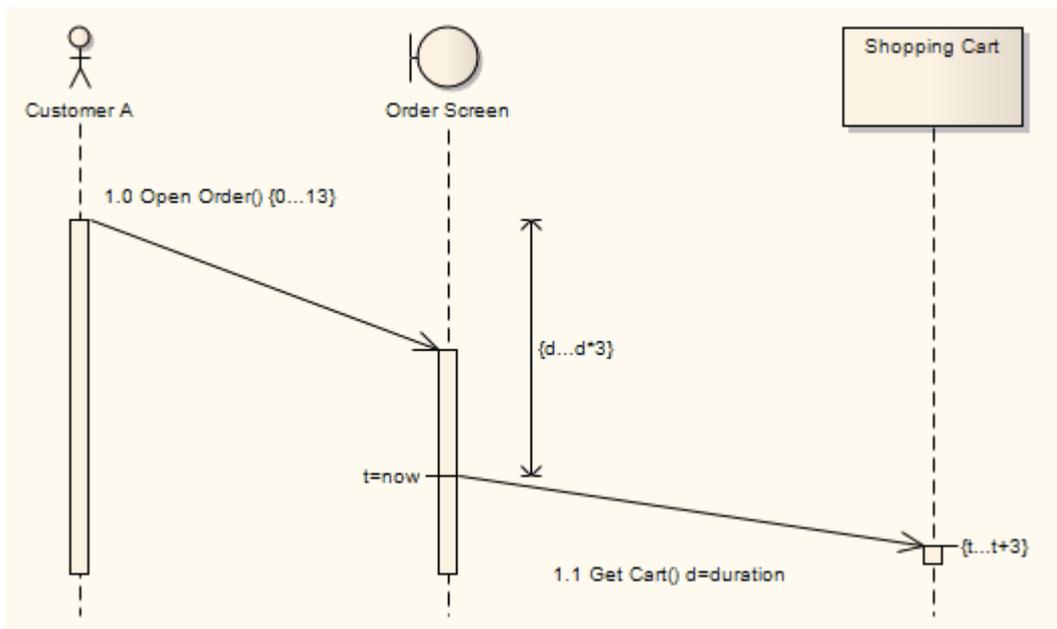
可以更改序列图中信息的时序细节。

## 访问

上下文菜单	右击信息 计时细节
-------	-----------

## 更改时间

请参阅 *OMG Unified Modeling Language规范 (v2.5.1, p. 511)*。



在此图中，在未结订单信息上：

- '持续时间约束' 已设置为 0...13

在获取购物车信息上：

- 消息之间的持续时间约束"已设置为 d...d\*3
- 持续时间观察"已设置为 d=duration
- 时间约束"已设置为 t...t+3
- 时间观察"已设置为 t=now

通过在“持续时间约束”字段中输入值，您可以调整信息角度。单击“时间详细信息”对话框上的确定按钮后，单击信息连接器的头部并向上或向下拖动连接器以更改角度。您不能将角度延伸到连接序列object的生命线之外或创建小于5度的角度。

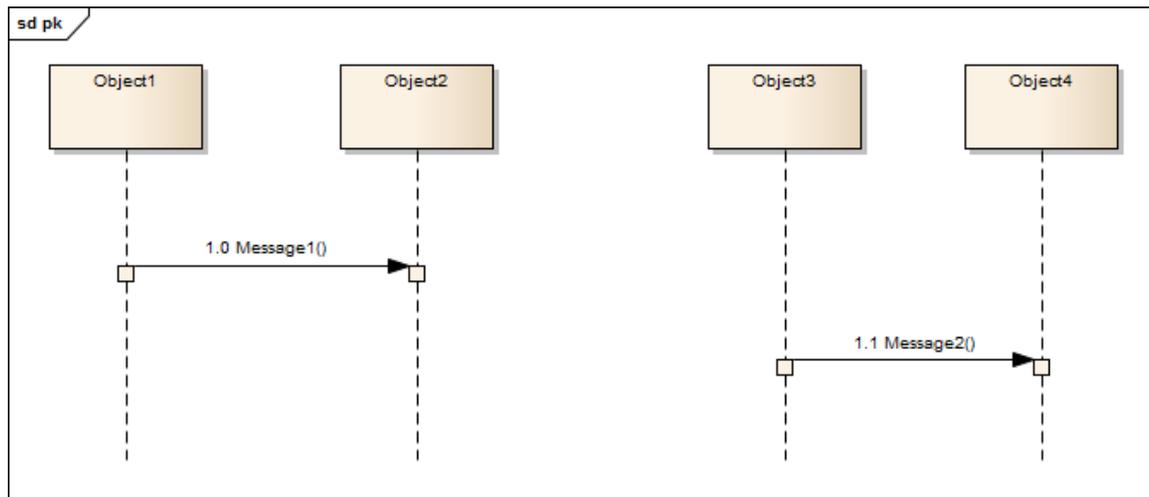
您还可以通过将“General Ordering”箭头向上拖动到上一条消息加入当前消息的源生命线的点来创建“DurationContemporaryBetween约束”行。将显示A对话框，您可以在其中输入约束值。创建线后，您可以将其移动到当前消息中途和上一条消息中途的任何点，以避免与其他消息计时细节重叠。您可以通过“时间详细信息”对话框或右键单击该行本身并选择适当的上下文菜单选项来编辑或删除该值。

字段	行动
----	----

期限约束	指明消息可以持续多长时间的最小和最大限制。
消息之间约束	表示在当前消息的源生命线发送或接收上一条消息与发送当前消息之间的最小和最大间隔。
持续时间观察	捕获消息的持续时间。
时间约束	指示消息应到达目标的最短和最长时间。
时序观察	捕获发送消息的点。

# 一般订购

在序列图中， workflow 由图表下方的消息序列表示。图表顶部附近的消息在图表下方的消息之前传递。

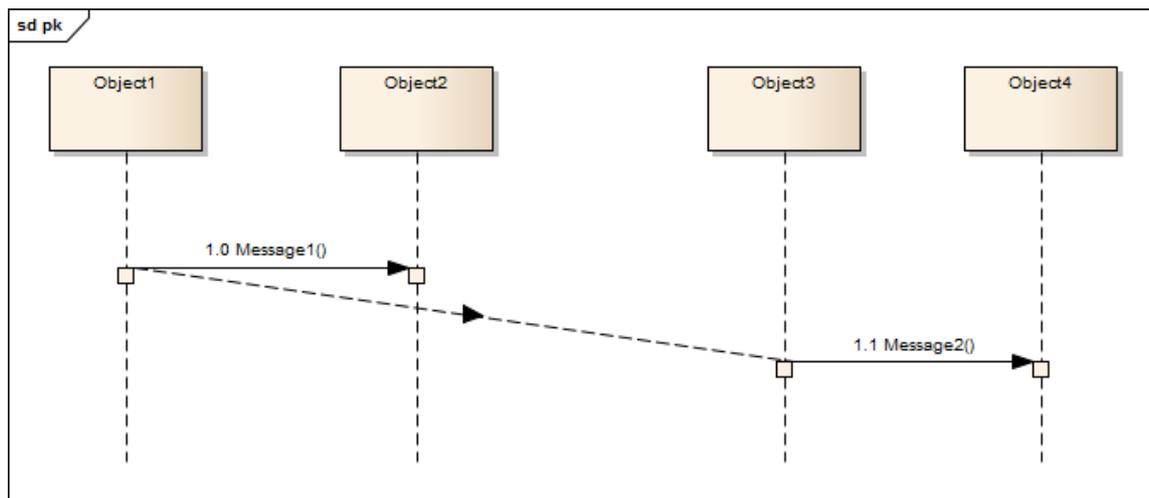


图中，信息1早于信息2。但是，在复杂的图表中，或者在表示精细定时操作或并行处理时，这可能并不明显。您可以使用“General序列”箭头加强顺序。

点击信息箭头。源锚点处会显示A小箭头。



单击此箭头并将其拖动到序列中下一个信息的开头（示例中为信息2）。显示一般排序箭头，指示第二个信息在第一个信息之后。



如有必要，您可以拥有多个从信息发出或信息的通用订购箭头。

## 异步信号信息

通过显示信息的属性窗口并将“同步”字段设置为“异步”，将“种类”字段设置为“信号”，您可以将信息定义为异步信号消息。同步消息不能传送信号，因此将“同步”字段设置为“同步”会禁用“Kind”字段。

“返回值”、“分配给”和操作，不适用于异步信号，被禁用。

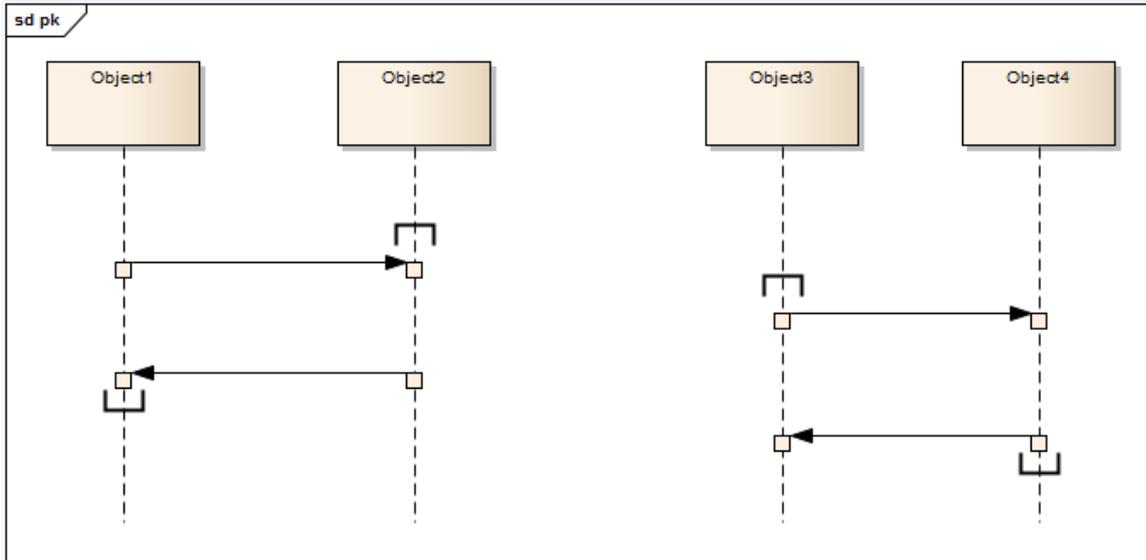
操作按钮变为信号按钮，单击该按钮可将异步信号消息与模型中的信号元素相关联。您可以将与信号属性对应的参数输入到“参数值 (s)”字段中。

当您单击信号按钮时，将显示“选择信号”对话框，您可以通过该对话框定位并选择所需的信号元素。

## 共同区域表示法

Co-Region notation 可以用作并行组合片段的简写。您可以使用 “Co-Region” 子菜单将此符号添加到 列图中。您可以通过右键单击序列图中的连接器并选择 “Co-Region” 选项来显示该子菜单。有四个子选项可用：

- 开始
- 在头结束
- 在开始
- 在尾部结束



# 序列和图表版本控制

您可以创建使用其他包中的元素作为图表中生命线的序列图。在这种情况下，在版本控制下签入和签出元素包时，图表可能会损坏。这是因为在签出过程中，元素首先从模型中删除，然后重新导入，虽然它们在图表中恢复，但连接它们的任何消息都没有。

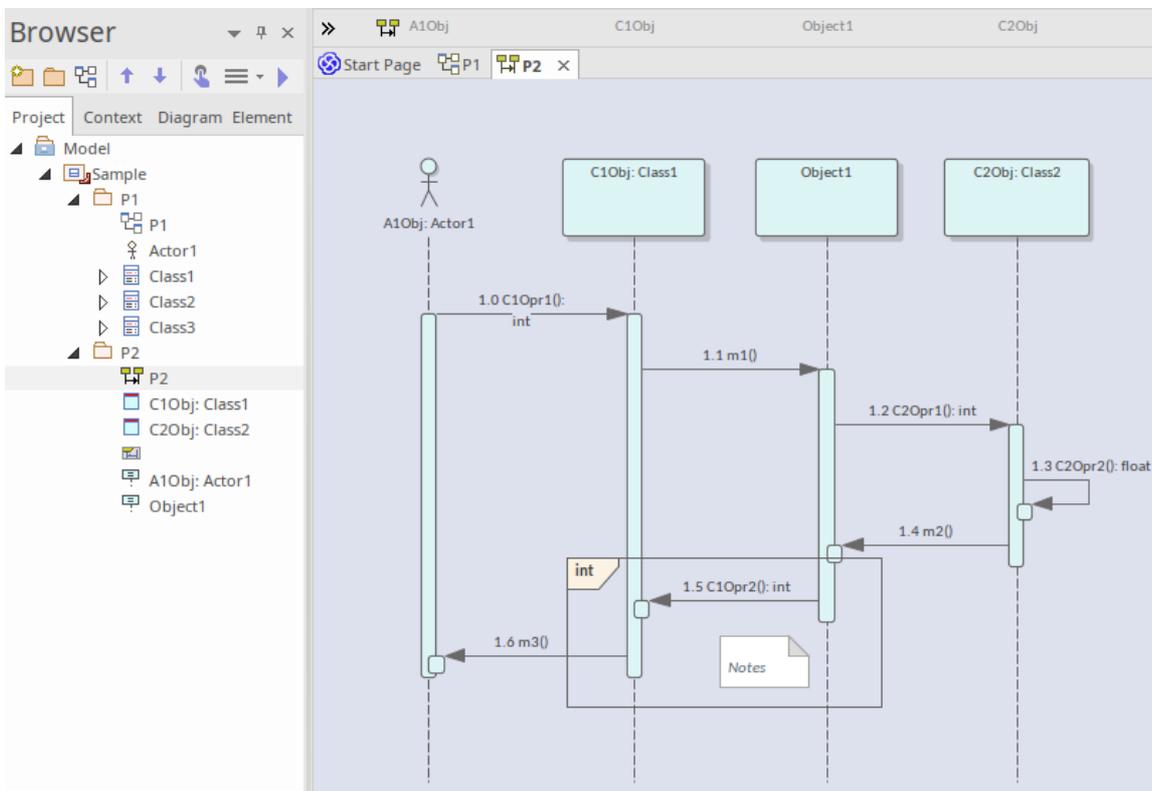
因此，如果图表及其元素位于不同的包中，则元素包通过版本控制的往返可能会损坏序列图。

解决方案是将每个类作为object拖放到序列图中 - 当您类序列到图中时，在“粘贴元素”对话框中选择“作为元素的实例（物件）”选项。这会根据所选的类元素在图表的父包中创建一个新object。然后，您在对象之间创建消息。

因此，为了保证一个序列图不被其他包通过版本控制来回损坏，请记住：

- 生命线必须是对象（即使您可以将元素作为生命线拖放到序列图中，它也不是严格符合UML的构造）
- 生命线必须与图表在同一个包中

此图显示了带有两个包的浏览器窗口：P1，包含元素，P2，包含使用这些元素的序列图。还显示了图表本身。



该图在版本控制中往返时不会损坏，因为所有生命线都是对象，并且这些对象与序列图位于同一个包中。

## 笔记

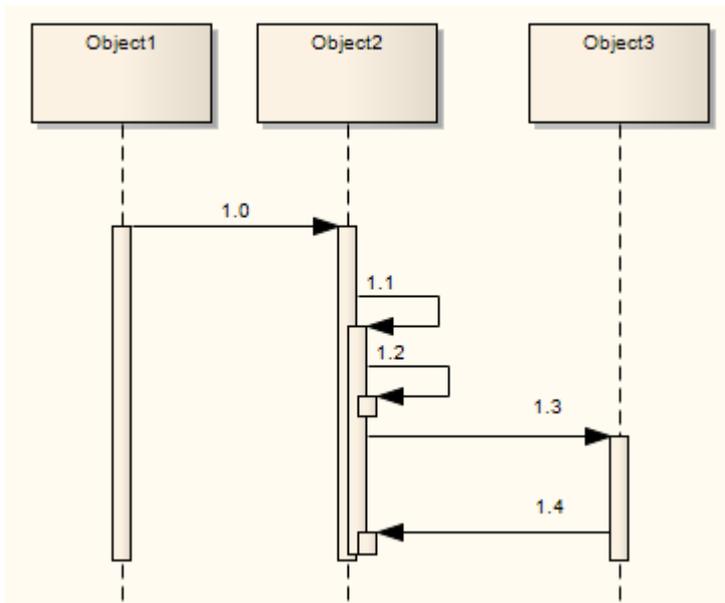
- 这些建议也适用于通讯图

# 序列元素Activations

序列图中的序列元素具有沿其生命线绘制的激活矩形。这些矩形描述了元素在整个处理期间处于活动状态的时间。可以通过右键单击序列图并选择“抑制激活”来抑制这种视觉表示。

一般，Enterprise Architect会为您计算激活周期，但在某些情况下，您可能需要微调矩形长度。序列信息上有几个上下文菜单选项可用于完成此操作。要访问上下文菜单，请右键单击消息并选择“激活”。

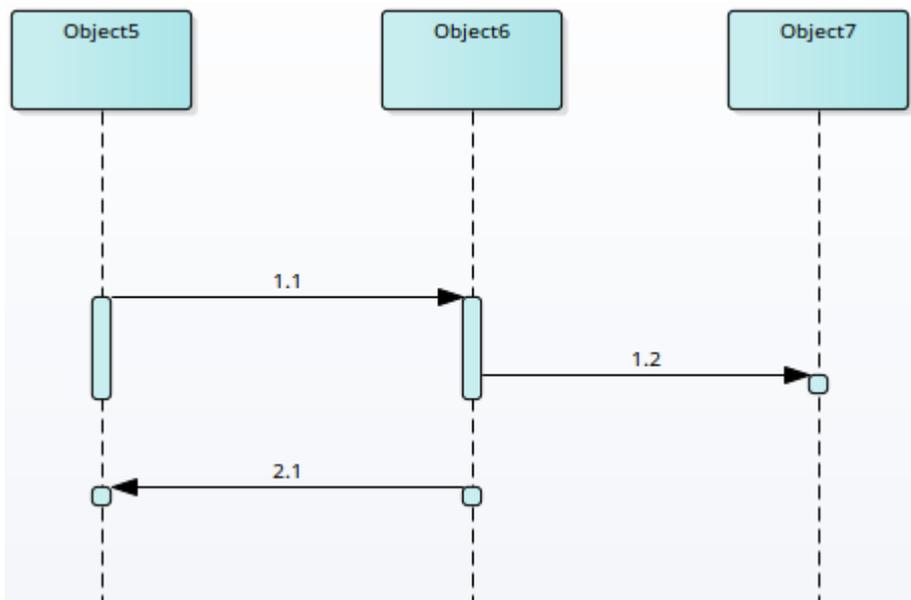
更改激活级别A更方便的方法是直接在图表上。只要合适，左箭头和/或右箭头就会显示在特定的连接器上。在此图中，请参见连接器1.3。单击箭头以提高或降低激活级别。



## 具有先前信息分支

[本节描述了在UML 2.0之前的UML中定义的表示并发消息的方法，并包含在内以支持可能仍然应用它的模型。从UML 2.0开始，符号已被片段取代。建议您考虑升级您的模型以利用片段和其他最近的改进。]

用适当的消息分组和激活级别设置生命线 and 序列消息后，您可能希望指示从生命线发出的不同信息组和不同激活级别的两条消息是分支，或同时执行。考虑这个例子：

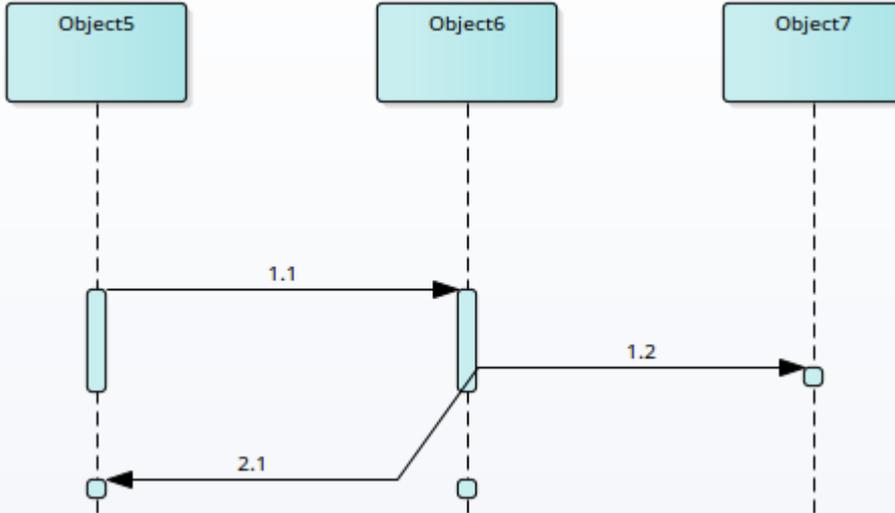


信息1 . 1从物件5传到物件6 , 然后信息1 .2传到物件7和信息2 . 1传回物件5 。消息似乎在序列1中 。 1 , 1 .2 然后 2 . 1 .但是 , 您要指出信息2 . 1 , 虽然是分开的 , 但与信息1 .2 同时存在 。

在这种情况下 :

- 右键单击后面的信息 ( 2 . 1 ) 并选择选项 “带有以前信息的分支”

信息2的源锚 . 1然后变成与信息1 .2 的源相同 , 即前一条消息 。它们是来自同生命线的独立但并发的消息 。



如果以后不需要显示消息是分支 , 请右键单击后面的消息 ( 2 . 1 ) 并取消选择 “带有先前信息的分支”选项 。

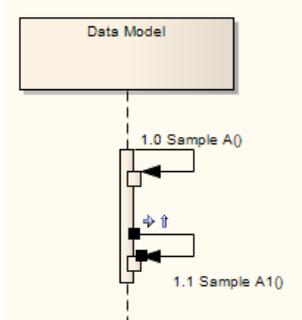
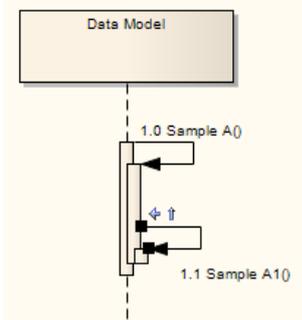
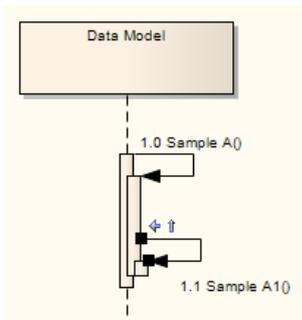
### 上下文菜单选项

选项	描述
新开始信息集团	在当前图表中开始新round的处理。 这使您能够在单个图表中描述多个处理方案。
向下扩展源激活	强制一个元素在正常处理周期之后保持活动状态。 这可以用来表达一个元素 , 它与其他进程同时继续它自己的处理。
向上扩展源激活	强制元素的激活向上。
结束源激活	在当前消息之后截断源元素的激活。 这对于表示源元素变为空闲之后的异步消息很有用。
结束目标激活	结束由 “扩展源激活”选项启动的强制激活。
提高激活级别	仅在适合使用的地方显示在上下文菜单上。 例如 , 在自发消息后 , 下一条消息默认以较低的激活级别开始 , 但上下文菜单上会显示 “提高激活级别”命令以使您能够提高其级别。
较低的激活级别	仅在适合使用的地方显示在上下文菜单上。

# 生命线激活水平

通过在单个生命线上添加激活层，可以轻松地协商复杂的处理系统并将其反映在序列图中。

## 例子

	<p>A类调用方法 Sample A，该方法又调用 Sample A1。</p> <p>要产生图中的排列：</p> <ol style="list-style-type: none"><li>1. 在搜索图表上的“工具箱项目”对话框中，单击 查找  项目”并指定显示“工具箱交互”。</li><li>2. 单击 交互关系”面板中的 社交关系”图标。</li><li>3. 点击生命线。</li></ol>
	<p>为了提高样品 A1 的激活级别，单击所选连接器的提高箭头。</p> <p>生命线现在直观地描述了在处理 Sample A 期间调用了方法 Sample A1。</p>
	<p>在此示例中，添加了更多的自我消息。</p> <p>消息 Sample A2a 是从 Sample A2 调用的，而 Sample A2 又是从 Sample A（不是 Sample A1）调用的。</p> <p>样本 A1 是从样本A中调用的。</p>

# 序列信息标签可见性

## 隐藏和显示序列消息中使用的标签

节	行动
1	右键单击序列图中的消息并选择“设置标签可见性”。 将显示“标签可见性”对话框。
2	选择或清除每个消息标签上的复选框以分别显示或隐藏。
3	点击确定按钮保存设置。

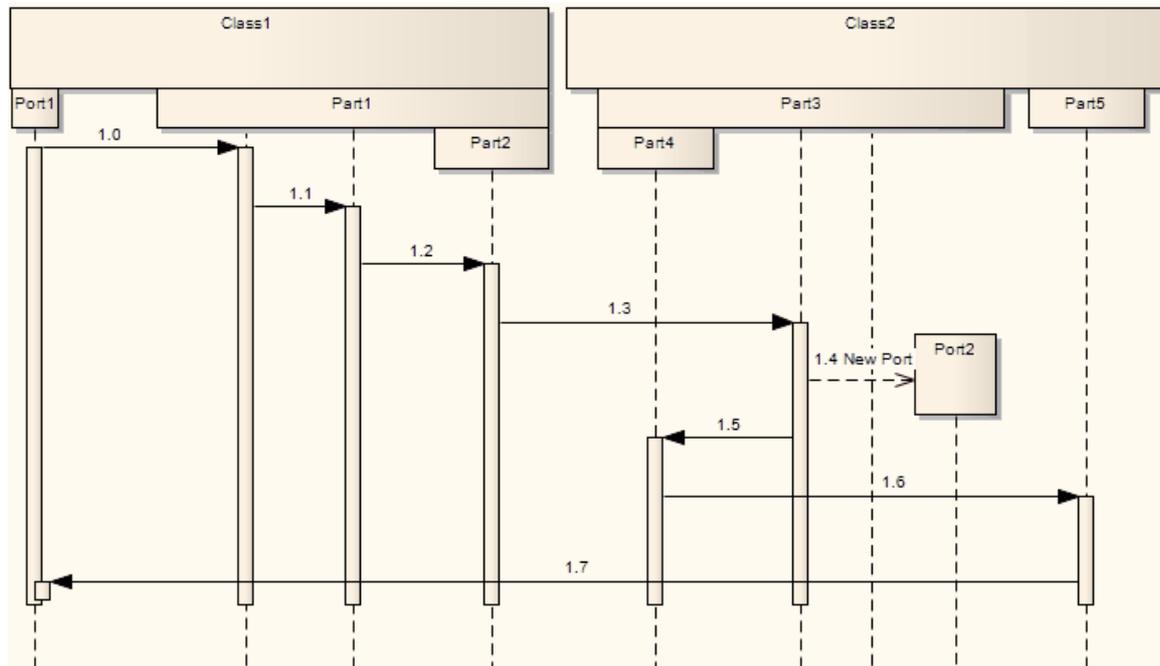
## 更改顶部边距

为了将顺序图的上序列从默认的 50 个单位更改，右键单击图并选择“设置上边距”选项。您可以将上边距设置为 30 到 250 个单位之间的任何值。然后，您可以使用此空间来添加笔记或文本元素，以在图表上提供文档。

# 内联序列元素

在序列图中，可以将现有的子部件和内联元素序列表示为它们的序列元素类，这些元素在端口元素下呈现。

## 图表部件和端口的示例序列图



## 表示部件图中的序列和端口元素

节	行动
1	右键单击包含子端口或部件的序列元素，然后选择“特征 交互点”。特征显示在“交互Points”选项卡上。
2	选中每个组件的部件或显示端口，然后单击关闭按钮。

# 通讯图表

通讯图是A运行时以与序列图相同的方式显示元素之间交互的图。然而，通讯图用于可视化对象间的关系，而序列图更有效地可视化随时间的处理。

通讯图使用有序的、带标签的关联来说明处理过程。表示加工的顺序和嵌套很重要。编号方案A是：

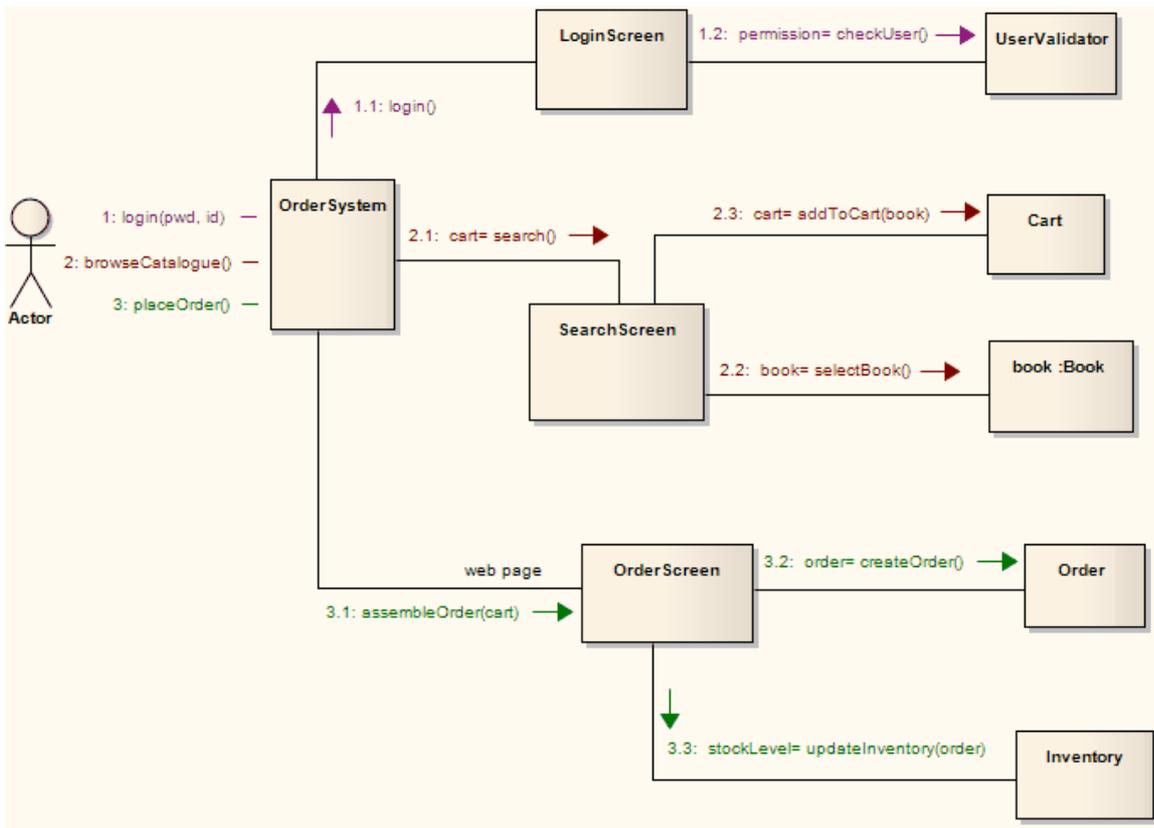
- 1
- 1.1
- 1.1.1
- 1.1.2
- 1.2 · 依此类推。

A新的数字段开始一个新的处理层，相当于一个方法调用。

您可以从图表工具箱的“通讯”页面图表通讯图元素和连接器。

## 示例图表

此示例说明了通讯object实例之间的通信图。注记使用消息级别来捕获相关流，以及消息的不同颜色。



## 通讯图表元素工具箱Icons

图标	描述
 Actor	参与者是系统的用户；用户可以指人类用户、机器，甚至是模型中的另一个

	系统或子系统。
 Object	物件是类在运行时的特定实例。
 Boundary	A是一个边界的物件，它模拟一些系统边界，通常是一个用户界面屏幕。
 Control	元素控件表示组织和安排其他活动和元素A控制实体或管理器。
 Entity	实体是一种原型化的物件，它对捕获系统中的信息或知识的存储或持久性机制进行建模。
 Package	包用于组织您的项目内容，但是当添加到图表时，它们可以用来描述您的模型的结构和关系。

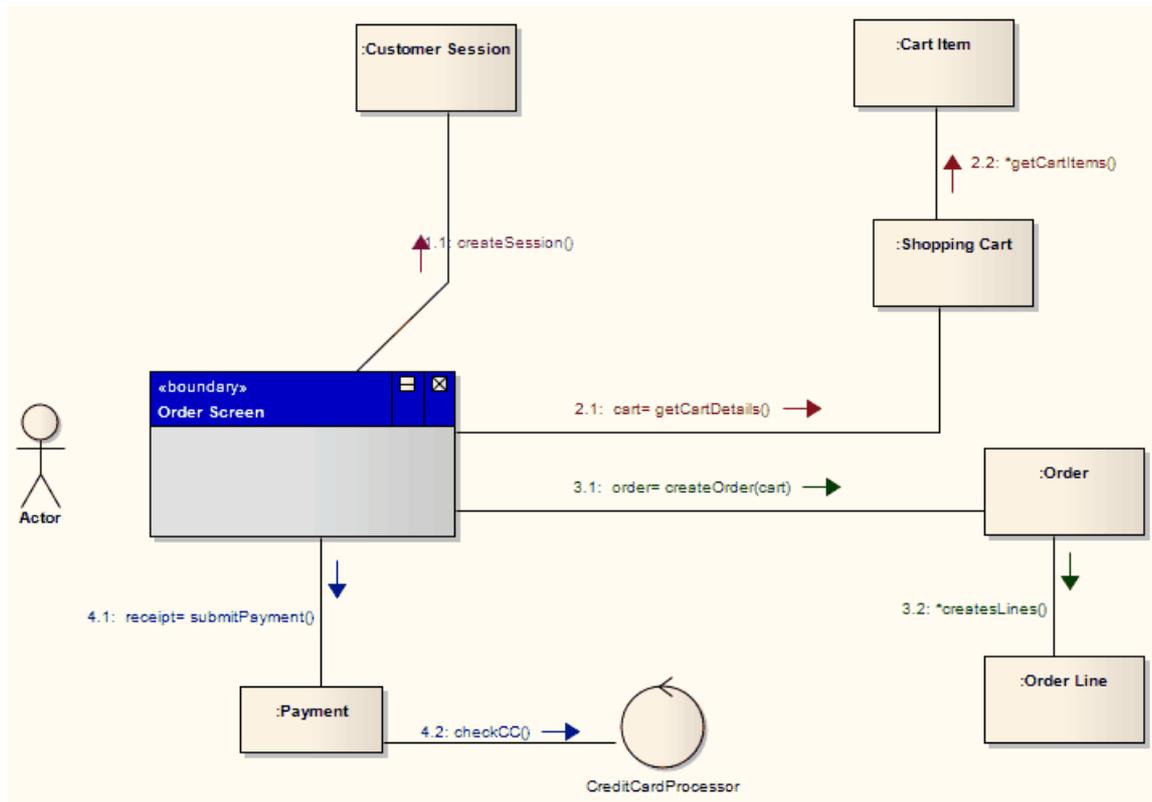
### 通讯图表连接器工具箱Icons

图标	描述
 Associate	关联意味着两个模型元素具有关系，通常实现为一个或两个类中的实例变量。
 Nesting	嵌套连接器的其他元素是一种替代图形，而不是表示包含或嵌套元素内的元素。
 Realize	A连接器表示源object实现或实现其目标object。

# 通讯图表颜色

可以为每个消息集使用不同的颜色突出显示通讯图中的特定消息流。

## 突出显示通讯图中的颜色



节	行动
1	选择 开始>应用程序> 首选项 > 首选项 >通讯颜色”。 显示 首选项”对话框的 通讯信息着色”页面。
2	选中 使用通讯信息着色”复选框。
3	单击每个 信息n”字段的下拉箭头，然后为每个消息组选择所需的颜色。
4	单击关闭按钮。 在您的通讯图上，序列消息都以不同的颜色显示，如图所示。

## 消息 ( 通讯图表 )

通讯图中A信息与序列图中的信息在含义上是等价的。它意味着一个object使用另一个object的服务，或向该object发送消息。通讯Enterprise Architect中的通讯消息总是与object实例之间的关联连接器相关联。始终先创建关联，然后将信息添加到连接器。

通过单击并拖动消息文本，可以将消息拖动到合适的位置。

通讯消息被排序以反映图表的顺序。编号方案应嵌套每个事件的编号。排序方案A是：

1

2, 2.1, 2.2, 2.3

3

这将表明事件 2.1、2.2 和 2.3 的单一序列发生在由事件1发起的操作中。这是Enterprise Architect应用的默认模式。

或者，序列可以是：

1

2

2.1, 2.1.1, 2.1.1.1

2.2, 2.2.1, 2.2.1.1

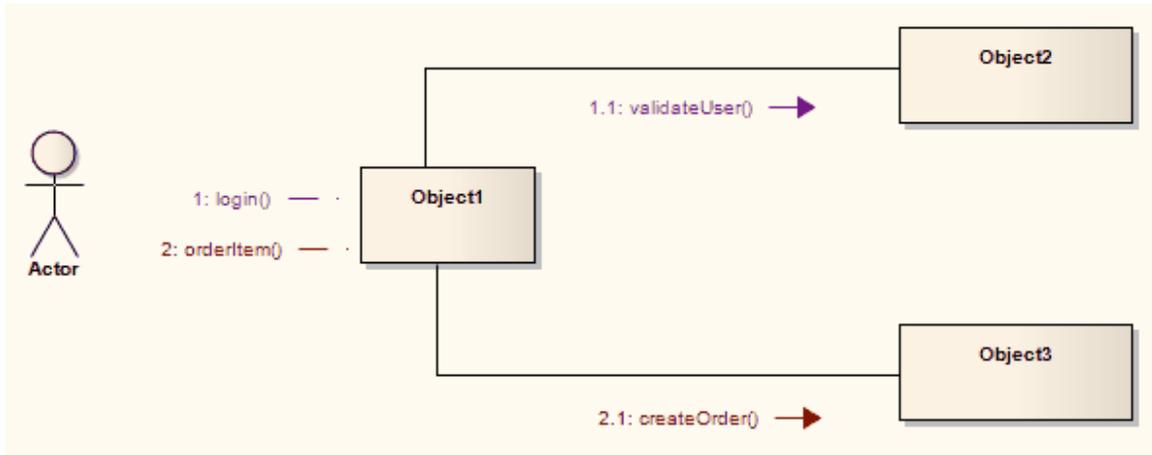
3

这表明事件 2 和 2 可以启动两个事件序列1和 2.2 是单独的序列，而不是一个序列中的连续事件。您可以使用信息的属性窗口和“序列通信”对话框设置序列模式和顺序。

如果目标object是一个类或具有它的实例分类器集，则可能的消息名称的下拉列表包括基本类型的公开操作。

# 创建通讯信息

## 创建通讯信息



节	行动
1	开一个图 ( : 通讯 · 分析活动 · 物件交互状态机 )
2	添加所需的对象。
3	在每对通信的对象之间添加关联关系。
4	右键单击关联以显示上下文菜单。
5	选择适当的选项将信息从一个object添加到另一个对象。
6	当信息的属性窗口显示时，输入名称和任何其他所需的详细信息。
7	点击确定按钮。 信息被添加，连接到关联和物件实例。
8	将信息移动到所需位置。

## 重新排序消息

在构建您的通讯图时，经常需要创建或删除信息组并重新序列消息的顺序。有两个显示可以帮助您执行这些任务：信息的属性窗口和“序列通信”对话框。

### 组织信息组

如果您有多个格式为1的消息。例如，1, 1.2, 1.3, 1.4，但想在第三个信息（即1.1, 1.2, 2.1, 2.2, 2.3），您可以将系列中的信息更改为开始组消息。

节	行动
1	双击信息名称。 信息显示的属性窗口。
2	要使所选信息成为新组的开始，请选中“开始新组”复选框。
3	如果需要，请在信息的注记窗口中键入说明性注记。 您可以使用字段顶部的注记工具栏来格式化文本。
4	单击保存图标以保存更改。

### 消息序列

在更大、更复杂的图表中，您可能必须在组中使用更深层次的消息；例如，1.1.1。1, 1.2。1.1。您可能还需要更改消息的序列，例如将信息1.3更改为信息1。1。

节	行动
1	在您执行以下操作后选择“序列通讯消息”选项： <ul style="list-style-type: none"> <li>选择“设计&gt;图表&gt;选项”功能区选项，或</li> <li>右键单击图表背景，或</li> <li>右键单击信息</li> </ul> 将显示“通讯消息”对话框。
2	单击信息进行调整，然后在对话框底部单击： <ul style="list-style-type: none"> <li>上移或下移（手形）按钮可将信息按顺序上移或序列（例如，信息1.2到信息1.1或1.3）</li> <li>向左移动或向右移动（手）按钮可将信息向上或向下移动一个级别（例如，信息1.2.1到信息1.2或信息1.2.1.1）</li> </ul>
3	重复步骤2，直到信息序列和级别符合您的要求。 您可能需要调整其他信息编号（按组、序列或级别）以适应您所做的更改。
4	单击确定按钮保存更改。

## 交互图表

将交互关系图可视化其他交互视图之间的服务流程图由于交互是活动图表的变体，大多数的符号都是构建图表的过程。

决策点、分叉、汇合、开始点和结束点是相同的。然而，代替活动元素，使用了两种类型的矩形元素：

- 通讯组件、内交互的交互，可以是四种类型中的任何一种（序列、通讯或显示交互）
- 交互发生是指在视觉上以现有的参考图表示的交互图：它们是框架，带有框架的标题空间；图表名称显示在框架内容中

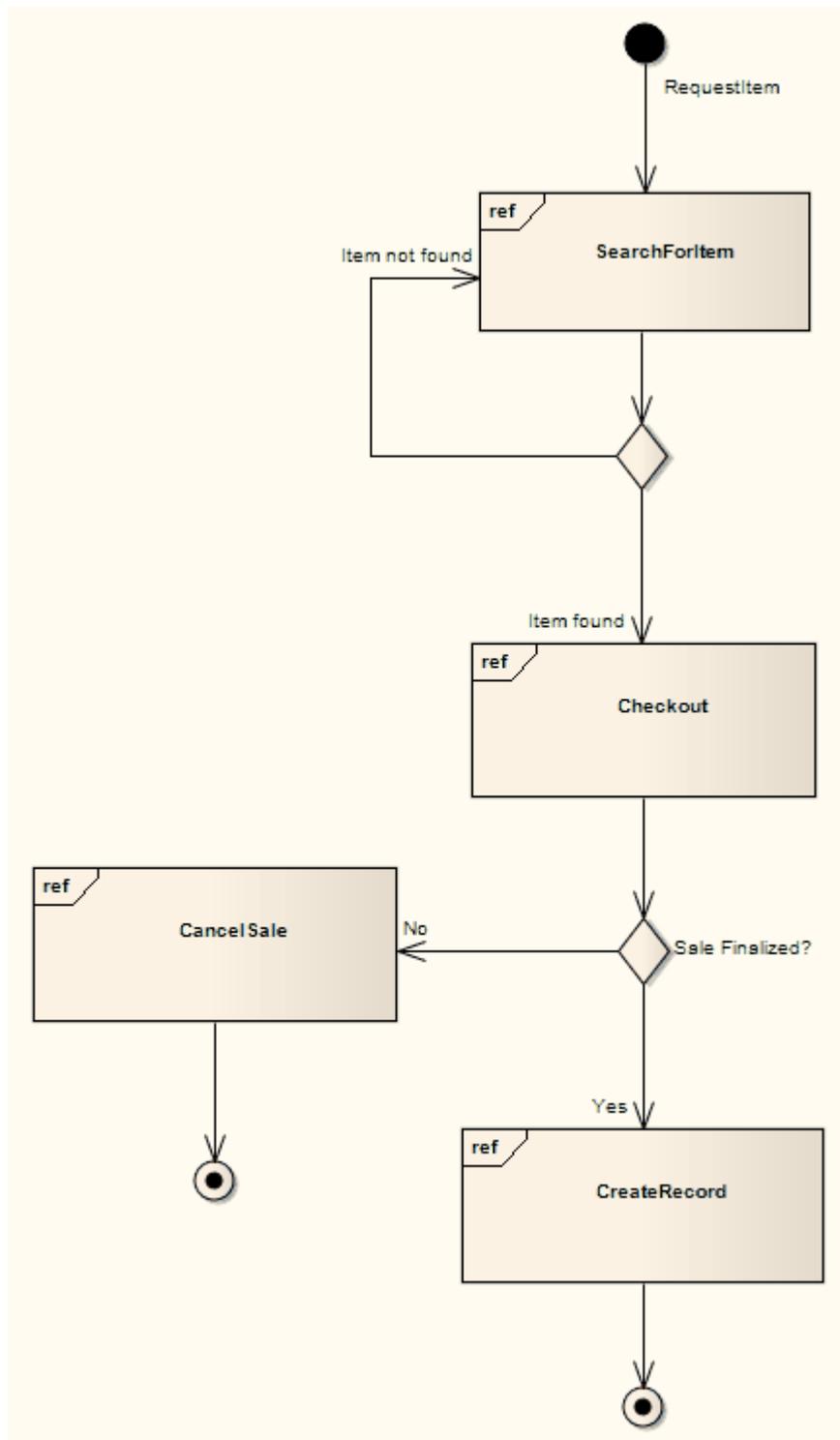
要创建一个交互发生，只需将浏览器上的一个交互图拖到您的交互图上即可。ref 框架显示，封装了交互图的一个实例。

您可以从“交互工具箱”的“活动视图”元素和连接器的页面图表工具箱。

## 示例图表

此图描绘了一个示例销售过程，在示例中显示了一个交互销售过程，其中显示了抽象的交互-发生率。

该图看起来与活动图非常相似，并且以相同的方式概念化；作为流程交互，必须交互各自的流程交互，才能进入到前一个流程的交互过程。



### 交互图表工具箱元素

图标	描述
 Partition	分区元素用于A逻辑上组织元素。
	决策是指示条件进展点A元素：如果条件为真，则处理以一种方式继续；如果

 Decision	没有，那么另一个。
 Send	发送元素用于描述发送信号的动作。
 Receive	接收元素用于定义请求A接受或接收。
 Synch	同步状态A用于指示并发路径已同步。它们用于拆分和重新加入并行处理周期。
 Initial	起始元素的 d 定义了在活动调用流。
 Final	终点活动元素，表示完成；到达终点，所有执行都被中止。
 Flow Final	活动流终点元素表示系统的退出，与活动终点，表示系统的完成。
 Expansion Region  Interruptible Region	Enterprise Architect支持两种类型的区域元素：扩展区域和可中断活动区域。 一个扩展区域围绕一个对输入数据多次施加的过程，对输入集中的每个元素一次。 一个可中断的活动区域围绕着一组活动元素，所有这些元素都受到某些中断的影响，这样在该区域内经过的所有令牌都会在中断被引发时终止。
 Exception	异常处理程序元素定义了发生异常时要执行的操作组。
 Merge	A Merge节点a Merge together a flow paths in活动,分析和选择交互图表。
 Fork/Join	分叉/汇合元素A用于：1) 将单个流拆分为多个并发流，2) 连接多个并发流或3) 将多个传入流连接和分叉到多个传出流。
 Fork/Join	分叉/汇合元素A用于：1) 将单个流拆分为多个并发流，2) 连接多个并发流或3) 将多个传入流连接和分叉到多个传出流。

### 交互图表连接器工具箱

图标	描述
 Control Flow	控件是连接两个节点的连接器，用于对活动转换进行建模。
 Object Flow	一个物件流连接两个元素，通过它的特定数据，模拟一个主动过渡。
 Interrupt Flow	Interrupt Flow 是一个连接，用于定义 Exception Handler 和 Interruptible活动区域的连接器的两个UML概念。

# UML元素

UML元素是模型的构建块。它们包含在存储库中，并用关系连接的图表进行描述，以创建描述企业、业务或软件系统的叙述。每个元素都有一个类型，该类型决定了它的表示方式以及管理它如何与其他元素连接的规则。元素在浏览器窗口中以层次结构显示，每个元素在定义被建模的系统中扮演着角色。它们被分为结构或行为元素类型，每种类型都可以在系统表示的任何阶段使用。例如，活动可用于定义组织执行业务函数的方式，或定义计算机算法中的步骤。

## 行为图表元素

生命线描述系统或业务流程的行为特征。在行为图上可以出现的元素包括活动、使用图、交互、状态机和状态用例。

## 结构图表元素

结构图描述了构成系统或函数的结构元素。可以出现在结构图中的元素包括类、部件、接口、节点和包。

## 行为图表元素

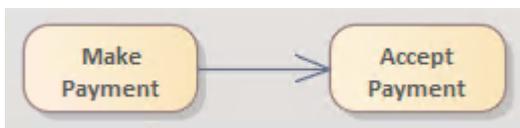
本节详细描述了在Enterprise Architect中使用行为图进行建模时常用的元素。

# 行动



## 描述

行动元素描述系统内发生的基本流程或转变，是活动图中的基本功能单元。行动可视为活动的子项；两者都表示流程，但活动可包含多个步骤或可分解的流程，每个步骤或流程都可由行动体现。行动不能进一步细分或分解。



为了进行模拟，您可以在元素的属性窗口的“行动”选项卡上定义基本（原子）行动的效果，在“影响”字段中使用JavaScript表达式来定义效果的持续时间，并选择在图表上显示效果。行动可以进一步定义前置条件和后置条件注记。

某些属性可以在行动上以图形方式描述。当您第一次将“行动”图标从工具箱拖到图表上时，系统提示您从要创建的更常见行动类型列表中进行选择。如果您在此列表中选择“其它”选项，则会显示“新建行动”对话框；该对话框中的“其它”下拉列表使您能够从完成类型列表中选择更专门行动行动类型。

<ul style="list-style-type: none"> <li>Atomic</li> <li>Call Behavior</li> <li>Call Operation</li> <li>Accept Event</li> <li>Accept Time Event</li> <li>Send Signal</li> <li>Write Variable</li> <li>Value Specification</li> <li>Create Object</li> <li>Destroy Object</li> <li>Other</li> </ul>	<ul style="list-style-type: none"> <li>AcceptCall</li> <li>AcceptEvent</li> <li>AcceptEvent Timer</li> <li>AddStructuralFeature Value</li> <li>AddVariable Value</li> <li>Broadcast Signal</li> <li>ClearAssociation</li> <li>ClearStructuralFeature</li> <li>ClearVariable</li> <li>CreateLink</li> <li>CreateLinkObject</li> <li>CreateObject</li> <li>DestroyLink</li> <li>DestroyObject</li> <li>Hyperlink</li> <li>RaiseException</li> <li>ReadExtent</li> <li>ReadIsClassifiedObject</li> <li>ReadLink</li> <li>ReadLinkObjectEnd</li> <li>ReadLinkObjectEndQualifier</li> <li>ReadSelf</li> <li>ReadStructuralFeature</li> <li>ReadVariable</li> <li>ReclassifyObject</li> <li>RemoveStructuralFeature Value</li> <li>RemoveVariable Value</li> <li>Reply</li> <li>SendObject</li> <li>SendSignal</li> <li>StartOwnedBehavior</li> <li>TestIdentity</li> <li>ValueSpecification</li> <li>WriteLink</li> <li>WriteStructuralFeature</li> <li>WriteVariable</li> </ul>
--	---

如果您后来认为该行动类型不合适，您可以在属性窗口的“行动”选项卡上进行更改 - 从“种类”下拉列表中选择所需的新类型。对于值规范行动，您也可以在此选项卡上设置值。

传出和传入行动数据值可以由行动销表示。对于基本行动以外的行动类型，您还可以分配行动销来代表特定属性。

行动也可以被描述为扩展节点以表明该行动由扩展区域组成。

如果您为行动元素定义了决策表，则可以在元素的上下文菜单上选择选项，将图表上的元素呈现为决策表，将规则显示为行或列。您还可以将元素恢复到正常的元素形状。

## 工具箱icon

A rectangular icon with a light gray background and a dark gray border. Inside the rectangle, there is a small blue square icon followed by the text "Action" in a dark gray font.

## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p. 443) 指出：

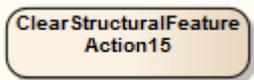
行动是行为中直接或间接包含的可执行功能的基本单元。行动的执行代表了建模系统中的一些转换或处理，无论是计算机系统还是其他系统。

OMG Unified Modeling Language规范 (v2.5. 1 , p. 443) 还指出：

一个行动可以接受输入并产生输出，分别由行动的 InputPins 和 OutputPins 指定。一项行动的每个销都指定了该行动的特定输入或输出的类型和行动。

# 行动类型

行动元素非常通用。Enterprise Architect支持广泛的特定行动类型，您可以使用它们来表示或制定离散object、操作或行为。大多数类型的行动被描绘为包含行动类型和行动名称的圆角矩形，如图所示。



## 行动元素表示法

某些类型的行动元素有自己的特定符号；例如：

行动类	符号
接受事件	
接受事件定时器	
呼叫行为	
呼叫操作	
发送信号	

## 接受事件行动

一个行动元素具有分配给它的一个可选择的输出结果行动销，以及一个或多个触发器表示行动所接受的事件类型。您可以在属性窗口的“触发器”选项卡上定义触发器。在模拟中，没有触发器的行动将在行动元素处阻止模拟。

字段	行动
名称	类型触发器的名称。

<p>类型</p>	<p>单击下拉箭头并选择触发器类型：调用、更改、信号或时间：</p> <ul style="list-style-type: none"> <li>调用-指定事件为调用，通过调用操作向关联object发送消息</li> <li>更改-指定事件是一个 ChangeEvent，这表明转换是属性值更改的结果</li> <li>信号指定事件为SignalEvent，对应接收到一个异步信号实例</li> <li>时间-对应于 TimeEvent；它指定了一个时刻</li> </ul> <p>状态机的代码生成当前仅支持更改和时间触发事件，并且需要规范值。在模拟中，每个触发器都应该有一个信号。结果将是接受信号。</p>
<p>规格</p>	<p>指定引发转移的事件转移</p> <p>.</p> <p>对于行动，您可以在此字段中键入一个JavaScript表达式，计算要等待的滴答数。</p>

## 发送信号行动&广播信号行动

A发送信号行动有一个指定的目标行动销和一个信号。信号可以具有将其属性参数绑定为参数的输入行动销。例如：

```
::Sender: sig.binding_To_sl: Integer
```

在模型模拟中，行动动作将参数的值传输到创建的信号实例的属性中。目标行动销可以有一个空的object，将信号发送到模拟空间的根。如果没有目标行动销，模拟将停止在行动行动。如果目标有一个物件，信号将被发送到该物件。您必须在物件的分类器中指定目标行动销物件销

BroadcastSignal行动类似于A行动，只是它没有目标行动销。在模拟中，它总是将其信号发送到模拟数据的根。您可以使用模型的属性窗口的“信号”选项卡来元素要发送的信号和要传达的相关参数。

字段/按钮	行动
<p>信号</p>	<p>单击  并从“选择信号”对话框中选择所需的信号。</p>
<p>属性</p>	<p>单击下拉箭头并选择与参数关联的属性（如先前在信号元素中创建的）。</p>
<p>价值</p>	<p>类型属性的适当值。</p>
<p>添加</p>	<p>单击此按钮并从“选择销”对话框中选择行动销的移动销售商，以识别信号的信号。</p> <p>要指定多个行动销，请在选择每个行动销员时按 Ctrl 键。</p>
<p>节省</p>	<p>单击此按钮以保存您的更改。</p>

## 呼叫行为

A行动具有诸如活动之类的行为，以及将放置返回值的可选行动销结果。行动也可以将其论点行动销的值转移到它的行为中，如果它们被绑定在一起的话。在模拟中，如果行动没有行为，则模拟被阻止。

## 发送对象行动

A行动销将一个物件的副本从请求行动销中发送到目标行动销。在模拟中，行动必须同时具有行动销，否则模拟在动作处被行动。

## 结构特征行动

行动作用于建模结构特征，例如活动A端口、部件或属性或物件的分类器的物件，您可以在行动元素中识别这些特征。Enterprise Architect支持这些类型的结构特征行动：

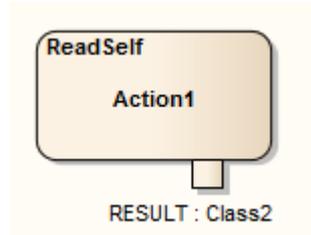
行动	描述
添加结构特征值	<p>需要一个object输入行动销，目标object将被输入，并需要一个结果输出行动销来保存读取的结果。如果object端口在运行时没有值，则进程将在移动时行动。</p> <p>在你的设计中，端口应该连接到一个端口的模型活动端口或一个物件的物件节点。必须正确设置端口或物件节点的属性，设置端口值，以便在动作生效时行动输入值。</p> <p>结果行动销可以连接到输入端口或行动销。例如，它可以在下一个行动中用作输入值。</p>
清晰的结构特征	清除结构属性或物件或活动的结构端口的单个值，并将结构特征的值设置为空。
读取结构特征	<p>与端口类似，只是该值不需要端口。</p> <p>在模拟中，如果物件的端口在运行时没有值，则模拟将在移动时行动。</p>
移除结构特征值	<p>与 ClearStructuralFeature 类似，只是它调用一个值行动销来输入一个值，如果该值与指定的结构特征的值匹配，它将值设置为 null。</p> <p>如果值不匹配，则行动不清除结构特征值。</p>
写结构特征	与 AddStructuralFeatureValue 相同。在模拟中，端口值必须设置为在模拟运行移动时行动输入值。

## 设置结构特征

节	行动
1	右键单击行动元素并选择 高级 "设置结构特征：添加'。
2	<p>在 选择属性"对话框 ( 选择&lt;项目&gt;"对话框的变体 ) 中，浏览或搜索适当的结构特征，然后双击它。</p> <p>特征名称和位置显示在 设置结构特征"对话框的 结构特征"字段中。</p>
3	点击确定按钮保存设置。

## 阅读自我

A行动动作将它的宿主object名读入它自己的端口。您必须为结果设置输出行动销。



行动必须在一个类中，该类在运行时被实例化。当模拟通过行动端口时，它会将类实例的名称放入结果中。物件是一组物件行动中的一个，具有 CreateObject 和 DestroyObject。

# 可变行动

变量行动与过程中对象的行为和动作的模拟密切相关。它们有一个标记值变量形式的关联变量，其值是运行时的物件名称。那是：

`sim.ObjectName`

变量行动不仅将变量作为物件提供，而且作为物件的属性（如属性或端口）物件。例如：

`sim.a.a1`

参数 `a.a1` 可以有一个整数值。

可变行动包括：

- ReadVariable
- 写变量
- ClearVariable
- 添加变量值
- 移除变量

## ReadVariable

A `ReadVariable`行动作为输出端口的行动销。端口的值将是要读取的结果，这是读取的变量的副本。因此，它不受变量值更改的影响。例如变量为`sim.物件.a`，值为3，读取后其值变为5，则读取的值仍为3。

读之前：

```
sim.物件.a = 3;
```

```
sim.Action1.result = null;
```

看完之后：

```
sim.物件.a = 3;
```

```
sim.Action1.result = 3;
```

变量值改变后：

```
sim.物件.a = 5;
```

```
sim.Action2.value = 3;
```

在该示例中，该值是通过物件流连接器连接到端口的结果端口的 `Action2` 的端口。

## 写变量

这个行动有一个价值行动销作为一个输入端口。端口的值将被写入其变量。结果值是来自值端口的变量的副本。

## ClearVariable

此行动清除变量的所有值，该变量要么是一个物件，要么是一个值。

## 添加变量值

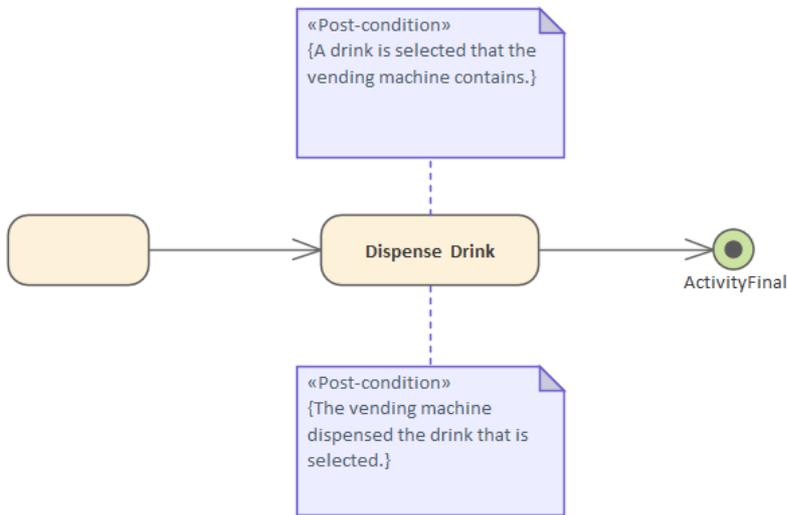
这个动作实际上与行动行动相同，因为模拟的当前变量不支持多个值。

## 移除变量值

此动作实际上与行动ClearVariable行动，因为模拟的当前变量不支持多个值。

## 当地的前置/后置条件

行动可以进一步定义为前置条件和后置条件注记，限制一个行动的进入和退出。



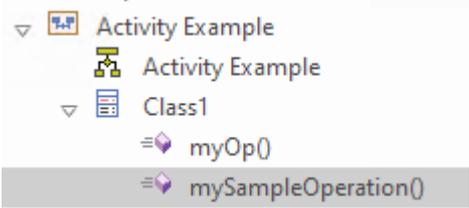
## 将约束附加到行动

节	行动
1	右键单击行动并选择 新子元素 附上约束选项。 在图表上创建A注记，连接到行动。
2	右键单击注记并选择 视图属性"选项。 将显示 约束"对话框。
3	在 约束类型"字段中，单击下拉箭头并选择所需的约束类型。
4	在 约束"字段中，输入约束文本。
5	单击确定按钮以保存约束。

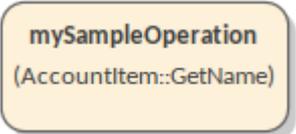
# 类操作在图表中

类可以由任何图表（例如活动、自定义或分析图表）上的操作行动表示。当操作显示为行动时，元素的表示法会显示操作的名称，前缀是它来自的类的名称。

## 向图表中添加操作

节	行动
1	打开目标图。
2	从浏览器窗口打开一个类并找到要添加到图表中的操作。
3	将操作拖到图表上。 
4	将操作添加到图表后，行动类似于： 

## 更改一个行动所指的操作

节	行动
1	右键单击行动并选择 高级 设置操作选项。 将显示 设置操作”对话框。
2	如有必要，在 转到命名空间”字段中，选择包含该操作的模型。 浏览操作。
3	找到操作后，双击它。 行动更新以显示新的分类器和操作名称。 

## 注记

- 如果要在浏览器窗口中找到创建行动的操作，请右键单击图表中的行动并选择 查找 | 在项目浏览器中定位操作”选项
- 如果要显示包含操作的类的先前生成的代码，请单击图表中的行动，然后按**Ctrl+E**或**F12**；将显示 代码编辑器”视图，其中包含为类生成的代码（如果没有为类生成代码，则不会显示 代码编辑器”）
- 在模拟中，**CallOperation**行动必须有一个调用操作和一个目标**object**行动销，该操作属于来自目标行动销的**object**；如果不设置这些属性，模拟将在行动处被阻止
- 如果 **CallOperation** 的 “名称” 属性为空，则类操作名称将显示在其位置。如果操作的名称被修改，则显示的名称将更新以反映该更改。

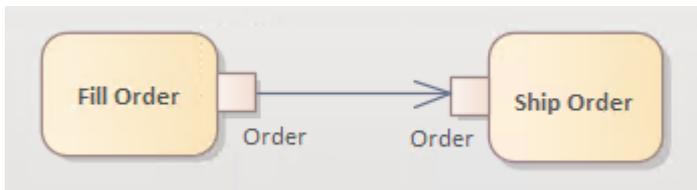
# 行动销



## 描述

行动销用于定义传入和传出行动数据值。输入销为行动提供值，而输出销包含该行动的结果。

行动销在这里用来连接两个行动：



行动销可以进一步定义为定义异常参数、流或状态。将状态与销关联起来定义了输入或输出值的状态。例如，销可以称为“订单”，但状态可以是“已验证”或“已取消”。

要将行动销添加到行动中，请右键单击该行动以显示上下文菜单，然后选择“新子元素|行动销”选项。（您还可以分配行动销，以定义行动的特定属性。）

行动销的属性窗口有一个“销”选项卡，您可以在其中定义该销的具体操作。

销作为调用行为行动和调用操作行动的参数 -销名称和参数显示在行动元素属性窗口的“参数”选项卡上。当某个行动与模型中的有效行为关联时，关联行为的参数会列在“参数”字段下拉列表中，以方便自变量和参数之间的一对一映射。“销”选项卡的“参数值”面板中的字段仅对属调用行动的销启用，并且仅当该行动与具有有效参数的有效行为相关联时才启用。要观察这一点：

1. 创建一个活动元素并给它一个活动参数（右键单击它并选择“新子元素|活动参数”）。
2. 创建一个行动并将“Kind”属性设置为“CallBehavior”（在行动元素的属性窗口的“行动”选项卡上）。
3. 将活动元素设置为行动的分类器（在行动的属性窗口中，单击“元素”选项卡，在“高级”部分中单击“分类器”浏览按钮，然后找到并选择活动“选择<项目>”对话框）。
4. 特征窗口立即显示在“交互点”选项卡上。选择“显示拥有/继承”复选框；选择此选项后，活动参数将列在“已定义元素”面板中。选中活动参数旁边的复选框。
5. 行动元素现在有一个代表参数的行动销，活动参数作为参数的参数。

您还可以在元素属性窗口的“销”选项卡上更改行动销的 objectState 属性。

## 分配行动销

除了将行动销添加到任何行动之外，您还可以将专门的输入或输出行动销行动给具有特定类型（即非基本或原子行动）的行动销。这些输入/输出销表示行动的各种属性——它们在图表上不作为结构可见，除非它们以前被添加，但在浏览器窗口中作为行动的属性列出。

您只能销已添加或分配给行动的行动，或者是专门为分配给行动而创建的行动。

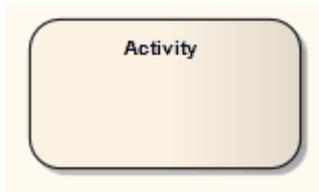
### 将行动销分配给行动

节	行动
1	单击行动中的移动，然后在属性的窗口中单击 行动”选项卡。单击 种类”字段中的下拉箭头并确保您具有正确的行动类型，然后单击 保存”图标。
2	单击 无素”选项卡，然后选择构造型属性组。该组包含不同的字段，具体取决于行动类型。通过键入或浏览适当的object名称或选择复选框来填充这些字段。 如果您使用浏览器屏幕，您可以浏览并分配现有对象 - 在这种情况下，行动销- 或单击 加新”按钮并创建并分配新的行动销。
3	注记行动销不会显示在图表上，而是显示在浏览器窗口中的行动元素下。 点击确定按钮返回属性窗口。

### 注记

- 要查看指定的行动销的确切位置，您可以在属性窗口中右键单击销名称，然后选择 在项目中查找浏览器”选项

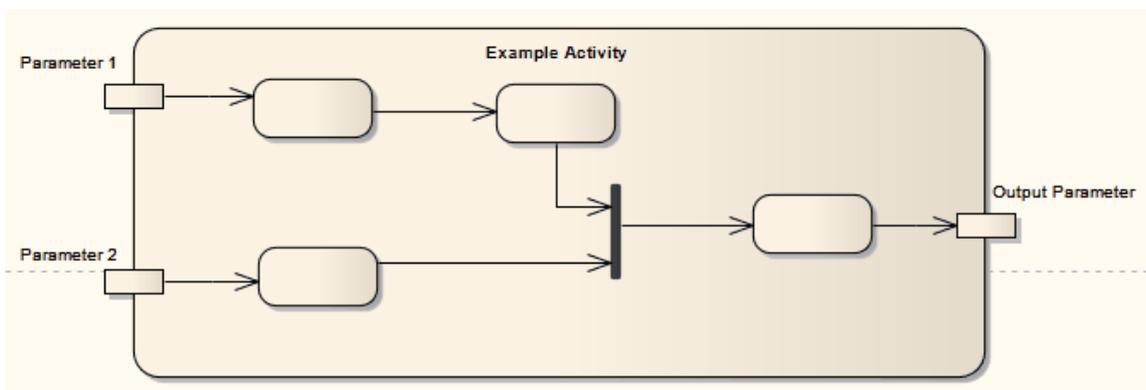
# 活动



## 描述

活动组织和指定从属行为的参与，例如子活动或行动，以反映流程的控制和数据流。活动在活动图中用于各种建模目的，从系统设计的过程型应用程序开发到组织结构或工作流的业务流程建模。

这个简单的活动图表包含行动元素并包括输入参数和输出参数。



您可以在创建期间或以后编辑期间将活动定义为复合元素。在创建复合活动元素时，应用创建结构活动元素的机制更简单，这减少了工作步骤的数量。如果转换现有的活动元素，右键单击元素并选择“新子图表|复合结构图表”选项。

某些属性可以在活动上以图形方式描述。活动中的行动可以通过活动分区进一步组织。

一个活动也可以描述为一个 Expansion 节点，表示该活动由一个 Expansion 区域组成。

如果您已经为活动元素定义了决策表，您可以在元素的上下文菜单上选择选项，将元素在图表上呈现为决策表，将规则显示为行或列。您还可以将元素恢复为正常的元素形状。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , pp.373-374) 指出：

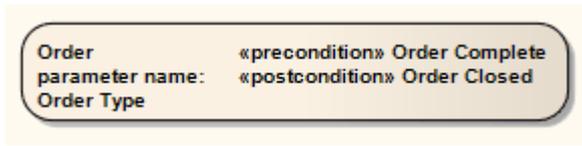
活动是一种行为，指定为下属单元的排序，使用控制和数据流模型。由这些模型协调的从属行为可能会在模型中的其他行为完成执行、因为对象和数据变得可用或者因为事件在流外部发生时被启动。执行流程被建模为由 ActivityEdges 连接的 ActivityNodes。ExecutableNode 是从属行为的执行，例如算术计算、对操作的调用或 object 内容的操作 (...)。ActivityNodes 还包括控制流结构，例如同步、决策和并发控制。

令牌在活动中没有明确建模，但用于描述活动的执行。object 标记是流过 ObjectFlow 边缘的值的容器（一些 object 标记可以流过 ControlFlow 边缘，由建模者指定，请参阅第 15.4 节中 ObjectNodes 的 isControlType）。没

有值的object标记称为空标记。控制令牌影响 ActivityNode A 执行，但不携带任何数据，并且仅流过 ControlFlow 边缘。每个标记都与其他标记不同，即使它包含与另一个标记相同的值。

## 活动符号

某些属性可以在活动元素上以图形方式描述，如下所示：



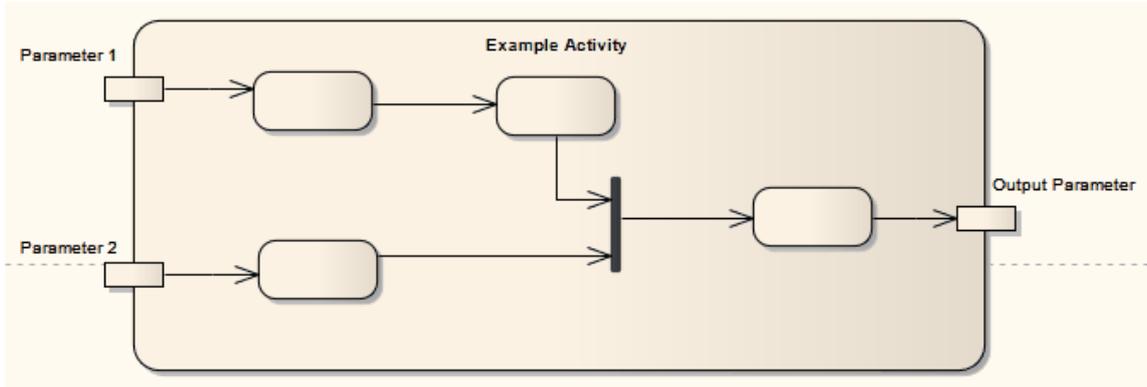
要定义这些属性，请右键单击活动并选择“属性”选项，然后选择“对话框的 属性高级”选项卡。

您还可以使用JavaScript表达式定义活动的持续时间（等待的滴答数）。打开属性窗口，单击“行为”选项卡并在“规范”字段中键入JavaScript表达式。

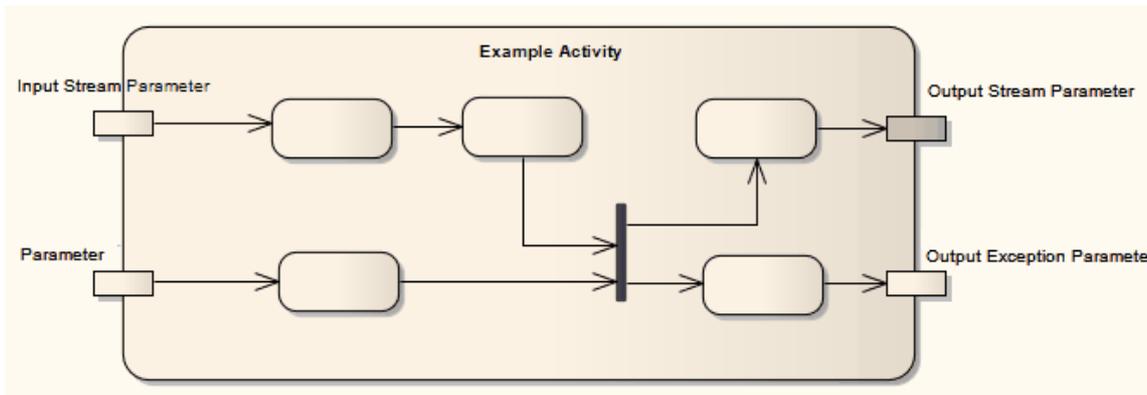
# 活动参数Nodes

## 描述

一个活动参数节点接受一个活动的输入或提供一个活动的输出。  
此示例描述了为活动定义的两个入口参数和一个输出参数。



## 为一个活动定义一个活动参数节点



节	行动
1	右键单击元素并选择 新元素 活动参数”选项。
2	将显示 属性”对话框，其中提示嵌入元素的名称和其他属性。
3	要进一步定义新的活动参数，请选择 属性”对话框的 参数”选项卡并定义： <ul style="list-style-type: none"> <li>• 类型</li> <li>• 默认值</li> <li>• 方向</li> <li>• 这是否是一个固定值</li> <li>• 多重性上限和下限</li> <li>• 是否允许重复和</li> <li>• 多重性是否有序</li> </ul>

活动参数节点也有 'Exception' 和 'Stream' 选项：

- 异常表示一个参数可以在排除其他输出时发出一个值，通常是因为一些错误
- Stream 指示参数在活动执行期间是否可以接受或发布值

## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.398) 指出：

作为一种行为，活动可能具有参数 (..) 。当活动被调用时，值可以通过活动活动（即具有方向 in 或返回）。

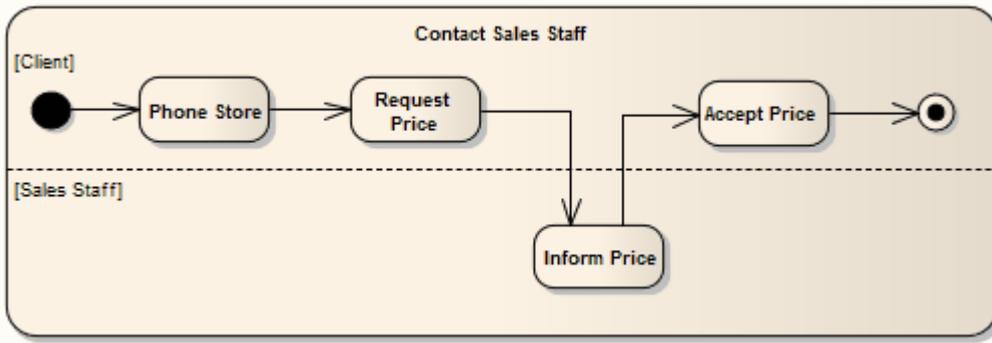
在活动中，活动的输入和输出使用 ActivityParameterNodes 处理。每个 ActivityParameterNode 都与拥有该节点的活动的一个参数相关联。ActivityParameterNode 的类型应与其关联参数的类型相同。

# 活动分区

Enterprise Architect支持两种类型的活动分区：

- 本主题中描述特征活动分区，用于在逻辑上组织活动元素
- 活动分区元素，用于按逻辑组织活动图

实际上，这些是相同的。它们在不影响令牌流的情况下划分活动的行动，有助于构建活动的视图或部分。此处显示了功能分区活动的示例：



## 定义分区

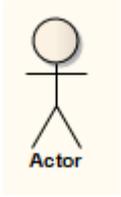
节
在图表中，右键单击活动元素并选择 高级 分区活动选项。 将显示 活动分区”对话框。
在 名称”字段中，输入分区的名称。 单击保存按钮。
对要创建的每个分区重复步骤 2。
单击关闭按钮。 如果分区没有显示在元素上，请单击元素右侧的  图标，以切换分区的显示。
单击分区边界并将它们拖动到适当的位置以包含适当的元素。

## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.408) 指出：

ActivityPartition 用两条通常平行的线（水平线或垂直线）和一个在一端的框中标记分区的名称来表示。放置在这些线之间的任何 ActivityNodes 和 ActivityEdges 都被认为包含在分区内。ActivityPartition 的这种表示法通俗地称为泳道，(…)

# 参与者



注记在构造图上，参与者元素默认为 Rectangular Notation 并且看起来像类元素。

## 描述

参与者是系统的用户；用户可以指人类用户、机器，甚至是模型中的另一个系统或子系统。任何从外部或系统边界与系统交互的东西都被称为参与者。Actor 通常与使用案例相关联。

参与者可以通过图形用户界面、批处理界面或其他媒体使用系统。参与者与用例的交互记录在用例中，其中详细说明了系统必须提供以满足用户要求的功能

Actors 还代表用户在序列图中的角色，您可以在其中使用矩形符号显示它们。Enterprise Architect支持用于业务建模的原型参与者元素。业务建模元素也将 Actor 表示为原型对象。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, pp.640/647) 指出：

参与者指定用户或与主题交互的任何其他系统所扮演的角色

其参与者的使用模型是由实体扮演的角色类型，该角色与案例的主体进行交互（例如，通过交换信号和数据）。参与者可以代表人类用户、外部硬件或其他系统所扮演的角色。

笔记。一个特定的参与者不一定代表相关的物理实体，而是与规范相关的用例相关的某个实体的角色。因此，单个物理实例可能扮演几个不同的 Actor 的角色，相反，一个给定的参与者可能由多个不同的实例扮演。

笔记。术语“角色”在这里被非正式地使用，并不暗示在本规范其他地方找到的该术语的任何技术定义。

当一个参与者有一个多城市的应用用例大于一个应用用例时，这意味着一个给定的参与者可以参与该类型的多个用户案例。这种多重参与的具体性质取决于手头的情况，在本规范中没有定义。因此，参与者可以并行（同时）或在不同时间点启动多个用例。

# Central Buffer节点



## 描述

Central Buffer节点是A object节点，用于管理来自多个源和目的地的流，在活动图中表示。它充当来自其他object节点的多个流入和流出的缓冲区，但不直接连接到行动。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.398) 指出：

A充当传入*ObjectFlow*和传出*ObjectFlow*之间的缓冲区。它接受在所有传入流上提供给它的所有object令牌，然后由节点持有。持有的object令牌根据*ObjectNode*的一般排序规则提供给传出流。当下游object节点接受令牌的提议时，该令牌将从*CentralBufferNode*中移除并移动到接受object节点，与任何object节点一样。

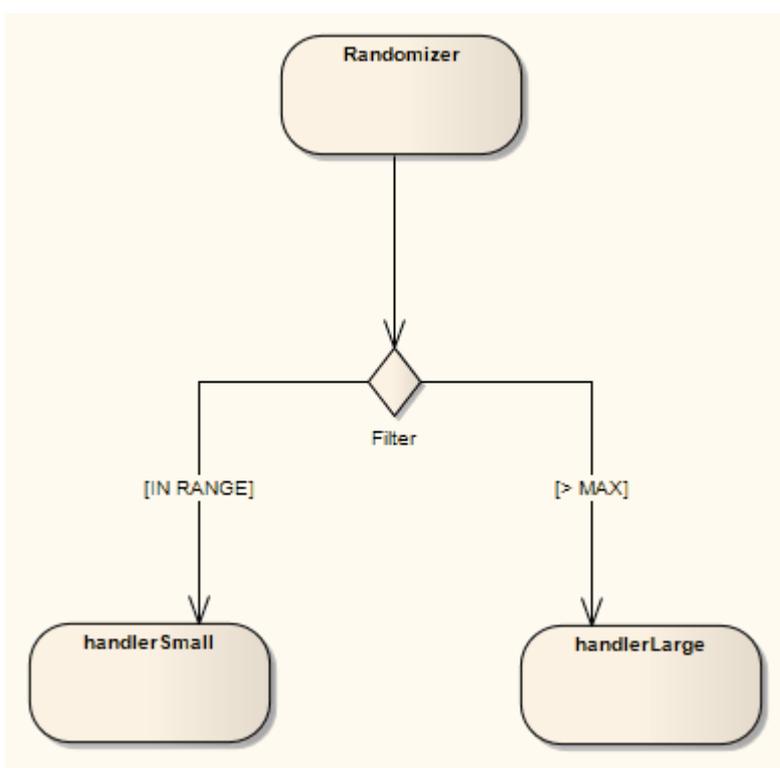
## 选择



### 描述

选择伪状态用于在例如状态机图中构成复杂的过渡路径，其中传出过渡路径由动态的运行时条件决定。运行时条件由状态机在导致选择的路径上执行的操作确定。

这个例子描述了选择元素。在达到过滤器伪状态时，根据传递给过滤器的运行时值触发到适当状态的转换。在形式上与连接点伪状态非常相似，选择伪状态的区别在于在运行时决定转换路径。



### 工具箱icon



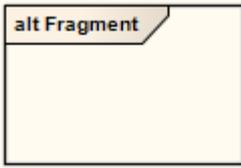
### OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.313) 指出：

这种类型的 Pseudostate 类似于 junction Pseudostate (...) 并用于类似的目的，不同之处在于，当复合转换遍历到达此 Pseudostate 时，动态评估所有传出转换的保护约束。因此，选择用于实现动态条件分支。它允许将复合转换拆分为多个替代路径，从而决定采用哪条路径可能取决于在到达选择点之前在同一复合转换中执行的行为执行

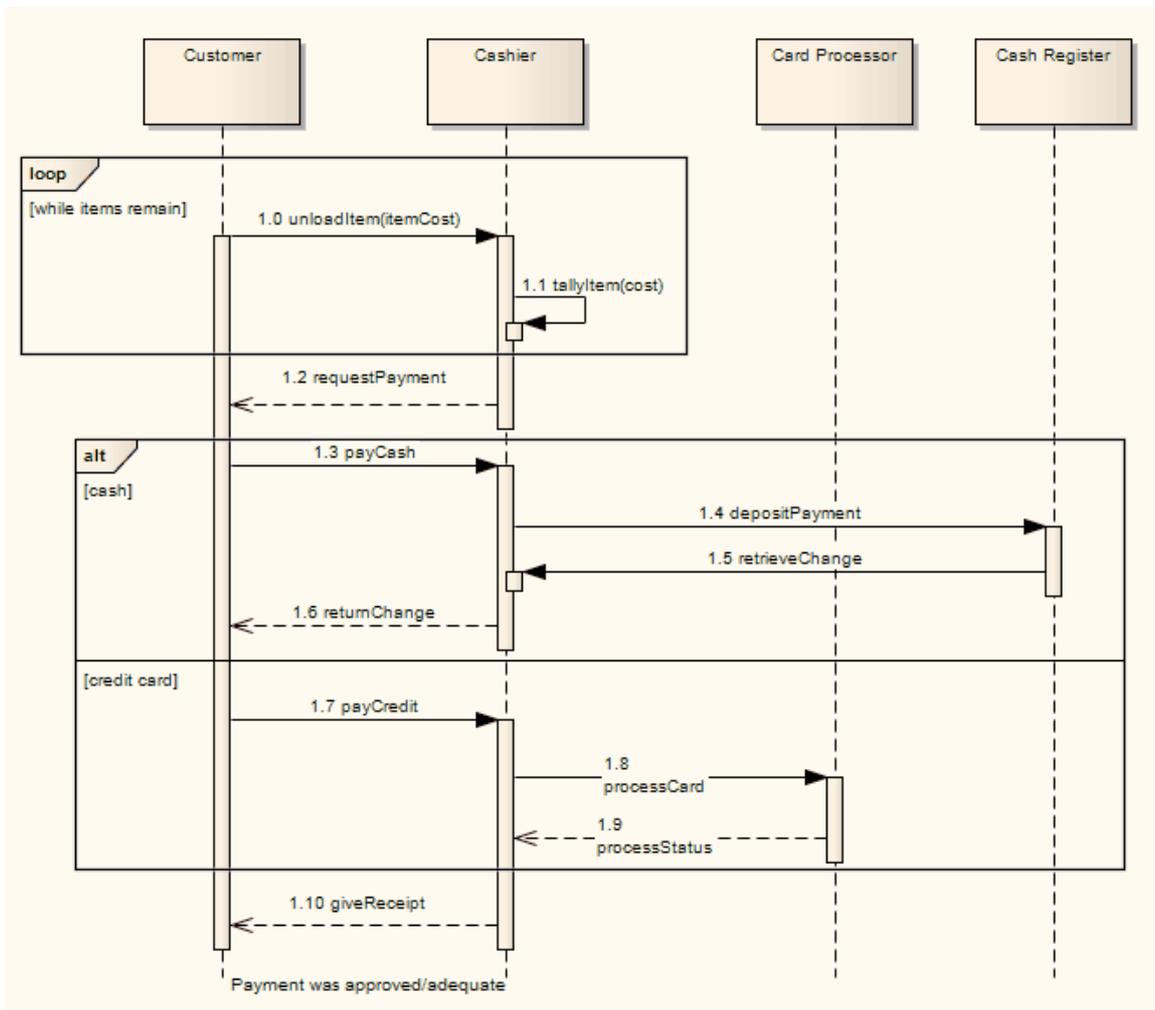
的结果。如果不止一个守卫评估为真，则选择相应的转换之一。未定义进行此选择的算法。如果没有一个守卫评估为真，则认为模型不正确。为避免这种情况，建议定义一个传出转移为每个选择 Pseudostate 使用预定义的 “else” 保护。

# 组合片段



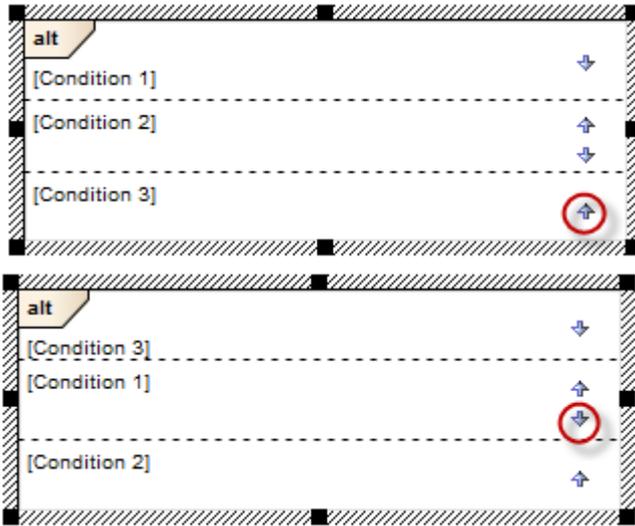
组合片段反映由交互运算符控制的交互A一个或多个方面（称为交互操作数），相应的布尔条件称为交互约束。Fragment 显示为一个透明窗口，每个操作数被水平线分隔。

此序列图说明了使用组合图对片段的购买流程建模。循环片段表示A未知数量的购买项目进行迭代，之后收银员请求付款。另一个片段表示支付选项，该片段被划分为显示两个操作数现金和信用卡。片段完成跟踪后，收银员在满足付款要求的情况下向客户提供收据。



交互片段条件的顺序可以直接在图上改变：

1. 选择一个定义了多个条件的交互片段；向上和向下箭头出现在每个条件的右侧。
2. 单击相应的箭头以更改顺序。

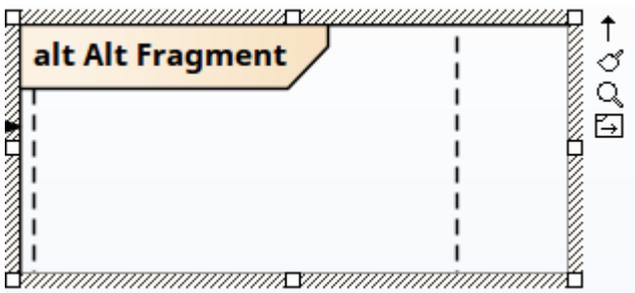


### 选择和移动组合片段

为了选择一个组合片段，您必须在内部边缘附近单击或在片段周围拖动一个选择矩形；这是为了防止在 Fragment 内移动消息时意外选择。

一旦包含在 Fragment 或 Fragment 操作数中，当消息在图表中上下移动时，它们将继续包含在其中。要将信息移出片段，或将信息移到片段内序列中的不同位置，请在将信息拖动到相应位置时按住 Alt 键。序列图上 A 片段将在其中的信息向上或向下移动时调整大小，以继续包含该信息。

当您选择元素图标时，图表上的交互会显示两个片段之一（位于右上角的控制您如何自由移动片段和片段内的任何消息的消息）和片段。



要独立于其内容移动组合片段，请确保“自由移动”元素图标  可见；如果未显示，请单击“移动内容”图标  并拖动元素边框。

交互在组合片段中的片段操作数不能被移动到片段之外，除非片段处于“自由移动”模式。移动操作数行会将该行下方的任何对象和消息向下或向上移动操作数行的移动量。

包含其他片段的片段在调整包含的片段大小时调整大小（除非片段处于“自由移动”模式）。

### 填充不透明度

虽然在交互片段中，有时会出现一些其他元素，但这些元素通常可能是隐藏和充分显示它们的原因，或者可能只是表明它们存在，这取决于图表的直接目的。您可以通过更改元素交互。

在设置不透明度之前，请检查元素是否具有填充颜色。

您可以使用以下两个弹出式元素工具栏中的图标设置不透明度：

- 点击交互元素和  图标：



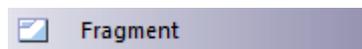
- 右键单击组件元素并在上下文菜单交互查看片段



单击  图标并选择：

- 100% 的总不透明度，其中隐藏的元素后面和重叠或仅被覆盖到交互的元素（您可以右键单击单个元素并选择 Z-Order | 带上“选项以公开这些元素）
- 0% 表示没有不透明度，没有应用填充颜色，并且填充后面的交互内容都完全可见
- 75%、50% 或 25% 以设置适当的不透明度并使被覆盖的元素可见但过度着色

## 工具箱icon



## OMG UML规范

OMG Unified Modeling Language规范 (v2.5. 1 , p.607) 指出：

A CombinedFragment 定义了一个 InteractionFragments 的表达式。A CombinedFragment 由一个交互操作符和对应的 InteractionOperands 定义。通过使用CombinedFragments，用户将能够以简洁明了的方式描述许多痕迹。

# 创建组合片段

## 创建组合片段

节	行动
1	将工具箱交互图表工具元素”页面上。
2	在 类型”字段中，单击下拉箭头并选择各种类型的交互操作符之一。
3	在 条件”字段中，为每个操作数指定条件或交互约束。
4	将显示A矩形框，由线条划分为每个操作数的段。
5	调整框架以包含每个操作数所需的事件发生。

## 注记

- A消息在其中移动时，消息将始终包含在片段或片段操作数中
- 当消息被向下移动时，图上的片段序列将调整大小，以确保消息一旦在片段中，始终保留在片段中

# 交互

创建组合片段时，您必须应用适当的交互操作符来表征片段。本表对各种操作符及其相应的描述提供了指导。

## 交互

操作符	行动
alt	根据布尔条件划分交互片段。
选择	附上一个可选的交互片段。
标准杆	指示操作数并行操作。
环形	指示操作数重复多次，由交互约束指定。
批判的	指示一个不能被其他处理中断的序列。
否定	断言片段无效，并暗示所有其他交互都是有效的。
断言	指定要出现的唯一有效片段。此运算符通常包含在考虑或忽略操作数中。
严格的	表示操作数的行为必须严格按序列处理。
序列	表明组合片段是弱排序的。这意味着操作数内的顺序保持不变，但操作数之间的顺序未定义，只要第一个操作数的事件发生在第二个操作数的事件发生之前，如果事件发生在同一生命线上。
忽视	指示在执行期间应该忽略哪些消息，或者可以出现在执行跟踪中的任何位置。
考虑	指定应在跟踪中考虑哪些消息。这通常用于使用断言运算符指定结果事件的发生。
参考	提供对另一个图表的参考。 引用片段不是使用创建组合片段主题中描述的方法创建的。要创建 ref 片段，只需将现有图表从浏览器窗口拖到当前图表上。

## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, pp.583-585) 指出：

*interactionOperator* 的值对于 *CombinedFragment* 的语义很重要，如下文针对每个 *interactionOperator* 枚举值指定的那样。

### 备择方案

*interactionOperator alt* 指定 *CombinedFragment* 代表行为的选择。最多选择一个操作数。选择的操作数必须有一个显式或隐式的保护表达式，在交互的这一点上计算结果为真。如果操作数没有保护，则隐含隐含的真实保护。定义选择的跟踪集是操作数的（受保护的）跟踪的并集。由 *else* 保护的 *alt* 操作数指定一个保护，它否定封闭的

CombinedFragment 中所有其他保护的析取。如果没有一个操作数具有评估为 true 的保护，则不会执行任何操作数，并且执行封闭的 InteractionFragment 的其余部分。如果在 alt CombinedFragment 的任何 InteractionOperand 中使用了内部 CombinedFragment 门，则该 alt CombinedFragment 的每个 InteractionOperand 必须使用具有相同名称的门。

### 选项

interactionOperator opt 指定 CombinedFragment 表示选择（唯一）操作数发生或什么都不发生的行为。一个选项在语义上等同于一个可选的 CombinedFragment，其中一个操作数具有非空内容，而第二个操作数为空。

### 休息

interactionOperator break 指定 CombinedFragment 代表一个中断场景，因为操作数是一个被执行的场景，而不是封闭的 InteractionFragment 的其余部分。当守卫闭合的其余部分为真时，选择带有交互 A 中断运算符。当 break 操作数的保护为 false 时，会忽略 break 操作数并选择其余的封闭 InteractionFragment。没有保护的中断操作数和其余的封闭 InteractionFragment 之间的选择是非确定性的。带有 interactionOperator break A CombinedFragment 应该覆盖封闭 InteractionFragment 的所有生命线。

### 平行

interactionOperator par 指定 CombinedFragment 表示操作数的行为之间的并行合并。不同操作数的 OccurrenceSpecifications 可以以任何方式交错，只要保留每个操作数强加的顺序。并行合并定义了 A 组跟踪，这些跟踪描述了操作数的 OccurrenceSpecifications 可以交错而不妨碍操作数内的 OccurrenceSpecifications 顺序的所有方式。

### 弱测序

interactionOperator seq 指定 CombinedFragment 表示操作数行为之间的弱排序。弱排序由具有以下属性的跟踪集定义：

1 每个操作数内的 OccurrenceSpecification 的顺序在结果中保持不变。

2 来自不同操作数的不同生命线上的 OccurrenceSpecifications 可能以任何顺序出现。

对来自不同操作数的同一生命线上的 3 个 OccurrenceSpecification 进行排序，以使第一个操作数的 OccurrenceSpecification 排在第二个操作数之前。

因此，当操作数在不相交的参与者集上时，弱排序会简化为并行合并。当操作数仅对一个参与者起作用时，弱排序会降低到严格排序。

### 严格测序

interactionOperator strict 指定 CombinedFragment 表示操作数行为之间的严格排序。严格排序的语义用 interactionOperator strict 定义了 CombinedFragment 内第一级操作数的严格排序。因此，包含的 CombinedFragment 中的 OccurrenceSpecifications 不会直接与封闭的 CombinedFragment 的其他 OccurrenceSpecifications 进行比较。

### 消极的

interactionOperator neg 指定 CombinedFragment 表示定义为无效的跟踪。定义带有 interactionOperator 负数的 CombinedFragment 的跟踪集等于其（唯一）操作数给出的跟踪集，只是该集是一组无效而不是有效的跟踪。所有与 Negative 不同的 InteractionFragment 都被认为是积极的，这意味着它们描述了有效且应该可能的跟踪。

### 关键区域

interactionOperator 关键指定 CombinedFragment 代表一个关键区域。临界区域意味着 A 区域的轨迹不能被其他 OccurrenceSpecifications 交错（在该区域覆盖的那些生命线上）。这意味着在确定有效跟踪集时，该区域由封闭片段原子地处理。即使封闭 CombinedFragments 可能意味着某些 OccurrenceSpecifications 可能会交织到区域中，例如使用 par-operator，但通过定义区域可以防止这种情况发生。因此，封闭构造的轨迹集受到关键区域的限制。

### 忽略/考虑

interactionOperator 忽略指定有一些消息类型未显示在此组合片段中。这些消息类型可以被认为是无关紧要的，如果它们出现在相应的执行中，它们就会被隐式忽略。或者，可以将忽略理解为意味着被忽略的消息类型可以出现在跟踪中的任何位置。相反，interactionOperator 考虑指定在这个组合片段中应该考虑哪些消息。这等效于定义要忽略的所有其他消息。

### 断言

`interactionOperator` 断言指定 `CombinedFragment` 表示一个断言。断言操作数的序列是唯一有效的延续。所有其他继续都会导致无效的跟踪。断言通常与忽略或考虑结合使用，如图 17.17 所示。

### 环形

`interactionOperator` 循环指定 `CombinedFragment` 代表一个循环。循环操作数将重复多次。

守卫条件可以包括循环的下限和上限迭代次数以及布尔表达式。语义是这样的，循环将迭代最少的 `minint` 次数（由保护中的迭代表达式给出），最多迭代 `maxint` 的次数。在执行了最 次数的迭代并且布尔表达式为 `false` 后，循环将终止。循环构造表示 `seq` 运算符的递归应用，其中循环操作数在较早迭代的结果之后排序。

如果循环包含带有规范的单独 `InteractionConstraint`，则只有在该规范在执行期间评估为 `true` 时，循环才会继续，而不管循环中指定的最小迭代次数如何。

# 约束

```
{Constraint: Legal : The attached  
Requirements are subject to legal  
verification.}
```

## 描述

约束元素表示对其他元素A约束；它可以连接到任何类型的其他元素。通过工具箱的“公共”页面，可以在任何Enterprise Architect图表中找到约束元素图标。

该元素是一种记录相关元素存在约束这一事实的方法；它对其他元素没有影响。但是，约束是一个命名元素，列在浏览器窗口中，并且可以在适当的情况下复制和重复使用。您在项目参考数据中定义约束类型，将它们应用于元素“属性”对话框中的元素，并通过“责任”窗口管理它们。

## 工具箱icon

 Constraint

# 数据存储



## 描述

数据存储是用于定义永久存储数据A元素。进入数据存储A数据令牌被永久存储，更新已存在数据的令牌。A数据存储的数据令牌是原始数据的副本。

使用物件流连接器将元素（例如活动）连接到数据存储，因为值和正在节点之间传递。选择和转换行为共同构成一种查询，可以根据数据访问的性质进行指定。例如，选择行为确定哪些对象受到与数据存储的连接的影响。变换行为可能会进一步指定与所选object有关的属性的值。

要定义对数据存储的访问行为，请在注记物件流连接器上附加注释。为此，请右键单击物件流并选择“附加注记或约束”选项。A对话框指示活动图表中的其他流，您可以在其上附加注记（如果该行为适用于多个流）。为了符合UML 2.x，在行为前加上符号 «selection» 或 «transformation»。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.399) 指出：

*DataStoreNode*是A *CentralBufferNode*，它在其活动执行时永久保存其object令牌。当下游object节点接受对*DataStoreNode*持有的object令牌的提议时，根据通常的*CentralBufferNode*语义，从*DataStoreNode*中删除所提供的令牌。但是，已删除object令牌的副本具有相同的值，并立即放回*DataStoreNode*。因此，*DataStoreNode*持有的值似乎在其包含的活动的每次执行期间都保持不变，即使令牌从节点向下游移动也是如此。当*DataStoreNode*接受object令牌时，如果该令牌包含与该节点已持有的令牌中包含的object具有相同身份的object，则不应将重复的object令牌放置在*DataStoreNode*上。与常规*CentralBufferNode*不同，*DataStoreNode*包含唯一的对象。

# 决策

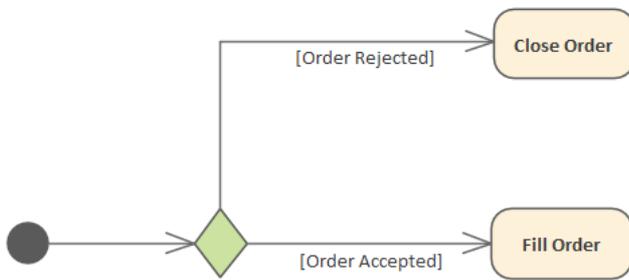


## 描述

A活动决策是一个元素的一个点或一个点：如果一个条件交互的条件为True，则处理继续一个方式如果没有，那么另一个。

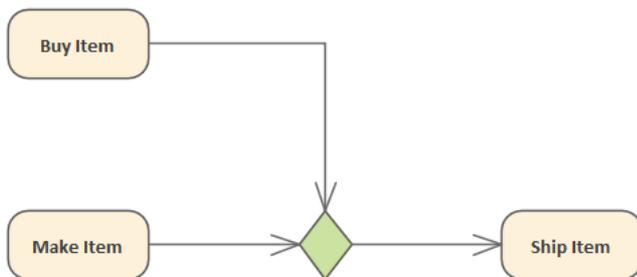
元素也可以用作合并节点，因为可以合并（但不同步）多个替代流以形成一个流。这些示例显示了使用决策元素的这两种方法。

用作决定：



请参阅 [OMG Unified Modeling Language规范](#)，（ 1，图 12.77，第 363 页。）

用作合并：



请参阅 [OMG Unified Modeling Language规范](#)，（ v2.5.1，图1，第 388 页。）

您可以在属性窗口中选择一个行为元素作为UML决策决策属性。要在图表上显示选定的决策输入属性，请在决策上附加注记，然后右键单击注释决策注记连接并选择“将此注记链接到元素特征”选项。然后选择“决策输入”作为链接特征。

您还可以选择一个物件流作为决策的决策输入流（UML：决策）。选择传入的物件流并在属性窗口中选择“决策输入流”选项。

## 注记

- 移动图一般不会影响包内元素的位置；如果将图表从一个包中移到另一个包中，则图表中的所有元素都保留在原始包中

但是，决策元素仅在一个图表中使用，在该图表之外没有任何意义，并且永远不会在任何其他图表中重复使用；因此，如果您移动包含这些元素的图表，它们将与图表一起移动到新的父包

## 工具箱icon



## OMG UML规范：

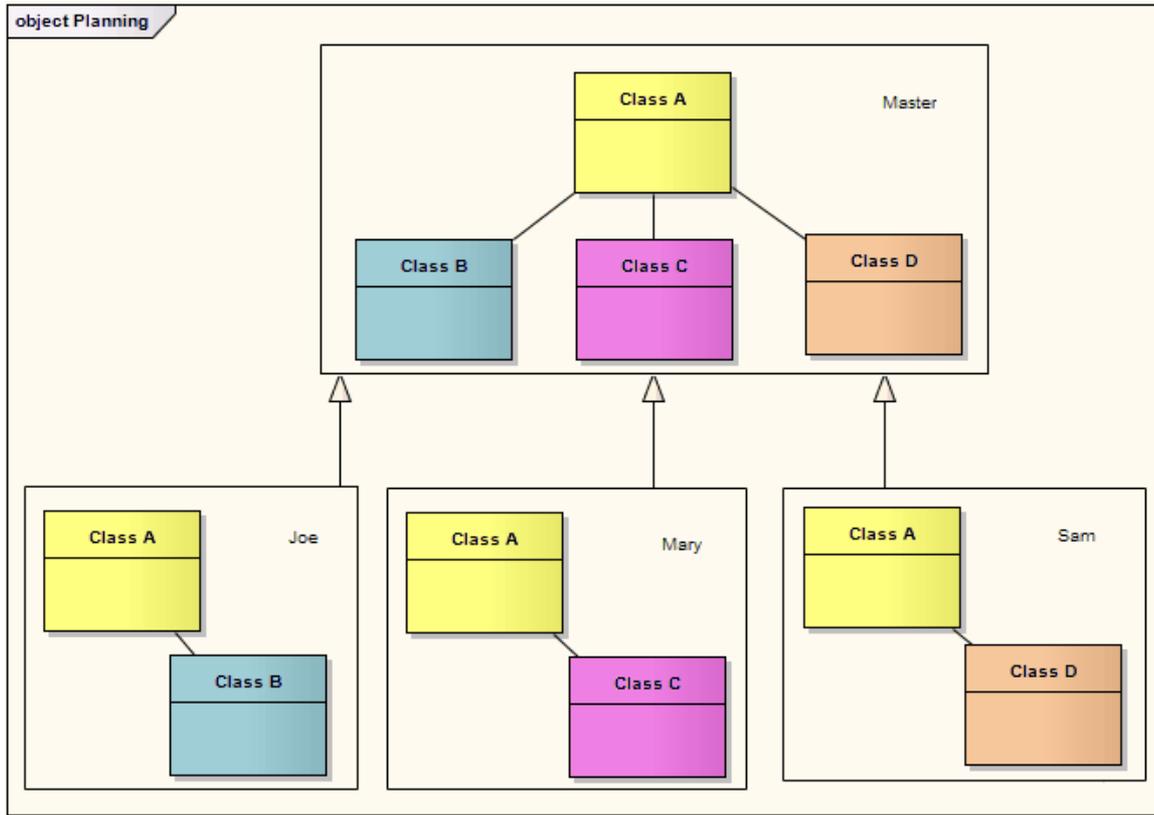
OMG Unified Modeling Language规范 (v2.5.1, 决策(决策节点)) 指出：

DecisionNode 是在传出流之间进行选择A ControlNode。A DecisionNode 应具有至少一个且最多两个传入的 ActivityEdge，以及至少一个传出的 ActivityEdge。如果它有两个输入边，那么一个被识别为decisionInputFlow，另一个被称为主输入边。如果DecisionNode 只有一个入边，那么它就是主入边。如果 DecisionNode 的主要输入边是 ControlFlow，那么所有输出边都应该是 ControlFlows，如果主要输入边是 ObjectFlow，那么所有输出边都应该是 ObjectFlows。

DecisionNode A其主要传入边缘上接受令牌并将它们提供给其所有传出边缘。但是，在主要传入边缘上提供的每个令牌应最多遍历一个传出边缘。令牌不重复。

如果 DecisionNode 的任何传出边具有保护，则为每个传入令牌评估这些。评估警卫的顺序没有定义，可以同时评估。如果DecisionNode 的主要传入边是ObjectFlow，并且DecisionNode 没有decisionInput 或 decisionInputFlow，则传入object令牌中包含的值可用于评估传出ObjectFlows 上的保护。

# 图框



图框元素是从浏览器窗口拖放到另一个图表的图表A再现。它是一种带有“交互符”参考的组合片段。但是，它可以在任何类型的图表上创建，并且与其他组合片段的创建方式不同。

当您将图表从浏览器窗口拖放到打开的图表上时，会显示一个对话框，提供（以及其他）以下选项：

- '图框' - 一个图框被插入到图表中，包含被丢弃的图表的图像
- '图表参考' - 在框架标签中插入一个空框架，其中包含已删除图表的名称

在这两种情况下，object都充当指向实际引用图的超链接。您还可以通过右键单击object并选择元素“属性”选项来定义对象的属性，就像其他元素一样。

'类型'对话框中的所有选项都在Add图表to图表帮助主题中讨论。

## 图框外观

与其他元素一样，您可以更改图框的外观，但可用选项是针对此元素类型量身定制的。如果您右键单击框架并选择“外观|图框外观”选项，则会显示一个包含以下选项的子菜单：

- 'Normal' - 带有可见框架标签的可见矩形框架的默认外观；您可以在使用其他选项之一后使用此选项重置外观
- '边界' - 隐藏图框框的框架标签
- '带名称的边界' - 隐藏框架标签的边框
- '名称Only' - 隐藏图框和框架标签的边框，只留下文本
- 'Hidden' - 隐藏图框的边框和文字

在 SysML、状态或状态机图中：

- 如果框架设置为不可选择，它将自动调整大小以适应图表的边界，从其默认大小扩展但不会缩小
- 图表显示使用Enterprise Architect 14.0 或更高版本应用的图表如果使用早于 14.0 版本的Enterprise Architect打

开，则在图表上绘制父object

## 通过图表移动元素

作为图表参考图表A有用的特征是它们提供了将当前显示在主机图表上的元素通过框架移动到被引用图表、被引用图表的父包或两者兼而有之的功能。您还可以将元素添加到引用的图表中，作为指向其当前位置的链接。

要移动或链接元素，只需将其拖到当前图表上的图框。将显示A对话框，列出以下选项：

- 将元素移动到引用的图表
- 在引用的图表上创建指向元素的链接
- 将元素移动到父包中
- 将元素移动到引用的图表及其父包中

如果您选择其中一个选项将元素移动到引用的图表，则它和它所具有的任何连接器都将从当前图表中删除并放置在引用图表的空白区域中。如果元素已经与图表上的其他元素有关系，则包括这些关系。

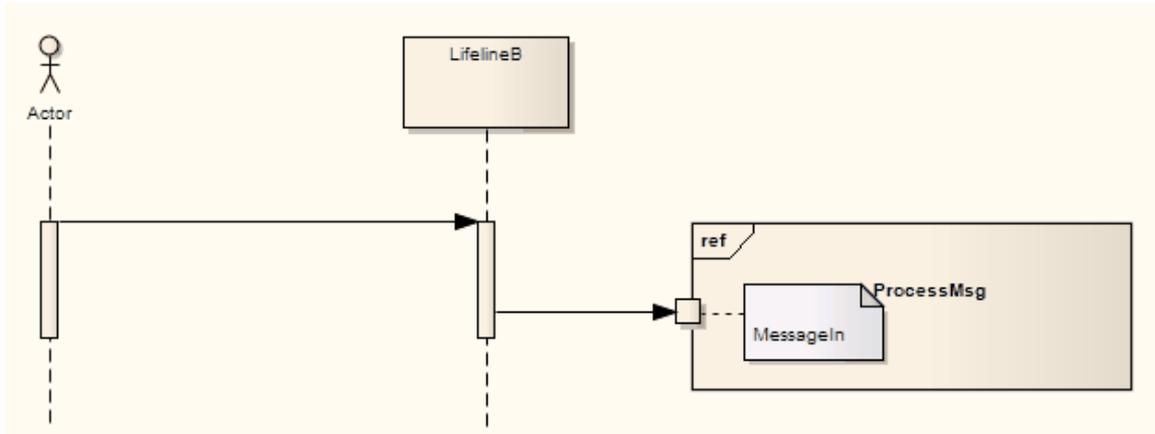
## 注记

- 您可以更改两个对象的大小，但不能将图框到小于所附图表的大小
- 您不能在图框更改图表；要编辑图表，请在框架内双击并编辑原始图表
- 图框元素与您可以在打印输出的图表图像上设置（使用“首选项”对话框的“图表”页面上的“图表框架”面板）的图表框架边框不同，复制到文件或粘贴到其他工具中；可以（但不常见）将剪贴板中的图表图像粘贴到另一个Enterprise Architect图表中，在这种情况下，图像最初看起来与图框元素相同，但元素选项在此图像上函数



### 描述

图表门是一个简单的图形方式，用于指示可以将消息传输进和传输出交互片段的点。可能需要一个片段来接收或传递消息；内部；一个有序的消息反映了这一要求，在片段帧的边界上标有一个门。任何与此内部消息“同步”的外部消息都必须适当地对应。Gates 可以出现在时序图上，或序列片段（或通讯上），交互交互组合交互。



### 工具箱icon



### OMG UML规范：

OMG Unified Modeling Language规范 ( 1 , pp.575-576 ) 指出：

门是一个 MessageEnd，它被用于在一个交互、一个 InteractionUse 或一个 CombinedFragment 的边界上，用于为每个信息建立具体的发送者和接收者。

门实例，因为它们以成对的方式出现，链接两个信息实例，它们本身也没有明确排序。Gates 是 MessageEnds，它在以下两者之间提供连接点：

- InteractionUse 之外的信息实例和使用的交互中的信息实例，或
- CombinedFragment 外的信息实例和 CombinedFragment 内的 InteractionOperand 内的信息实例。

# 端点

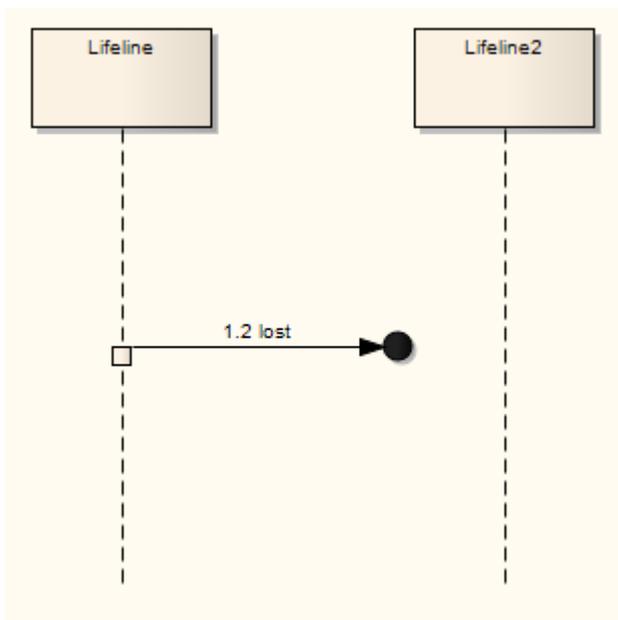


## 描述

交互序列用于在交互图、通讯地址或发现信息中序列模型，将端点元素拖到工作区上。

使用序列图，将信息从相应的生命线EndPoint。使用时序图，将生命线连接到端点的信息需要一些时序规范来绘制连接。

此示例在序列图中描述了丢失的信息。



## 工具箱icon



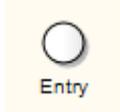
## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.574) 指出：

丢失信息信息A发送事件发生但没有接收事件发生的信息。我们将此解释为因为 [丢失]信息的目的地超出了描述的范围。

A的信息是已知接收事件发生但没有（已知）发送事件发生的信息。我们认为这是因为信息的来源超出了描述的范围。例如，这可能是我们不想在细节中描述的噪音或其他活动。

# 入口



## 描述

入口伪状态用于定义状态机的开始。每个区域都存在一个入口，指导初始并发状态配置。

## 工具箱icon



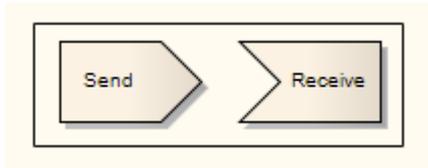
## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.313) 指出：

*entryPoint* Pseudostate 表示状态机的入口点或提供状态或状态机内部封装的复合状态。在拥有入口点的状态机或复合状态的每个区域中，最多有一个转移

从入口点到该区域内的区域。

# 事件



## 描述

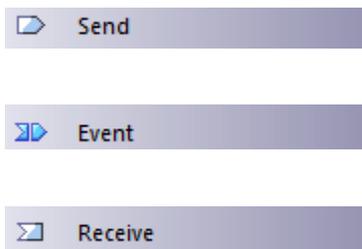
两个要素用于模型事件；这：

- 发送事件，它模拟系统中刺激的生成并将该刺激传递给系统内部或系统外部的其他元素
- 接收事件，被描绘为一个矩形，左侧有一个凹陷的“V”，这表明由于一些外部或内部刺激而在系统中发生了一个事件；通常这会调用进一步的活动和处理

可以从工具箱的分析、状态和活动元素页面添加发送和接收事件。

如果您应该选择错误的事件类型，或者想要更改类型，请右键单击事件并选择 高级|“使发件人'或'高级|酌情选择 Receiver' 选项。

## 工具箱icon



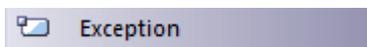
# 例外



## 描述

异常处理程序元素定义了发生异常时要执行的操作组。在活动图中，受保护的元素可以包含一组操作，并通过中断流连接器连接到异常处理程序。元素部分中包含的任何已定义错误都可以触发流程移动到异常。

## 工具箱icon



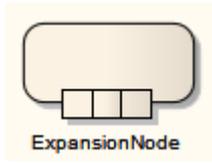
## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.404) 指出：

异常是用于标识执行的非正常完成模式的值。如果在 ExecutableNode 的执行中引发了异常（例如，使用 `RaiseExceptionAction ; ...`）并且没有在该执行中处理，则终止执行并将异常传播到 ExecutableNode 之外。

一个 ExecutableNode 可能有一个或多个 ExceptionHandlers 用于处理可能从 ExecutableNode 传播出来的异常，ExecutableNode 是那些处理程序的 protectedNode。如果异常从受保护节点传播出去，则检查处理程序集以查找与该异常匹配的处理程序。A异常的类型与处理程序的异常类型之一相同或（直接或间接）子类型，则处理程序匹配。如果匹配，则处理程序捕获

# 膨胀节点

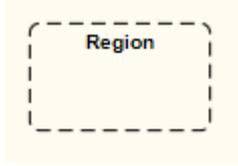


## 描述

将行动或活动表示为扩展节点是一种简写符号，表示行动/活动由扩展区域组成。

要将行动或活动指定为扩展节点，请右键单击行动并选择 新子元素|膨胀节点的选项。

## 拓展区域

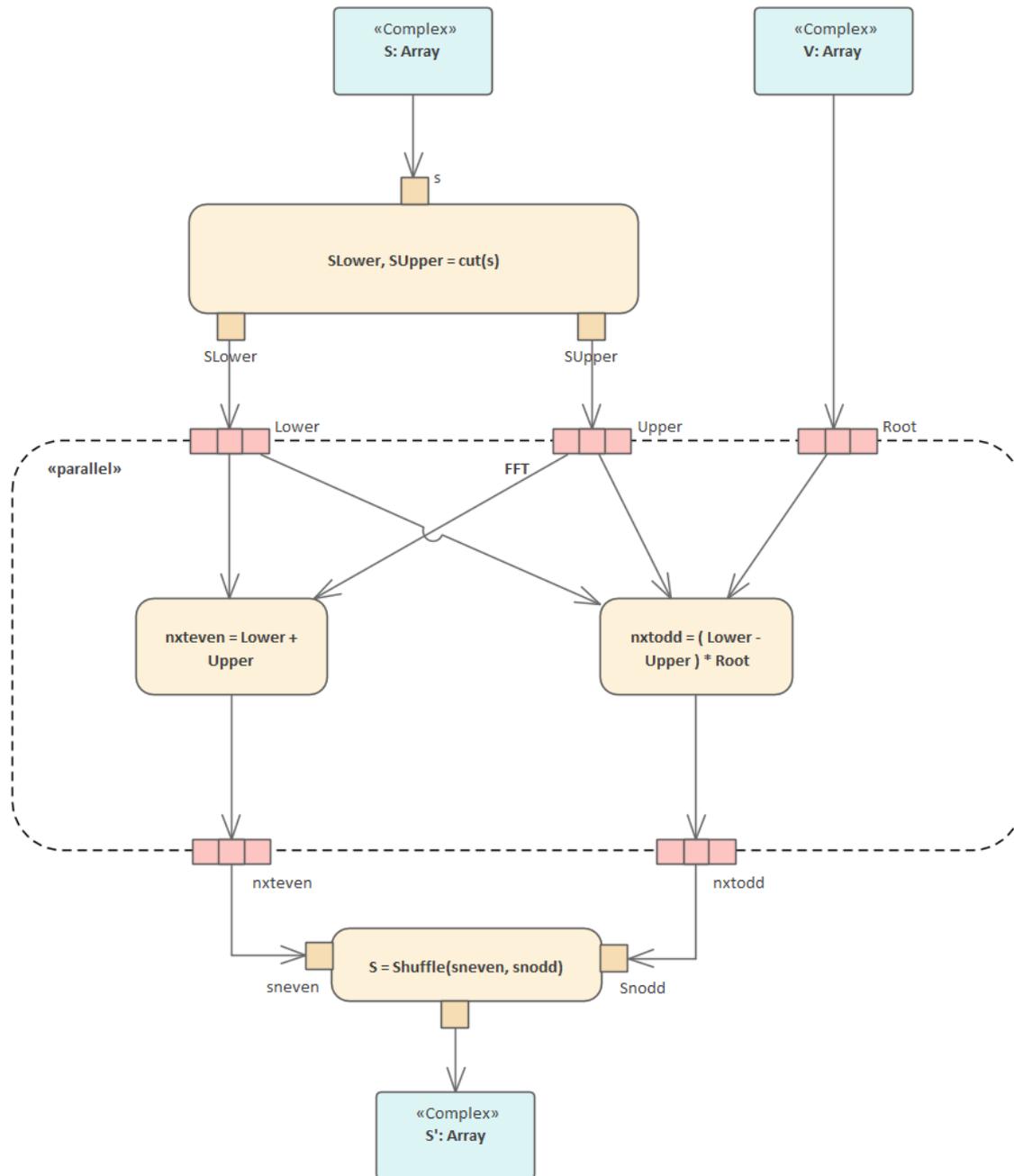


### 描述

在活动图中，Expansion区域包含一组 ActivityNode 和 ActivityEdge。这些 ActivityNode 和 ActivityEdge 将对传入数据执行多次，对输入集中的每个元素执行一次。如果有多个输入，则集合大小应匹配；如果他们不这样做，则最小的集合决定执行次数。集合也必须是相同的类型（例如 set 或 bag）。任何输出必须是至少与输入集合大小相同的集合形式；如果每次执行可以产生多个输出，则输出集合可以更大。

您可以使用工具箱的活动页面创建一个扩展区域作为区域的一种图表区域（另一个是可中断的活动区域）。系统会提示您指定扩展区域的多次执行（并行、迭代或流）的并发性。并行反映传入集中的元素可以同时处理或重叠处理，而迭代并发模式指定执行必须顺序发生。流模式Expansion区域表示输入和输出A流的形式进出，Expansion区域的流程必须有某种方法来支持流。

要修改扩展的区域，请右键单击它并选择“属性|特殊行动”选项，然后选择“高级”选项卡并单击“模式”字段中的下拉箭头。



### 工具箱icon

-  Expansion Region
-  Interruptible Region

### OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, pp.480-481) 指出：

*ExpansionRegion*是一个 StructuredActivityNode，它多次执行其包含的元素，对应于输入集合的元素。

*ExpansionRegion* 是一个 *StructuredActivityNode*，它将一个或多个值集合作为输入，并在这些集合中的每个值上执行其包含的 *ActivityNodes* 和 *ActivityEdges*。如果计算产生结果，则可以将这些结果收集到输出集合中。输出集合的数量可以不同于输入集合的数量。

# 出口



## 描述

状态机元素和状态机图中使用退出点来表示机器退出的点，并且触发了针对状态机元素的该退出点的转换。出口点是状态机图中使用的一种伪状态。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.313) 指出：

*exitPoint* Pseudostate 是状态机或复合状态的退出点，它提供状态或状态机内部的封装。过渡终止于复合状态的任何区域内的退出点或由子状态机引用的状态状态意味着退出此复合状态或子机状态（执行其相关的退出行为）。如果状态内正交区域的多个转换在此伪状态上终止，则它的作用类似于连接伪状态。

# 终点

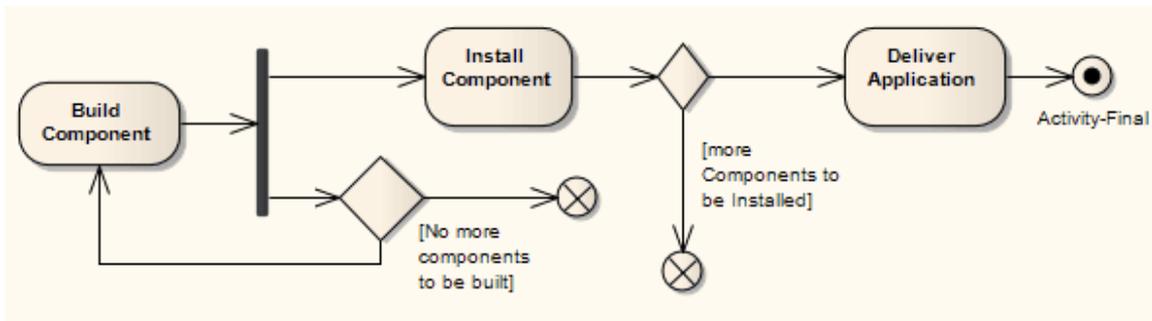


## 描述

在一个活动中定义两个终端UML 2.1都可以定义为终点状态终点节点。活动终点元素表示活动的完成；到达活动终点后，图中的所有执行都被中止。最终节点的类型，即来自其他执行系统的其他执行系统的流终点，描述了对活动中的流没有影响的出口。

下一个示例说明了应用程序的开发。当没有更多的构件需要建造时，该过程到达一个流终点节点；注记分叉元素表示与新构件的建造和已完成构件的安装并行的过程。流终点终端仅终止子过程构建组件。类似地，只有那些进入决策分支的令牌会随着连接更多组件的安装而流终点（即停止安装其他组件，但继续安装其他组件）。只有在交付应用程序活动完成后，控制流到达终点节点后，所有流才会停止。

启动流的节点是初始节点。



请参阅 OMG Unified Modeling Language规范，（ 1，图 12.91，p.374。）

## 注记

- 移动图一般不会影响包内元素的位置；如果将图表从一个包中移到另一个包中，则图表中的所有元素都保留在原始包中

但是，终点元素只在一个图表内使用，在该图表中没有任何意义，并且永远不会在任何其他图表中重复使用；因此，如果您移动包含这些元素的图表，它们将与图表一起移动到新的父包

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.388) 指出：

A是活动中的流停止的 ControlNode。A不应有传出的 ActivityEdges。A接受在其传入的 ActivityEdges 上提供给它的所有令牌。

FinalNode有两种：

1 A是终止流的 FinalNode。FlowFinalNode 接受的所有令牌都被销毁。这对活动中的其他流程没有影响。

2 ActivityFinalNode 是在活动(...) 中停止所有流的 FinalNode。到达某个活动所拥有的 ActivityFinalNode A令牌会终止该活动的执行。如果一个活动拥有多个 ActivityFinalNode，那么第一个接受令牌（如果有）的活动将终止该活动的执行，包括任何其他 ActivityFinalNode 的执行。

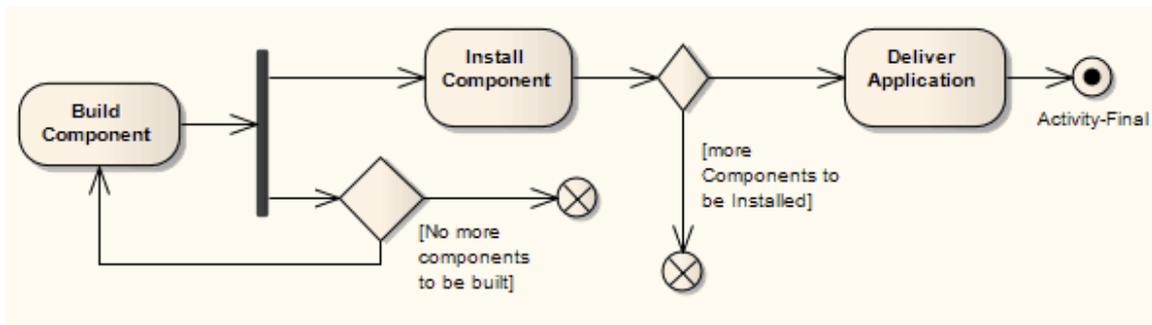
# 流终点



## 描述

在UML中定义的活动中有用于定义最终状态1为类型 2 的终点节点，两个节点。活动流终点元素表示系统的退出，与活动终点，表示系统的完成。只有进入流终点节点的流量退出活动；其他流动继续不受干扰。

此示例活动图说明了应用程序的开发。当没有更多的构件需要建造时，该过程到达一个流终点节点；注记分叉元素表示与新构件的建造和已完成构件的安装并行的过程。流终点终端仅终止子过程构建组件。类似地，只有那些进入决策分支的令牌会随着连接更多组件的安装而流终点（即停止安装其他组件，但继续安装其他组件）。只有在交付应用程序活动完成后，控制流到达终点节点后，所有流才会停止。



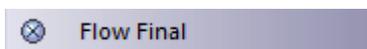
请参阅 [OMG Unified Modeling Language规范](#)，（ 1 · 图 12.91 · p.374 。 ）

## 注记

- 移动图表一般不会影响包中元素的位置：如果将图表从一个包中移到另一个包中，则图表中的所有元素都保留在原始包中

但是，流终点元素仅在一个图表内使用，在该图表之外没有任何意义，并且永远不会在任何其他图表中重复使用；因此，如果您移动包含这些元素的图表，它们将与图表一起移动到新的父包

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.388) 指出：

A是终止流的*FinalNode*。FlowFinalNode 接受的所有令牌都被销毁。这对活动中的其他流程没有影响。

# 分叉/汇合



分叉/汇合元素可用于：

- 分叉或将流拆分成多个并发流
- 汇合多个并发流的流
- 将多个传入流加入和拆分为多个传出流

这些元素用于活动和状态机图中，无论是垂直方向还是水平方向。对于状态机图，分叉和汇合作为伪状态。其他伪状态包括History其它、Entry Points和Exit Points。分叉用于将传入的转换拆分为并发的多个转换，导致不同的目标状态。汇合用于将并发的多个转换合并为一个导致单个目标的转换。它们是语义倒数。要了解有关这些单独元素的更多信息，请参阅它们的特定主题。

## 示例图表

描述	图表
分叉或将流拆分成多个并发流	
汇合多个并发流的流	
汇合和分叉	

## 工具箱icon



或者



## OMG UML规范：

分叉活动中的图表

OMG Unified Modeling Language规范 (v2.5. 1 , p.388) 指出：

分叉顶点用于将传入的过渡分割成两个或多个过渡，终止于正交目标顶点（即复合状态的不同区域中的顶点）。从 fork 顶点传出的段不能有防护或触发器。

分叉in状态机图表

OMG Unified Modeling Language规范 (v2.5. 1 , p.313) 指出：

( A ) *fork* 伪状态用于拆分传入的转移

成两个或多个过渡，终止于复合状态区域的正交区域中的顶点。从 fork Pseudostate 传出的 Transitions 不能有警卫或触发器。

汇合in活动图表

OMG Unified Modeling Language规范 (v2.5. 1 , p.389) 指出：

A是同步多个流的 ControlNode。A JoinNode 应该只有一个传出的 ActivityEdge，但可能有多个传入的 ActivityEdge。如果 JoinNode 的任何传入边是 ObjectFlow，则传出边应是 ObjectFlow。否则，出边将是一个控制流。

汇合在状态机图表

OMG Unified Modeling Language规范 (v2.5. 1 , p.313) 指出：

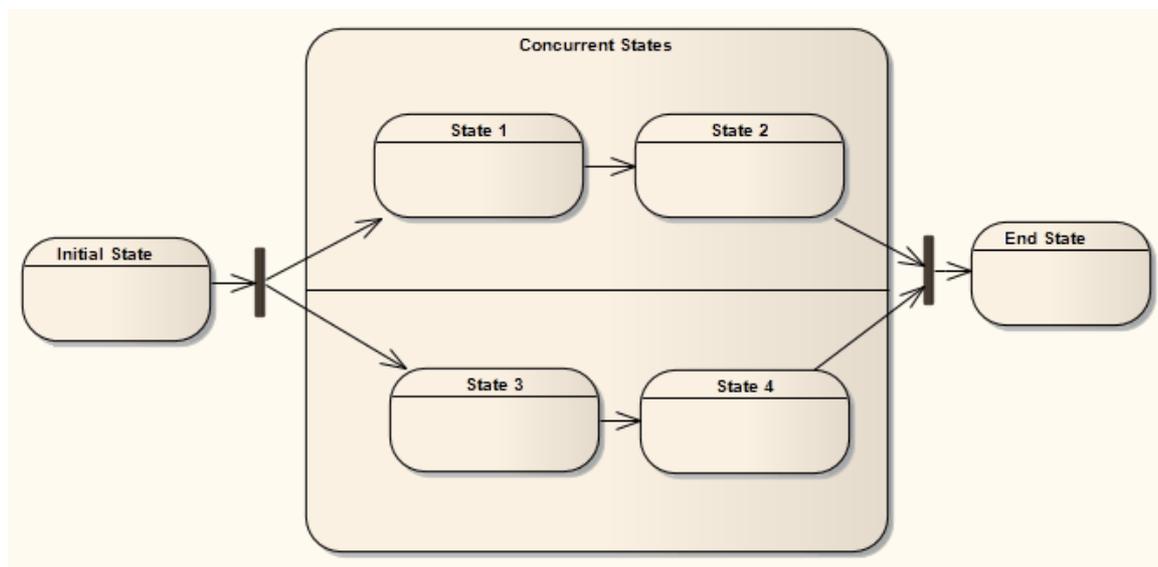
( A汇合 ) Pseudostate 作为来自不同正交区域中的Vertices 的两个或多个Transitions 的公共目标Vertex。在加入伪状态时终止的转换不能有警卫或触发器。与 Petri 网中的连接点类似，join 伪状态执行同步函数，其中所有传入的转换必须完成才能继续执行传出的转移

# 分叉

## 描述



分叉元素用于活动和状态机图中。就状态机图而言，分叉伪状态表示它的传入转换来自一个状态，并且它有多  
 个传出转换。这些转换必须同时发生，需要使用并发区域，如复合状态中所述。与选择或连接点伪状态不同，  
 分叉不能有触发器或守卫。该图演示了一个分叉伪状态分为两个并发区域，然后通过汇合伪状态返回到结束状  
 态。



## OMG UML规范：

分叉活动中的图表

OMG Unified Modeling Language规范 (v2.5. 1 , p.388) 指出：

分叉顶点用于将传入的过渡分割成两个或多个过渡，终止于正交目标顶点（即复合状态的不同区域中的顶点）  
 。从 fork 顶点传出的段不能有防护或触发器。

分叉in状态机图表

OMG Unified Modeling Language规范 (v2.5. 1 , p.313) 指出：

( A ) fork伪状态用于拆分传入的转移

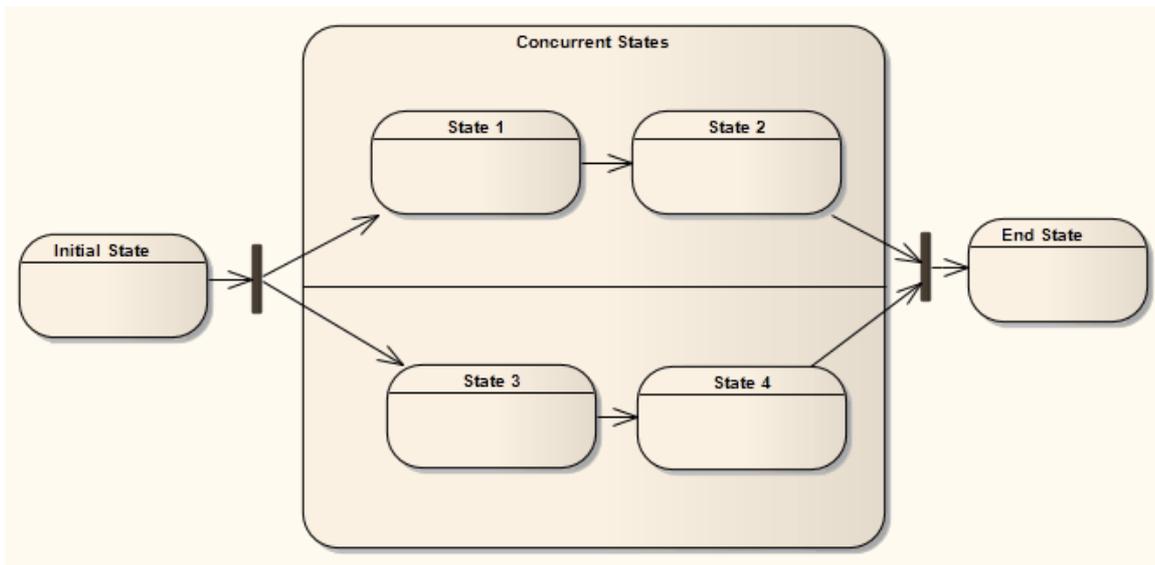
成两个或多个过渡，终止于复合状态区域的正交区域中的顶点。从 fork Pseudostate 传出的 Transitions 不能有警  
 卫或触发器。

# 汇合

## 描述



汇合元素用于活动和状态机图。该示例说明了活动之间的汇合转换。对于状态机图，汇合伪状态表示多个状态同时过渡到汇合并转变为单一状态。与选择或连接点伪状态不同，汇合不能有触发器或守卫。该图演示了一个分叉伪状态分为两个并发区域，然后通过汇合返回结束状态。



## OMG UML规范：

### 汇合in活动图表

OMG Unified Modeling Language规范 (v2.5. 1 , p.389) 指出：

A是同步多个流的 ControlNode。A JoinNode 应该只有一个传出的 ActivityEdge，但可能有多个传入的 ActivityEdge。如果 JoinNode 的任何传入边是 ObjectFlow，则传出边应是 ObjectFlow。否则，出边将是一个控制流。

### 汇合在状态机图表

OMG Unified Modeling Language规范 (v2.5. 1 , p.313) 指出：

( A汇合) Pseudostate 作为来自不同正交区域中的Vertices 的两个或多个Transitions 的公共目标Vertex。在加入伪状态时终止的转换不能有警卫或触发器。与 Petri 网中的连接点类似，join 伪状态执行同步函数，其中所有传入的转换必须完成才能继续执行传出的转移

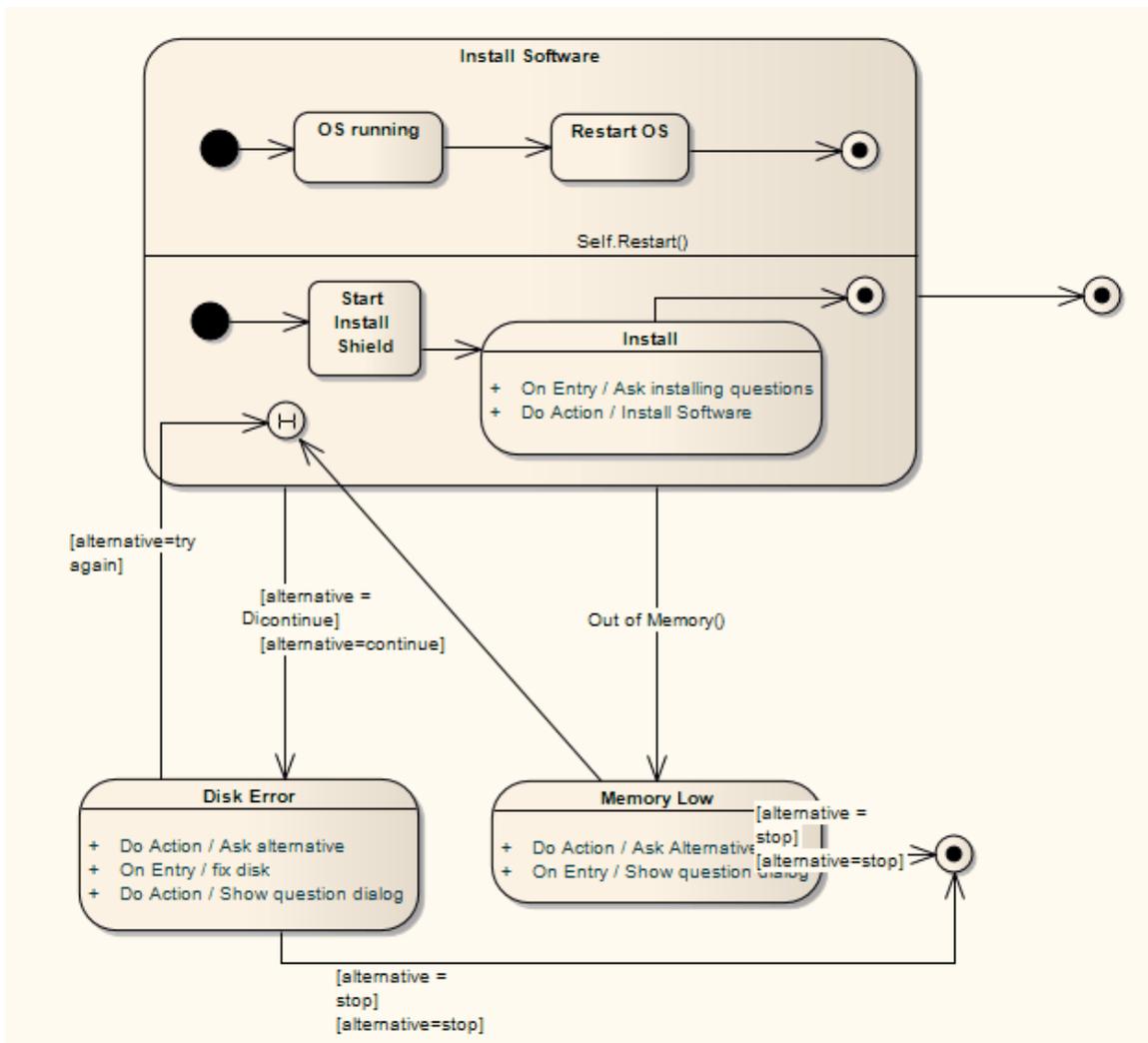
# 历史



## 描述

UML中定义了两种历史伪状态：浅历史和深历史。A历史子状态用于表示复合状态最近活跃的子状态；如果存在，则此伪状态不会递归到此子状态的活动配置中。可以A单个连接器来描述默认的浅历史状态，以防从未输入过复合状态。

相比之下，深层历史A状态反映了复合状态的最新活动配置。这包括所有区域的活动子状态，并递归到这些子状态的子状态（如果它们存在）。复合状态中只能存在一个深历史和一个浅历史。您可以使用“高级”元素上下文菜单将浅历史子状态重新分配为深层历史子状态。



## 工具箱icon

## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.309) 指出：

状态历史的概念是由大卫哈雷尔在最初的状态图形式主义中引入的。这是一个与复合状态区域相关联的便利概念，其中区域跟踪它上次退出时所处的状态配置。这允许轻松返回到相同的状态配置，如果需要，下一次区域变为活动状态（例如，在从处理中断返回之后），或者如果存在本地转移

回到它的历史。

提供了两种类型的历史伪状态。

深度历史（区域）代表最近访问包含区域的完整状态配置。效果和转移一样转移

终止于状态Pseudostate 相反，终止于保留状态配置的最内层状态，包括执行沿途遇到的所有入口行为。

浅历史（状态）表示只返回最近状态配置的最顶层子状态，使用默认进入规则进入。

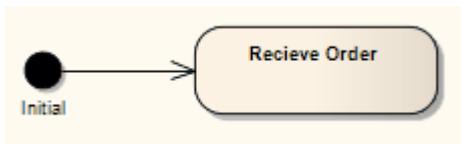
# 最初的



## 描述

活动状态机起始元素被状态图使用。在活动图中，它定义了调用活动时流程的开始。对于状态机，初始元素是一个伪状态，用来表示复合状态状态复合状态的每个区域可以有一个初始顶点。

这个简单的示例显示了接收订单的流程的开始。



请参阅 [OMG Unified Modeling Language规范](#)，（ 1，图 12.97，第 378 页。）

活动由终点或流终点节点完成。

## 注记

- 移动图一般不会影响包内元素的位置；如果将图表从一个包中移到另一个包中，则图表中的所有元素都保留在原始包中

但是，初始元素仅在一个图表中使用，在该图表之外没有任何意义，并且永远不会在任何其他图表中重复使用；因此，如果您移动包含这些元素的图表，它们将与图表一起移动到新的父包

## 工具箱icon



## OMG UML规范：

活动图形中的首图表

OMG Unified Modeling Language规范 (v2.5. 1 , p.387) 还指出：

InitialNode 是作为执行活动的起点的 ControlNode。一个活动可能有多个 InitialNode。如果一个活动有多个 InitialNode，则调用该活动会启动多个并发控制流，每个 InitialNode 一个。

InitialNode 不应有任何传入的 ActivityEdges，这意味着活动拥有的活动将始终在活动开始执行时启用，并且在活动执行开始时将单个控制令牌放置在每个此类 InitialNode 上。InitialNode 的传出 ActivityEdges 必须都是 ControlFlows。放置在 InitialNode 上的控制令牌在所有传出 ControlFlow 上同时提供。

初始状态机图表

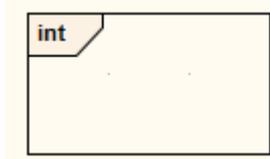
OMG Unified Modeling Language规范 (v2.5. 1 , p.312) 指出：

一个初始的 Pseudostate 代表一个区域的起点；也就是说，它是通过默认激活进入区域时开始执行其包含的行为的点。最多是一个转移

的源转移

- 它可能具有关联的效果行为，但没有关联的触发器或防护。在一个区域中最多可以有一个初始顶点。

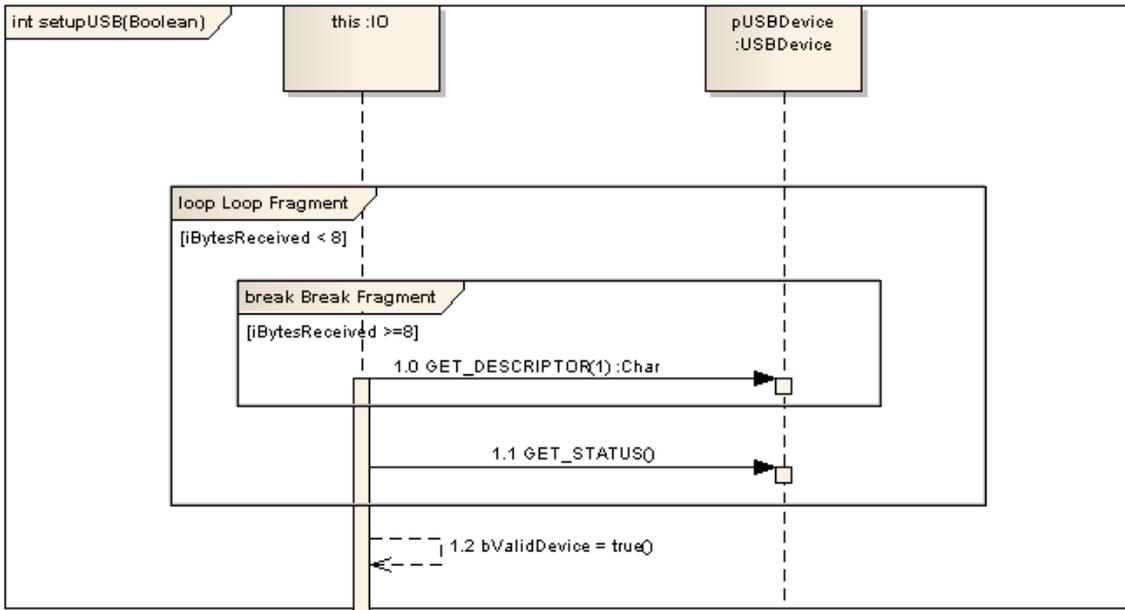
# 交互



## 描述

您可以使用一个交互来插入一个元素交互作为一个类元素。交互可以包含元素任何类型的图表：

- 序列
- 通讯
- 定时

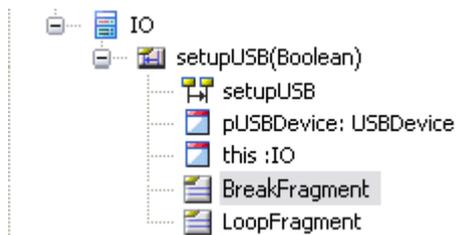


Enterprise Architect交互的元素被视为封装在其中的分类器的行为。它可以有参数和返回类型，这些参数和返回类型是通过使用交互元素的“属性”对话框的“行为”选项卡进行建模的。元素被解释为生成代码中包含类的方法（请参阅从行为模型生成代码主题）。

一个交互的调用交互发生分类器在元素图上设置为序列图，行动在活动图中设置为行为图。建立这样的关联（在行为和行为调用之间）有助于添加可以单独映射到相关行为参数的参数。

## 注记

- 行为元素期望序列图及其关联的Messages 和Combine片段全部封装在此示例代码生成引擎中（如交互）



( IO类可在 EAExample模型中的 系统工程模型|实施模型|软件”下找到 )

- '使用交互'添加的图标在'工具工具'工具栏的交互'页面上工具箱，但只能通过上下文上的元素菜单或浏览器中的元素添加到元素中

## 工具箱icon



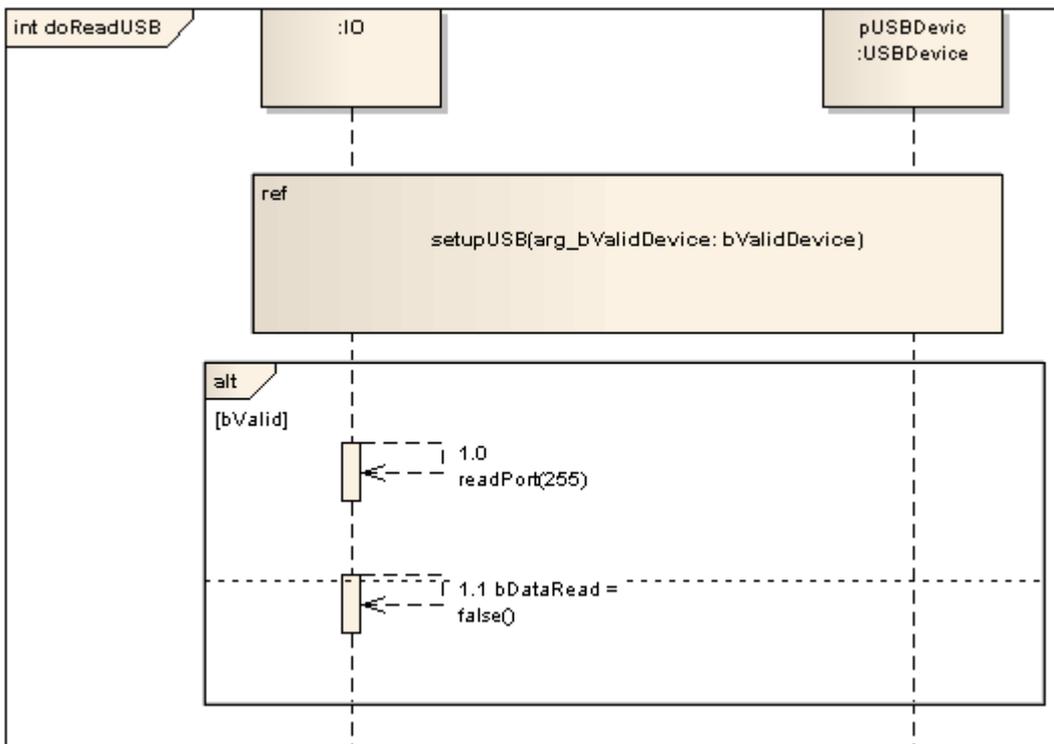
# 交互发生



一个交互发生（或交互）是对现有框架的交互（序列图。使用框架）在视觉上用框架表示，在标题空间中带有“事件引用”。图表名称显示在框架内容中。

要创建一个交互发生，只需打开一个序列（最好包含在一个交互元素），然后将另一个序列（最好包含在一个元素）交互其工作区中。将显示A对话框，提供配置选项。所产生的交互发生作为原始行为的一种交互。你可以用调用属性的属性来把元素设置为属性的交互，也可以将属性属性的属性元素不同的交互。

此图说明了一个交互发生在另一个交互中的使用（序列）图。您可以通过交互发生显示元素上的序列来表示顺序。



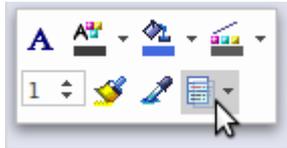
## 填充不透明度

虽然通常会交互发生一些其他元素的事件，但有可能是隐藏这些元素的原因，以充分显示它们，或者可能只是表明它们存在，这取决于这些元素的直接目的。您可以通过更改元素的不交互发生，将这些细微差别应用到显示元素背后、覆盖或重叠的显示中。

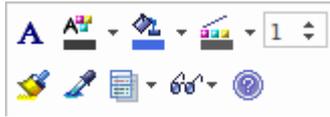
在设置不透明度之前，请检查元素是否具有填充颜色。

您可以使用以下两个弹出式元素工具栏中的图标设置不透明度：

- 点击交互发生元素：和 图标



- 右键单击交互发生事件的元素并查看上面的上下文菜单：

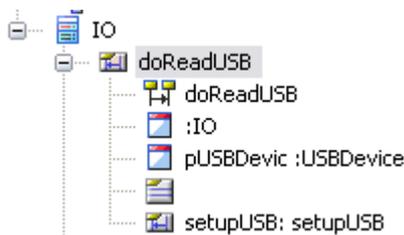


单击  图标并选择：

- 100% 的总不透明度，交互发生在隐藏元素后面和重叠或覆盖的元素（您可以右键单击单个元素并选择“Z-Order | 带到顶部”选项以仅显示这些元素）
- 0% 表示交互发生透明度、未应用填充颜色且填充颜色后面的任何内容完全可见
- 75%、50% 或 25% 以设置适当的不透明度并使被覆盖的元素可见但过度着色

## 注记

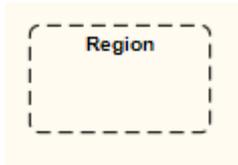
- 行为生成代码将其关联消息的序列dia和元素片段封装在一个交互引擎中（如本例中的序列）



## OMG UML规范：

OMG Unified Modeling Language规范，（ v.2.5. 1 p.620 ）指的是Unified交互发生作为一个InteractionUse 并指出：一个 InteractionUse 是指一个交互。 InteractionUse 是用来复制引用交互的 InteractionUse 的简写。为了准确起见，复制必须考虑用参数替换参数并将形式门与实际门连接起来。

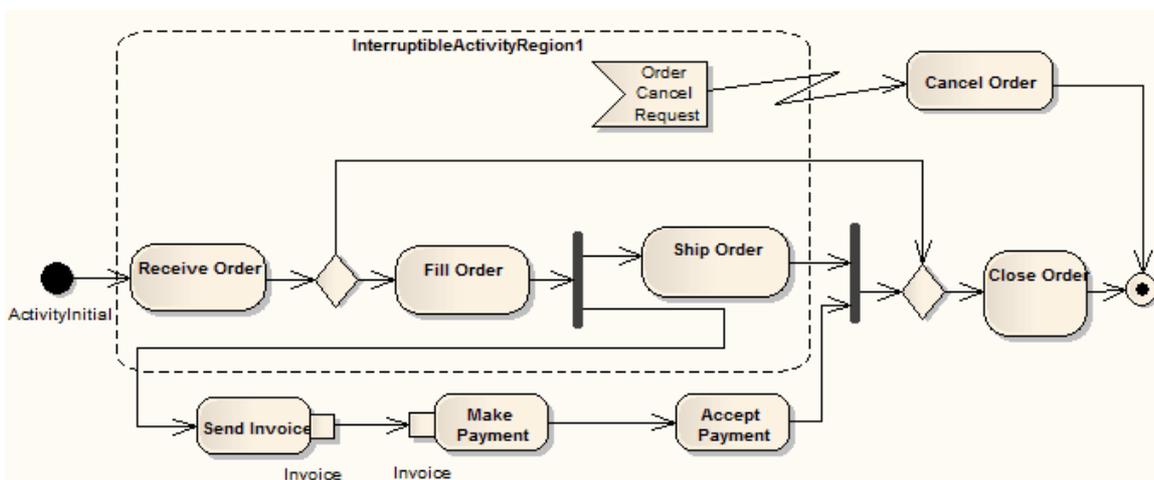
# 可中断活动区域



## 描述

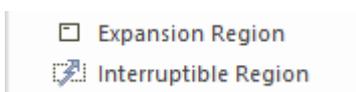
在活动图中，可中断区域围绕着一组活动元素，所有活动元素都受到某些中断的影响，因此在该区域内传递的所有令牌都会在中断被引发时终止。任何在可中断活动区域范围内发生的处理都会在跨中断流向外部元素发起流时终止。

您可以使用图表的活动页面创建一个可中断活动区域作为区域的一种工具箱区域（另一个是扩展区域）。此示例说明订单取消会终止在接收、填充或发货阶段对订单的任何处理。



请参阅 OMG Unified Modeling Language 规范，（ 1 · 图 12.100 · p.381 。 ）

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language 规范 (v2.5. 1 , p.407) 指出：

InterruptibleActivityRegion 是一个支持终止部分活动的 ActivityGroup。InterruptibleActivityRegion 仅包含 ActivityNode。它还将源在区域内而目标在区域外的某些 ActivityEdge 标识为中断边缘。当沿着 interruptingEdge 提供的令牌被接受并遍历该边缘时，该区域的所有包含节点的执行都将终止，并且所有令牌都将从它们中删除。

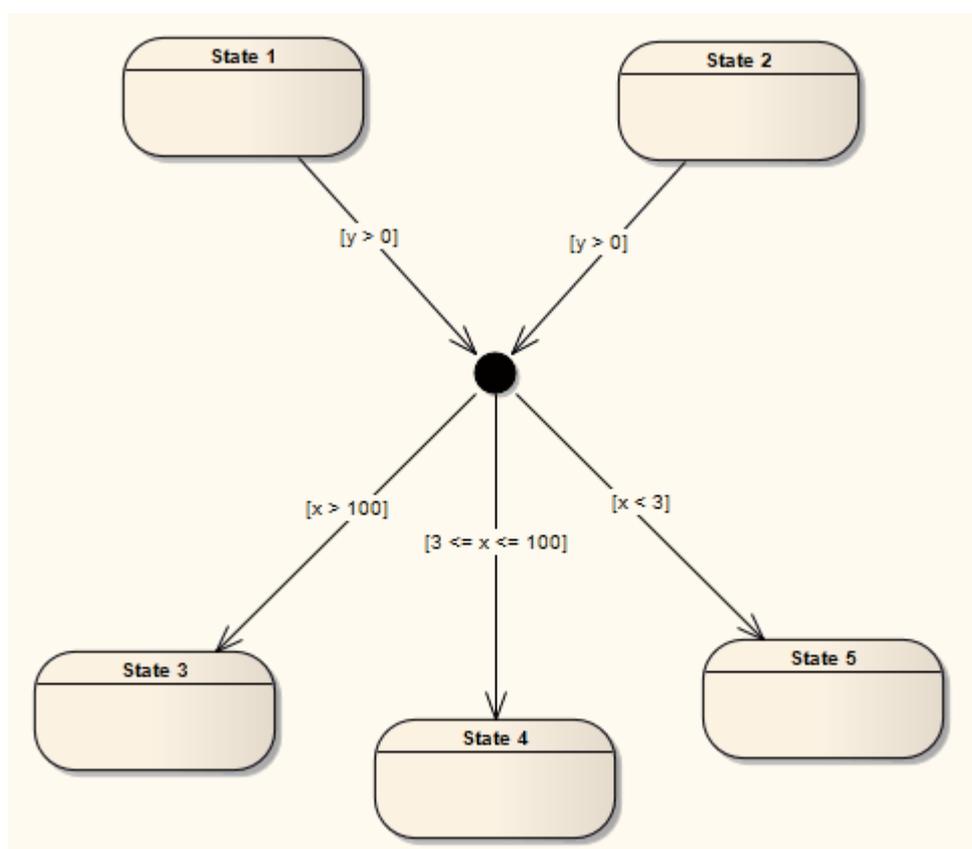
# 连接点



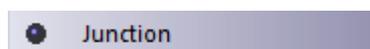
## 描述

连接点伪状态用于设计状态机图中复杂的过渡路径。A连接点可用于将多个路径组合或合并为共享转换路径。或者，连接点可以将传入路径拆分为多个路径，类似于分叉伪状态。与分叉或汇合不同，结合点可以对每个传入或传出转换应用保护，这样如果保护表达式为False，则禁用转换。

此示例说明如何将保护应用于进入或离开连接点伪状态的转换。



## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.313) 指出：

这种类型的伪状态用于将多个转换连接到状态之间的复合路径。例如，一个结点 Pseudostate 可用于将多个传入

转换合并为一个传出转移

表示共享的延续路径。或者，它可以用于拆分传入的转移

成多个传出转移

段有不同的约束。

笔记。在执行任何包含此伪状态的复合转换之前，会评估此类保护约束，这就是为什么这被称为静态条件分支。

对于特定的复合跃迁，可能会发生转移

的构型转移

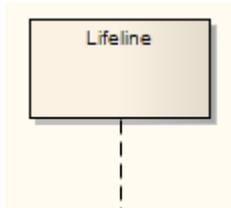
路径和保护值是这样的，以防止复合转换达到有效的状态配置。在这些情况下，即使启用了触发器转换，整个复合转换也会被禁用。（在某些情况下，作为避免这种情况的一种方式，可以将表示为“else”的预定义保护最多一个传出转移

相关联转移

.这个转移

如果附加到其他转换的所有守卫评估为false，则启用)。如果不止一个守卫评估为真，则选择其中之一。未定义进行此选择的算法。

# 生命线



## 描述

生命线是交互中A个体参与者（即生命线不能有多重性）。A生命线代表了一个独特的可连接元素。要在 Enterprise Architect 中指定该表示，请右键单击生命线并选择 高级|“实例分类器的选项。将显示 选择 <Item>”对话框，您可以使用它来定位所需的项目分类器。

生命线在序列图中可用。时序图有不同的生命线元素（状态生命线和价值生命线）；然而，虽然这两种图表类型的表示不同，但生命线的含义是相同的。

## 工具箱icon



## OMG UML规范：

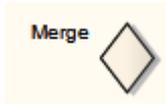
OMG Unified Modeling Language规范 (v2.5.1, p.572) 指出：

在交互图中，生命线描述了流程的时间线，其中时间在页面下方增加。时间线上的两个事件之间的距离并不代表任何时间的字面测量，只是已经过去了那个非零时间。

同一时间线上的事件在页面下方线性排序，除非它们发生在平行组合片段内，或沿着“coregion”内的生命线发生。（...）在并行组合片段或共同区域内，事件不是局部排序的，除非这是由一般排序构造直接强加的。（...）。

沿着生命线的生命线的顺序很重要，表示这些 OccurrenceSpecifications 将出现的顺序。然而，生命线之间的绝对距离与语义无关。

# 合并



## 描述

A Merge节点a Merge together a flow paths in活动,分析和选择交互图表。例如，如果在决策之后使用分叉，则从决策中出来的两个决策必须合并为一个，然后才能进入汇合；否则，汇合等待两个流，只有一个到达。

A Merge节点有多个输入边和一个输出边。进出 Merge节点的边必须是所有object流或所有控制流。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.427) 指出：

合并节点是汇集多个备用流A控制节点。它不用于同步并发流，而是接受多个备用流中的一个。

# 信息端点

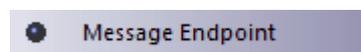


## 描述

A信息端点元素定义了时序图中状态或值生命线的终止。它表明信息：

- 终止于状态或价值生命线之外的未定义点，开始于生命线内的确定点，或
- 起源于状态或价值生命线之外的未定义点，终止于生命线内的确定点

## 工具箱icon



# 信息标签

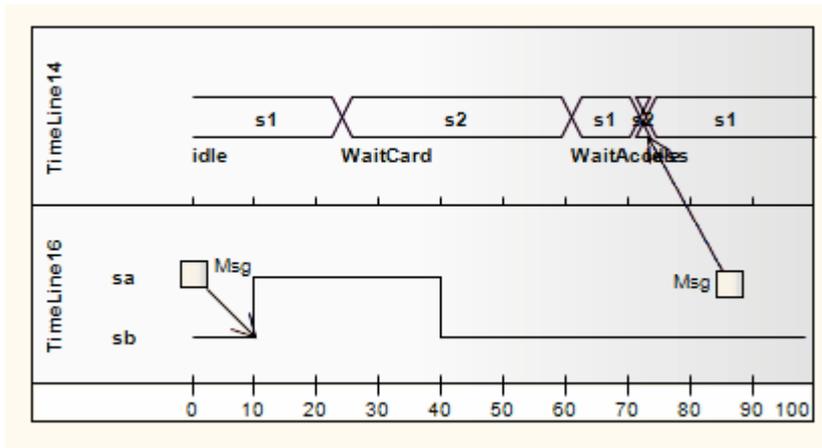


Message Label

## 描述

A信息标签是表示生命线之间消息的另一种方式，这对于整理“散布着消息的时序图很有用。要指示生命线之间的信息，请从源生命线中画一个连接生命线到信息标签中。接下来，绘制从另一个信息标签到目标生命线的连接符。注记标签名称必须匹配以反映消息出现在两个信息标签之间。

此图说明了如何使用信息标签在生命线之间构建消息。



## 工具箱icon

Message Label

## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.577) 指出：

出现在 request-message-label 中的 message-name 是信息的名称属性。如果信息有签名，这将是操作的名称或信号所引用的信号。否则，名称不受约束。如果 request-message-label 包含一个 input-argument-list，那么要么所有的 input-arguments 都必须有一个给定的 in-parametername 或者没有一个。

# 注记



## 描述

注记元素是A文本注释，可以附加到任何其他类型的一组元素上。附件是使用注释链接连接器单独创建的。通过工具箱的“公共”页面，可以在任何Enterprise Architect图表中获得注记和注释链接。

注记又称注解A

注记元素没有属性对话框。您可以在属性窗口中为它们命名，但这不会显示在图表上。

您可以配置Enterprise Architect以斜体显示所有注记元素中的文本。选择 开始>外观>首选项>首选项”功能区选项，然后在 图表>外观”页面上选择 斜体注记元素文本”复选框。这有立竿见影的效果，清除复选框以正常字体显示文本也是如此。

## 工具箱icon

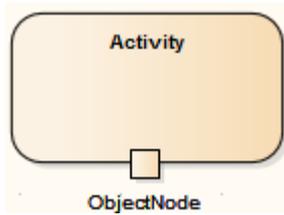


## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.40) 指出：

注解是可以附加到A组元素的文本注释。

# 物件节点

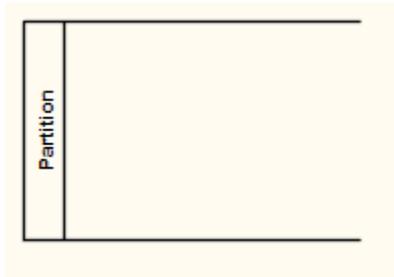


## 描述

一个物件节点保存了一个活动元素的输入或输出数据。要设置一个物件节点的类型，点击它并按Ctrl+L（选择一个实例分类器）。物件节点可以通过物件流连接器连接；如果物件流两端的物件节点是类型化的，那么它们的类型应该是兼容的。

行动销是物件节点的一种特殊形式。

# 分区

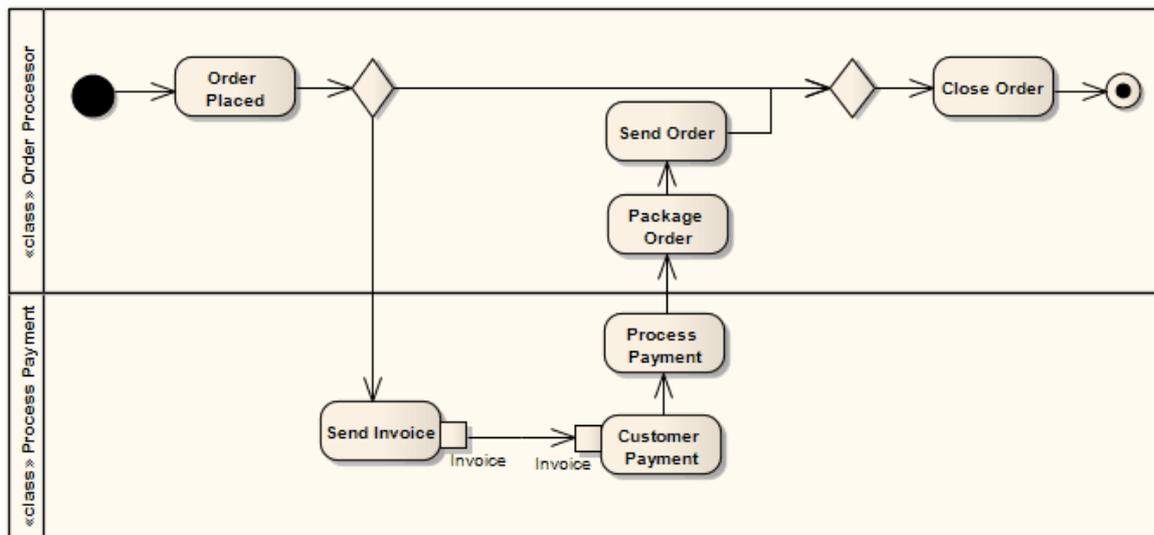


## 描述

Enterprise Architect支持两种类型的活动分区：

- 活动分区特征用于在逻辑上组织活动元素
- 活动分区元素，在本主题中描述，用于逻辑组织活动图

它们具有相似的效果——它们在不影响令牌流的情况下划分活动的行动，有助于构建活动的视图或部分。这个例子描述了 Classes进程Payment和Order Processor之间的划分。



分区方向默认为水平。要将其变成垂直分区，请右键单击它并选择 高级|“垂直分区”选项。

您可以使用元素上下文菜单 可停靠”选项整齐地对齐和加入图表上的活动分区。对于分区，该选项默认为选中。

## 设置不透明填充

虽然直接活动分区通常包含许多其他元素，但可能有隐藏这些元素的原因以及完全显示它们的时间，或者可能只是表明它们在那里，这取决于图表的目的。您可以通过更改元素填充颜色的不透明度来将这些细微差别应用到分区内容的显示中。

在设置不透明度之前，给元素一个填充颜色。

您可以使用以下两个弹出式元素工具栏中的图标设置不透明度：

- 点击活动分区元素和 图标：



- 右键单击活动分区元素并查看上下文菜单上方：



单击  图标并选择：

- 总不透明度为 100% · 其中活动分区的内容被隐藏（您可以首先右键单击单个元素并选择“Z-Order | 带到顶部”选项以仅显示这些元素）
- 0% 表示无不透明度 · 不应用填充颜色且活动分区内容完全可见
- 75%、50% 或 25% 设置适当的不透明度并使内容元素可见但过度着色

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.406) 指出：

ActivityPartition 是一种 ActivityGroup · 用于识别具有某些共同特征的 ActivityNode 。 ActivityPartitions 可以共享内容。它们通常对应于业务模型中的组织单位。它们可用于在活动的节点之间分配特征或资源。

# 接收

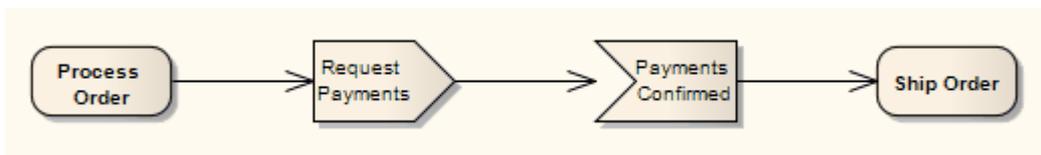


## 描述

接收元素用于A活动图中定义请求的接受或接收。仅当根据其规范完成接收时，才会发生从接收元素的移动。接收元素有两种形式：

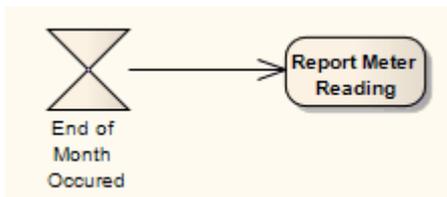
- 接受事件行动元素（三角旗形状）
- 接受时间事件行动元素（沙漏形状）

此示例反映了订单的付款流程。收到付款后（来自 Request Payments，发送元素），付款得到确认，流程继续运送订单。



请参阅 OMG Unified Modeling Language规范，（ 1，图 12.26，p.312。）

要描述接受时间事件，请使用工具箱中的标准接收元素。右键单击该元素，然后选择 高级|“接受时间事件”选项。这个例子展示了沙漏形的接受时间事件行动：



请参阅 OMG Unified Modeling Language规范，（ 1，图 12.27，第 312 页。）

## 工具箱icon



## OMG UML规范

OMG Unified Modeling Language规范 (v2.5. 1, p.489) 指出：

行动是一个等待一个或多个特定事件发生的动作。

## 区域



### 描述

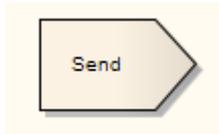
Enterprise Architect支持两种类型的区域元素：

- 拓展区域
- 可中断活动区域

### 工具箱icon



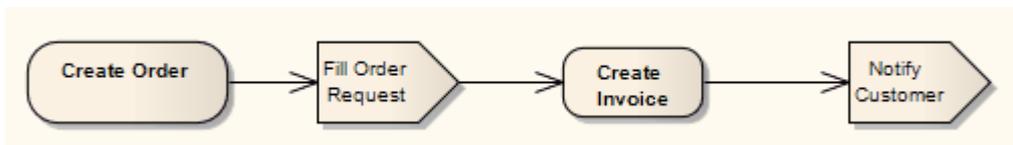
# 发送



## 描述

发送元素用于在活动图中描述发送信号的动作。它与接收元素相反。您还可以使用工具箱的“状态”页面上的“事件”图标创建发送事件。

此示例显示正在处理的订单，其中发送信号以填写已处理的订单，并且在创建生成的发票后，向客户发送通知。



请参阅 OMG Unified Modeling Language 规范，（v2.5.1，图1，第 408 页。）

## 工具箱icon

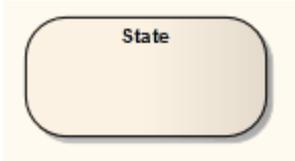


## OMG UML规范：

OMG Unified Modeling Language 规范 (v2.5.1, p.452) 指出：

SendObjectAction 是 A 调用动作，它将任何类型的 object 传输到其目标 InputPin 上给定的 object。要传输的 object 在 SendObjectAction 的单个请求 InputPin 上给出。如果 object 是一个信号实例，那么它可以由目标 object 以与从 SendSignalAction 或 BroadcastSignalAction 发送的实例相同的方式处理。否则，只能使用 AnyReceiveEvent (...) 来处理 object 的接收。

# 状态



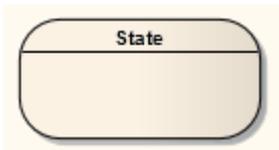
## 描述

A表示某些不变条件成立的情况；此条件可以是静态的（等待事件）或动态的（执行一组活动）。状态建模通常与类相关，并描述类或元素可以处于的可启用状态以及使元素能够移动到那里的转换。状态有两种类型：简单状态和复合状态，状态通过工具箱的“状态”图标创建。

此外，还有伪状态，类似于状态的某些方面，但具有预定义的含义。伪状态可以对复杂的过渡路径模型，并对常见的状态机行为进行分类。

您可以使用操作定义状态的进入、内部和退出操作。状态元素可以具有三个操作（进入、执行和退出），这些操作是通过特征窗口的“行为”选项卡（开始>所有窗口>属性>元素特征>特征）创建和定义的。仅当所选元素为状态时，该选项卡才会显示。它会自动列出这三个操作，您可以在“名称/注解”字段中键入文本值，也可以使用属性窗口的“行为”页面分配代码的行为元素（请参阅操作行为帮助主题）。

如果状态元素具有诸如操作、内部触发器或继承的操作和属性等特征，则图表中元素的描述在元素名称下会有一条线。如果特征被隐藏，此线仍然存在。如果在“首选项”对话框的“对象”页面上选中“显示状态隔间”复选框，也会显示该线（选择开始>外观>首选项>首选项“功能区选项和对象”页面）。



## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.308) 指出：

状态对执行状态机行为的情况A建模，在该情况下，某些不变条件成立。在大多数情况下，此条件不是明确定义的，而是隐含的，通常通过与状态关联的名称。

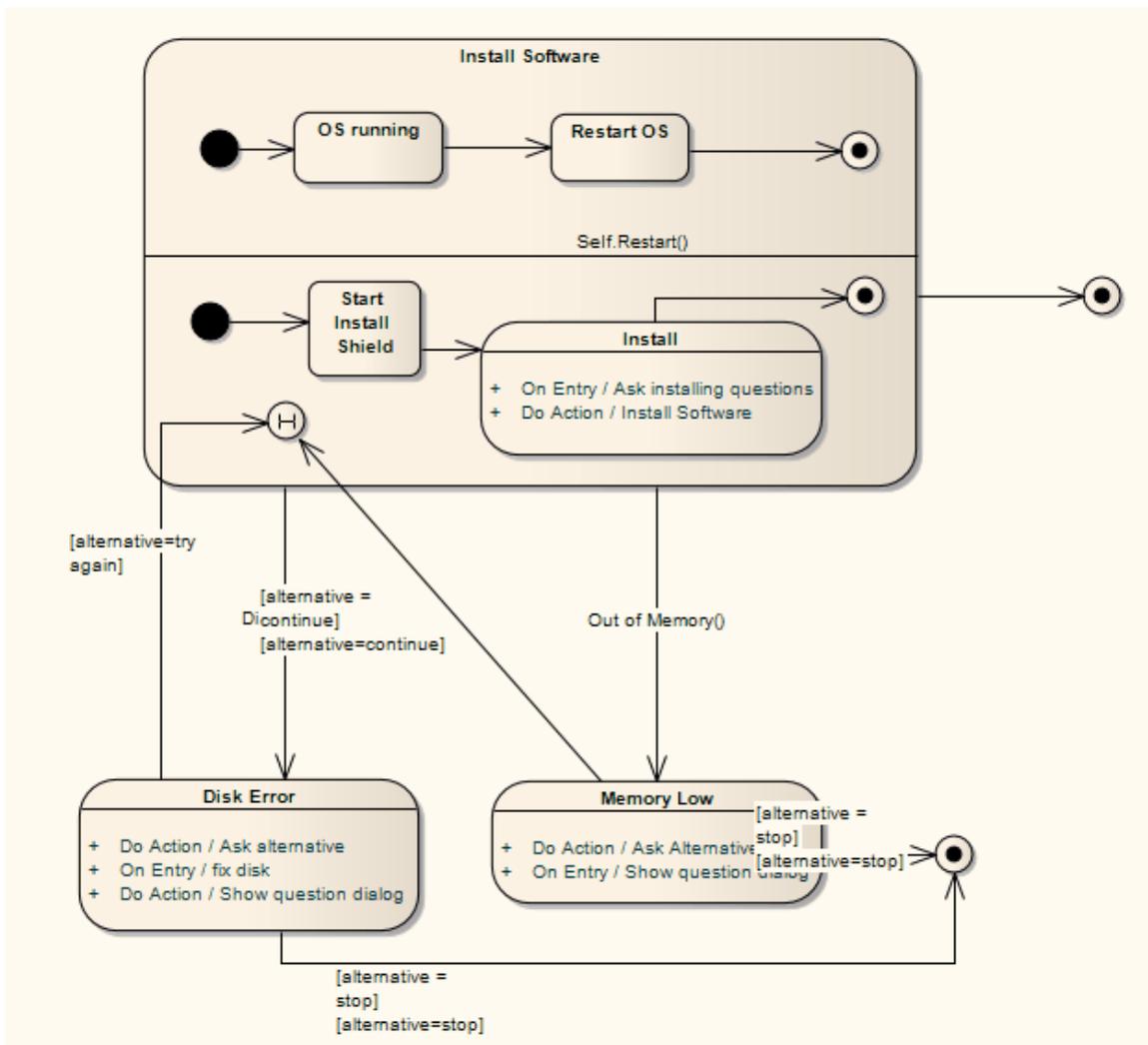
# 复合状态

## 描述

通过复合状态元素、添加区域（如果适用）并在其边界内拖动进一步的状态元素、相关元素和连接器，组合状态在状态机图中组成。然后将内部状态元素称为子状态。

（与许多其他类型的元素一样，您也可以将状态元素定义为复合元素；然后它具有到子图的超链接，该子图可以是模型中其他地方的另一个状态机图或其他类型的图。）

如果创建了区域，则复合可以是正交的。如果复合状态是正交的，则其条目表示单个子态状态在每个区域中同时处于活动状态。复合状态的主动区域嵌套使用，产生一个状态并发；这种情况称为活动状态配置。



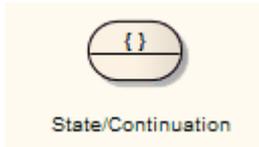
## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.308) 指出：

复合状态包含至少一个区域，而子状态是指整个状态机，从概念上讲，它被认为是“嵌套”在状态中。复合状态可以是具有恰好一个区域的简单复合状态，也可以是具有多个区域的正交状态（状态=true）。(...)任何状态在复合状态区域子状态。当它不包含在任何其他状态中时，称为直接子状态；否则，它被称为间接子状态。



# 状态/延续



## 描述

状态/延续元素两个不同的用途：作为状态交互序列 ) 图。系统提示您在创建元素时确定用途。

## 工具箱icon



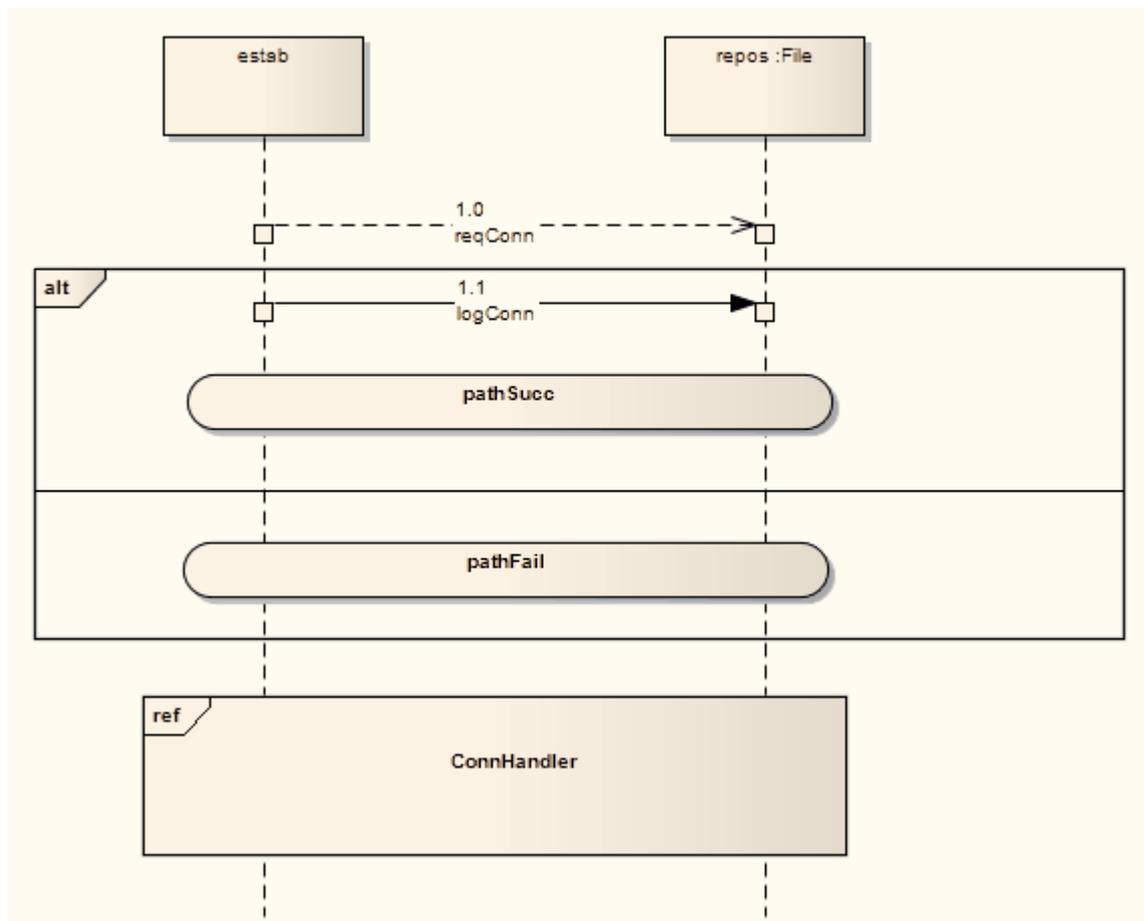
# 延续

## 描述

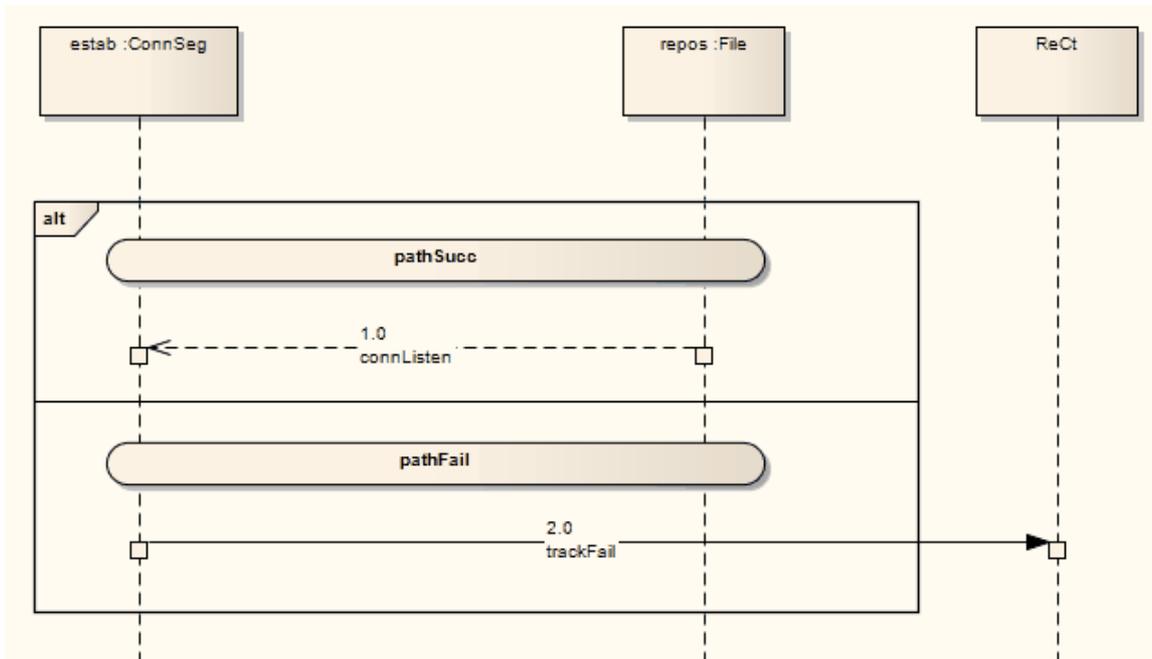
A用于 seq 和 alt 组合片段，表示操作数延续的延续分支。表示一个连续的结束，用一个连续的延续，并在一个连续的交互之前用一个匹配的延续（名字）来表示一个连续的分支。

您可以通过从交互元素元素状态延续工具箱延续创建一个。

对于这个例子，一个 alt 组合片段有 Continuations延续和 pathFail。这些交互发生位于后续事件中，后续事件基于 continuationnHandler。



此图显示了发生所引用的交互发生。



## OMG UML规范

OMG Unified Modeling Language规范 (v2.5. 1 , p. 609) 指出：

A是一种语法方式，用于定义替代延续的不同分支的延续。延续在直观上类似于表示控制流中的中间点的标签。

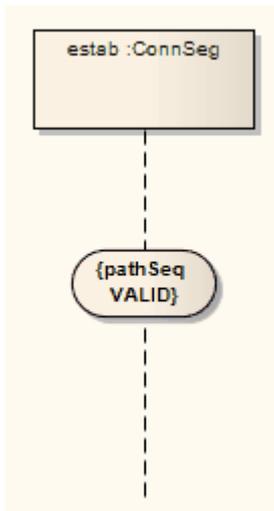
OMG Unified Modeling Language规范 (v2.5. 1 , pp. 582-583) 还指出：

延续仅与选择CombinedFragments 和 (弱) 排序有关。如果一个选择组合的延续以带有名称 (比如) X 的操作符结尾，则只能附加以 X 开头的延续 (或根本没有延续)。

## 状态不变

状态不变量是应用于生命线A条件，生命线必须满足该生命线才能存在。您通过状态工具状态/延续元素的交互来创建工具箱元素。

此图说明了状态不变量。



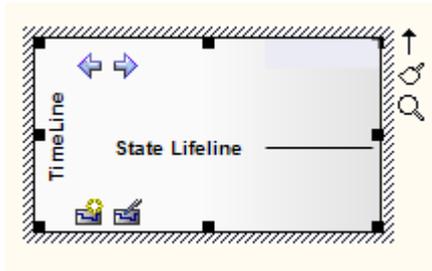
当状态不变量靠近生命线时，它会捕捉到中心。如果向左或向右拖动序列object，则状态不变量会随之移动。

### OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p. 630) 指出：

A是对销售参与者的运行时交互。它可用于指定各种不同类型的约束，例如属性或变量的值、内部或外部状态等。A是一个 InteractionFragment，它被放置在一条生命线上。

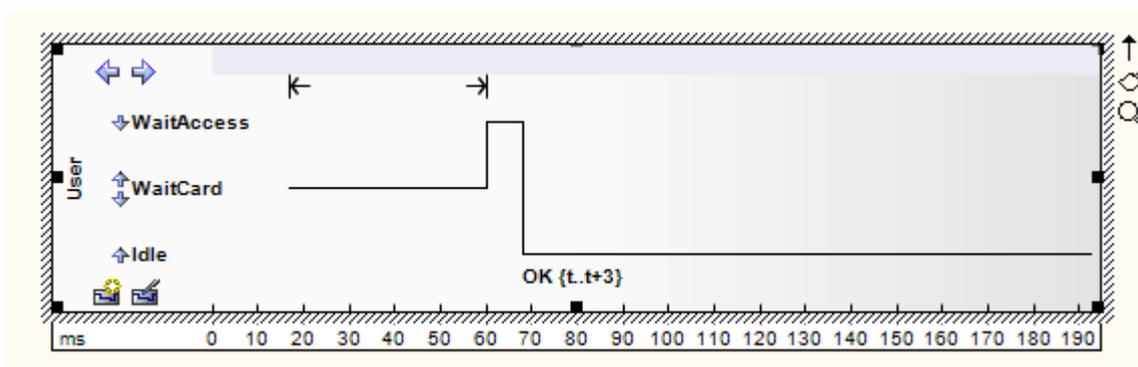
# 状态生命线



## 描述

生命线是object在一定时间内所走A路径，如 x 轴所示。有两种：状态生命线（在此处定义）和价值生命线，都用于时序图中。

状态生命线遵循状态之间A离散转换，这些转换沿时间线的 y 轴定义。任何转换都具有时序约束、持续时间约束和观察的可选属性。此处显示了状态生命线的示例：



## 转移点属性

A状态生命线由一组过渡点组成。可以使用以下属性定义每个转换点：

属性	描述
当时	指定状态更改的开始时间。
转移至	状态变化的状态。
事件	描述正在发生的事件。
时间限制	指状态在生命线内发生变化所用的时间，或传递消息所用的时间（例如，状态+3）。
时间观察	提供有关状态更改或发送消息的时间的信息。
	与特定状态下生命线的周期有关。约束可以由生命线内的状态变化或生命线

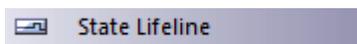
持续时间限制	收到消息引发。
持续时间观察	表示特定状态下生命线的间隔，从状态或消息接收开始。

## 示例属性

在示例图中，确定转换点具有以下属性：

属性	价值
在时间	68 毫秒
转移至	闲置的
事件	确定
时间限制	t..t+3
时间观察	-
持续时间限制	-
持续时间观察	-

## 工具箱icon



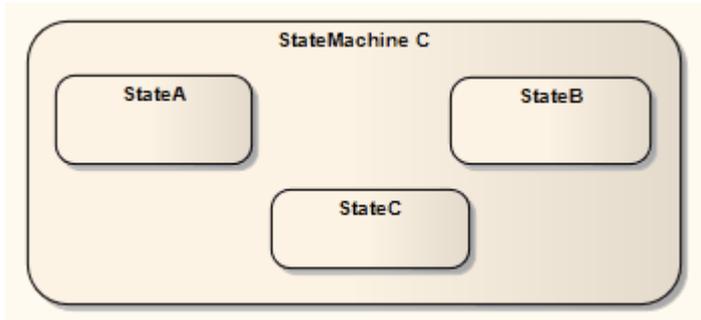
## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.604) 指出：

这是状态分类器或属性的状态，或一些可测试的条件，例如离散的可枚举值。

也可以让状态维度是连续的和离散的。这对于某些实体经历连续状态变化（例如温度或密度）的场景是说明性的

# 状态机



## 描述

状态机元素是一组相关状态元素的容器。您可以创建状态机图的部分，显示相互状态的元素的组织，并将每个部分包含在一个状态机元素中。您还可以在状态机元素上创建区域。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 ( 1 · p.306 ) 指出：

A行为状态机包含一个或多个区域，每个区域包含一个图（可能是分层的），该图包含一组由表示转换的弧互连的顶点。状态机执行由适当的事件触发。状态机的A执行由通过一个或多个区域图的一组有效路径遍历表示，由调度与这些图中的活动触发器匹配的事件发生触发。...在这样的遍历过程中，状态机实例可能会执行与正在遍历的图形的特定元素相关联的潜在复杂行为序列（转换效果、状态进入和状态退出行为等）

# 结构活动

结构活动元素用于活动图中。A结构活动是一个活动节点，可以有从属节点作为一个独立的活动组。您可以设置一个选项以确保没有其他活动或其副作用干扰此活动的处理（“结构活动元素”对话框中的“必须隔离”属性复选框）。

Enterprise Architect提供了多种形式的结构活动，包括基本的和专业的。

## 访问

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

## 创建结构化活动

当您把结构活动图标从工具箱拖到图表上时，会显示一个简短菜单，您可以从中选择以下选项之一：

- Loop节点
- 条件节点
- 其它

前两个选项专门创建一个 Loop节点或 Conditional节点。

其它“选项显示 新结构活动”对话框，您可以在该对话框上选择创建五种结构活动元素之一。

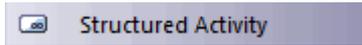
## 结构活动类型

类型	描述
简单复合活动	使用儿童活动图生成复合活动元素。
Loop节点	代表要在object内序列的行动和活动的顺序。
条件节点	代表行动和活动的安排，选择决定执行哪些活动。
结构活动节点	代表可执行活动节点（行动、决策、合并等）的有序排列，可以包括分支和嵌套节点；这是其他类型的结构活动派生的基础元素。
顺序节点	代表可执行活动节点的顺序排列。

## 注记

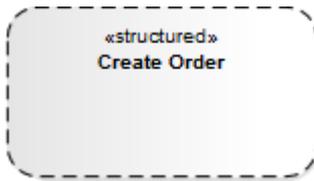
- 要保护循环或条件节点结构活动的处理免受其他活动或其副作用的干扰，请打开属性窗口并选中“循环”或“条件”选项卡上的“必须隔离”复选框

## 工具箱icon



## 结构化节点

在图上，结构活动节点的边界是断开的，如图所示。



可以嵌套节点下的其他元素，包括Conditional、Loop等其他结构化节点等元素。

### OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.476) 指出：

StructuredActivityNode 是A行动，它也是一个 ActivityGroup (...)，其行为由它所包含的 ActivityNodes 和 ActivityEdges 指定。与其他类型的 ActivityGroup 不同，StructuredActivityNode 拥有它包含的 ActivityNodes 和 ActivityEdges，因此节点或边只能直接包含在一个 StructuredActivityNode 中。StructuredActivityNodes 可以嵌套（作为一个 StructuredActivityNode，作为一个行动，也是一个 ActivityNode），但是，一条边或节点可能间接包含在多个嵌套的 StructuredActivityNodes 中。

## 顺序节点

在图表上，顺序活动节点具有打破边界，并且可以包含定义序列动作的嵌套元素。

时序节点在上下文菜单中被标记为复合元素（“新子图表|复合”）；但是，当您添加子图时，元素将转换为简单的复合活动。



### OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.480) 指出：

SequenceNode 定义了它包含的所有 ActivityNodes A完成、顺序排序，这些 ActivityNodes 必须都是 ExecutableNodes。当 SequenceNode 执行时，其中的每个节点都按顺序执行。SequenceNode 也可能在其节点之间包含 ActivityEdges，而 ActivityEdges 可能会跨入和跨出 SequenceNode。语义等价于包含相同节点和边的一般 StructuredActivityNode，但添加了控制流以按照为SequenceNode 指定的顺序对节点进行排序。

## Loop节点

循环结构活动节点A定义循环，通常与“While”、“Repeat”或“for”循环语句相关联。

每个 Loop 节点有三个分区：

- 安装程序通常会启动要在循环退出条件中使用的变量；它在进入循环时执行一次
- 测试定义循环退出条件
- 体可以包含行动执行的动作，直到测试产生false值

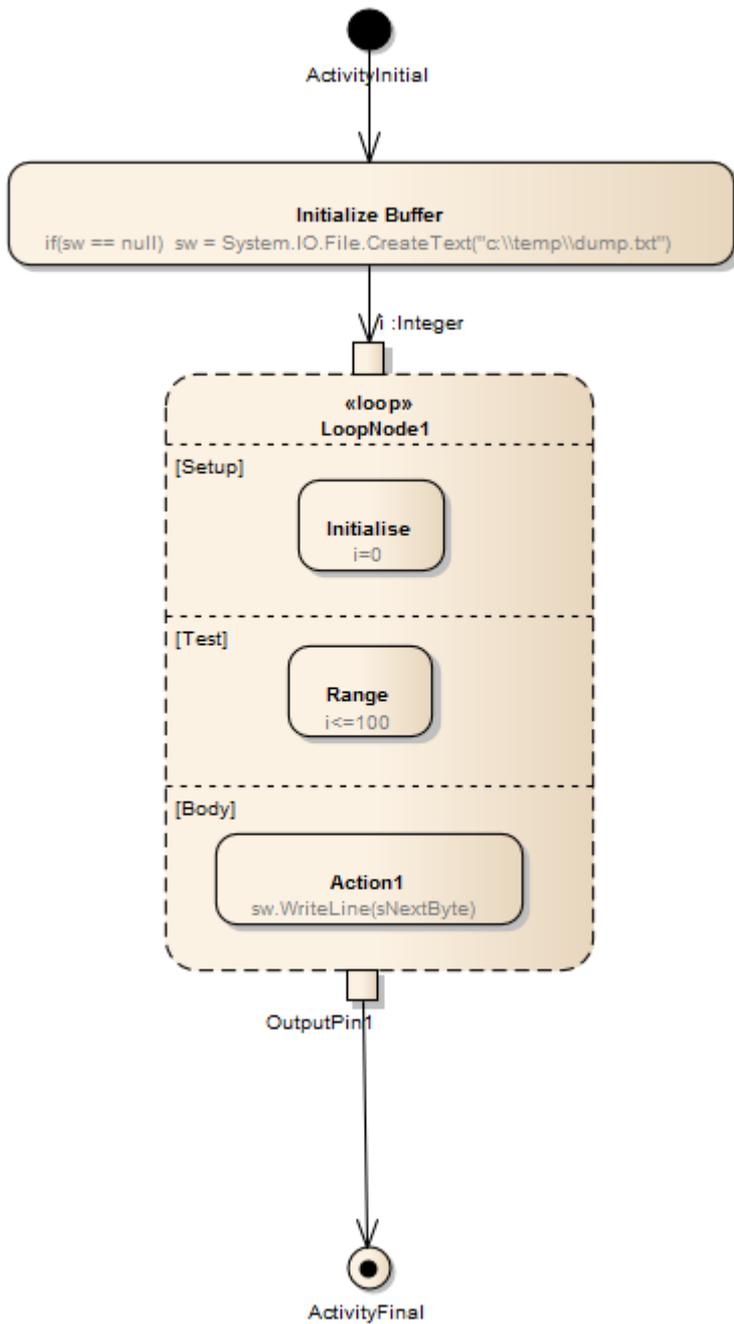
测试或体的最终执行结果在循环执行完成后可用。

### 创建一个 Loop 节点

活动图上A Loop节点是这样描述的：



您可以通过将图表工具箱行动设置”、“测试”和“体”分区来定义循环节点。“体”分区可以包含多个行动，可以链接并组织成所需的结构。元素在分区的左上角对齐，因此调整节点大小可以保持分区内和分区之间的结构组织。如果您尝试将节点缩小到结构大小以下，则节点会自动默认为“最佳拟合”大小。



节	行动
1	从工具箱的活动页面中，将一个结构活动图标图表活动图表上。显示A简短的菜单。
2	选择“循环节点”选项。 Loop节点显示在图上，带有元素的“属性”对话框（如果对话框不显示，双击元素）。
3	根据需要完全尽可能多的常见元素属性字段，然后关闭“属性”对话框。
4	显示 Loop节点的属性窗口（开始>应用程序>设计>属性），然后单击“循环”选项卡。根据需要设置这些复选框： <ul style="list-style-type: none"> <li>• 'Must Isolate' - 定义并发：如果选中，则节点内的任何object都不能在其外使用；对象与并行使</li> </ul>

	<p>用隔离</p> <ul style="list-style-type: none"> <li>'Tested First' - 定义循环类型；选择一个 For / While 循环，取消选择一个重复直到循环</li> </ul>
5	<p>对于这些字段中的每一个，根据需要单击  或添加按钮，以显示“选择销”对话框并选择一个行动销：</p> <ul style="list-style-type: none"> <li>决策者（测试“分区内的一个输出销，在执行测试后检查其值以确定是否执行循环体）</li> <li>循环变量输入</li> <li>循环变量</li> <li>体输出和</li> <li>结果</li> </ul> <p>选择销“对话框仅列出“循环变量输入”字段的输入销，而仅列出其他字段的输出销。</p> <p>如果所需的行动销尚不存在，您可以单击对话框上的“加新”按钮为节点自动创建输入引脚或输出引脚。</p>
6	<p>在“节点”面板中，单击“设置”、“测试”或“体”单选按钮之一，列出 Loop 节点相应分区中包含的行动和活动。</p> <p>元素必须完全低于要为该分区列出的分区的顶部边缘 - 如果它以任何方式与上面的分区重叠，则它被视为该分区的一部分。</p>
7	<p>点击确定按钮保存 Loop 节点的属性。</p>
8	<p>右键单击节点中的“特征节点”并选择“特征交互”选项。</p> <p>特征点显示“交互点”选项卡。</p> <p>选中交互点的复选框。</p> <p>行动图钉现在应该在图中可见，附在节点上。</p>

## 注记

- 您可以通过右键单击 Loop 节点属性窗口中的引脚名称并选择“在项目中查找浏览器”选项来查看现有行动销的确切位置；浏览器窗口中行动销的位置被展开并突出显示

## OMG UML 规范：

OMG Unified Modeling Language 规范 (v2.5.1, p.479) 指出：

A 是一个 StructuredActivityNode，它表示一个迭代循环。A 由 setupPart、test 和 bodyPart 组成，它们标识 LoopNode 中包含的 ExecutableNode 的子集。LoopNode 中的任何 ExecutableNode 都必须包含在 LoopNode 的 setupPart、test 或 bodyPart 中。当 LoopNode 开始执行时，其中的任何 InitialNodes 都会立即启用。但是，包含在 LoopNode 中的 ExecutableNode 只能在执行包含它的 setupPart、test 或 bodyPart 部分时启用。

当一个部分被执行时，该部分中没有强制输入数据且没有传入的 ControlFlow 与同一部分中的源的任何 ExecutableNode 都将被启用并接收单个控制令牌。然后根据活动的通常语义继续执行，除了在未执行的部分中对 ExecutableNode 所做的任何提议都不会立即交付，而是保持挂起状态。如果目标 ExecutableNode 最终作为包含它的部分的稍后执行的一部分执行，则它可以接受任何待处理的提议。

## 条件节点

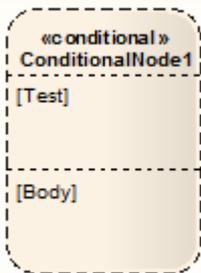
条件结构活动节点是 “If-Then-Else” 编程结构建模等价物。在最简单的情况下，它由一个子句组成，其中包含：

- 评估条件测试分区，以及
- 如果满足测试条件，则执行一个或多个操作体分区

您可以有多个子句，因此如果不满足测试条件，则忽略其体，处理移至下一个子句并评估另一个测试条件。

每个条款都有一个“Decider”行动销来保存测试的结果，以及一个“体输出”行动销来保存体动作的结果（如果执行的话）。行动销节点本身有一个结果移动节点，它提供了节点的整体结果（体要执行的节点的输出）。

条件节点在活动图中的表示类似于：



您可以通过将其他活动图表元素从工具箱页面拖到元素的适当分区中来定义条件节点，并根据需要链接和组织结构。元素在分区的左上角对齐，因此调整节点大小可以保持分区内和分区之间的结构组织。如果您尝试将节点缩小到结构大小以下，则节点会自动默认为“最佳拟合”大小。

创建 Conditional 节点时，将显示“属性”对话框。其中大部分您可以完成任何其他元素一样完成。但是，对于 Conditional 节点，您还可以显示属性窗口，其中有一个额外的“条件”选项卡。

在此选项卡的“结果”面板中，添加一个行动销以保存节点的结果，单击“添加”按钮以显示“选择销”对话框。

A 条件节点自动包含一个包含测试分区和体分区的子句，以及一个销器和体输出销。您可以根据需要添加更多子句。对于每个条款，您为决策者和体输出添加一个行动销。单击保存按钮以保存子句定义。

“选择销”对话框仅显示适合上下文输出如果所需的行动销尚不存在，您可以单击对话框上的“加新”按钮，在相应的父节点下自动创建一个输出销。

对于“结果”和“体输出”条目，您可以通过右键单击条目并选择“在项目中查找浏览器”选项来查看每个行动销的确切位置。

默认情况下，“节点”面板列出了测试分区中包含的行动和活动。单击“体”单选按钮以列出包含在体分区中的元素。元素必须完全包含在要在其中列出的体分区中 - 如果它与测试分区有任何重叠，则将其视为测试分区的一部分。

### 添加或删除子句

要添加另一个条款，请单击“条款”列表下方的“添加”按钮。这会在列表中插入一个新子句，并确定哪个是前面的（前身）子句和（如果合适的话）哪个是后面的（后继）子句。清除“条款”面板中的其余字段，以便您可以添加决策者和体输出行动销。新的“测试”和“体”分区会立即添加到图表上的元素中，您可以使用活动元素填充这些分区，然后在“节点”面板中识别这些元素。

要删除条款，请在列表中突出显示它并单击删除按钮。这会立即从图表中删除子句的相应分区，以及它们包含的所有活动元素。从其他两个子句之间删除一个子句会调整数字顺序；例如，如果从第1条和第3条之间删除第2条，则第3条将重命名为第2条，并且任何其他条款也上移一位。

### OMG UML 规范：

OMG Unified Modeling Language 规范 (v2.5.1, p.478) 指出：

ConditionalNode 是A StructuredActivityNode，它从一些可选的 ExecutableNode 集合中选择一个来执行。ConditionalNode 由A或多个子句组成，每个子句代表条件的一个分支。A子句由一个测试部分和一个主体部分组成，它们标识 ConditionalNode 中包含的 ExecutableNode 的不相交子集。ConditionalNode 中的任何 ExecutableNode 都必须包含在恰好一个子句的测试部分或正文部分中。

# 同步



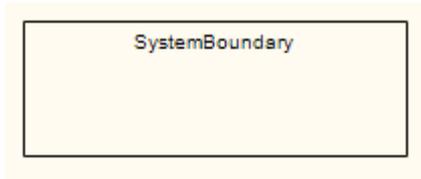
## 描述

同步状态对于指示状态机的并发路径是同步A很有用。它们用于拆分和重新加入并行处理周期。在将路径带入同步状态后，新出现的过渡表明一致。

## 工具箱icon



# 系统边界



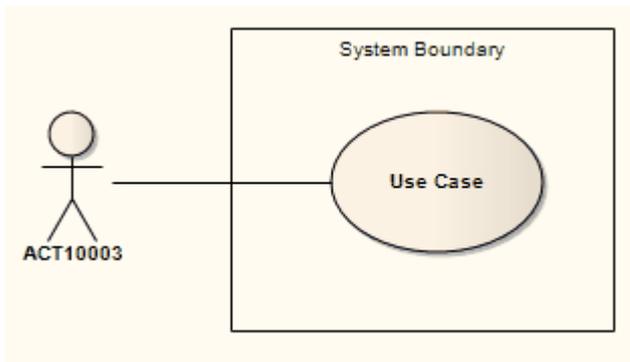
## 描述

系统边界是用于定义概念边界A非元素元素您可以使用系统边界来帮助对逻辑相关的元素进行分组（从视觉角度来看，不是作为UML模型的一部分）。

在 **OMG Unified Modeling Language规范 (v2.5.1)** 中，系统边界在有关使用案例的部分中进行了描述，因为系统边界通常用于指示将用例应用于另一个实体。在这个上下文，系统边界：

- 附上用例，和
- 通过 选择<Item>”对话框 分类器相关联，例如类、部件或子系统（参与者）

通过将系统的边界-而不是用例-与分类器分类器相关联，该器作为用户链接到使用用例，而不是作为所有者。



您还可以将用例定义为系统边界的分类器，以将包含在系统边界中的元素（例如活动图的部分）链接到它们在逻辑用例中的表示。

系统边界的元素属性元素名称、边框样式以及水平或垂直泳道的数量。您还可以更改系统边界的整体形状，其中包括向元素添加分隔线的选项，而不是使用泳道，并且您可以使元素完全不透明、完全透明或介于两者之间的不同程度的不透明度。

可以使用元素的上下文菜单A系统边界元素标记为“可选”。当元素不可选择时，您可以单击系统边界空间中的其他元素，而无需激活或选择系统边界本身。

## 工具箱icon



## 注记

- 系统边界是图像元素A基础，它使您能够将图标或背景添加到图表中，自动显示图像管理器窗口，从中选择适当的图像
- A系统元素边界边界例如，分析图

## OMG UML规范

OMG Unified Modeling Language规范 (v2.5.1, p.641) 指出：

A组用例（有时称为系统边界）的主题可以显示为一个矩形，其名称位于左上角，用例椭圆可视地位于该矩形内。相同的建模用例可能在视觉上被描绘为多个主题矩形内的单独椭圆。

# 系统边界属性

系统边界一个元素有一个小的属性集，主要与元素的外观有关。您还可以应用其他元素控制选项，例如默认外观、锁定元素以及将图像应用于元素。

必须将元素设置为“可选”，以便您能够更改其属性。

## 访问

上下文菜单	右键单击边界元素 属性 属性
键盘快捷键	Alt+Enter
其它	在属性窗口 ( Ctrl+2 ) 中，单击“边界”选项卡

## 属性系统边界

选项	行动
名称	( 可选 ) 类型元素的名称。( 此字段不会出现在属性窗口的“边界”选项卡上 )。
形状	单击下拉箭头并从以下选项中进行选择： <ul style="list-style-type: none"> <li>• '矩形' - 如果您之前已从带有尖角的默认矩形边框切换，请返回该默认值</li> <li>• '圆角矩形' - 将形状设置为圆角矩形</li> <li>• 'Ellipse' - 将形状设置为圆形或椭圆形以容纳封闭的元素</li> <li>• '用户定义 - 正交' - 启用在边框上设置拖动点以创建自定义正交 ( 块 ) 形状 ( 请参阅自定义系统-边界/自由部分 )</li> <li>• '用户 - Freeform' - 启用在边框上设置拖动点以创建自定义自由形状 ( 请参阅边界定义系统-正交/自由部分 )</li> <li>• '用户定义 - 自定义网格' - 启用将垂直和边界添加到边界或一个单元 ( 请参阅自定义系统-自定义边界部分 ) ；这种风格禁止在边界元素中使用泳道</li> </ul>
风格	单击下拉箭头并从以下选项中进行选择： <ul style="list-style-type: none"> <li>• Solid - 带有系统默认元素填充颜色的实线边框</li> <li>• Dotted - 没有元素填充颜色的虚线边框</li> <li>• 虚线 - 没有元素填充颜色的虚线边框</li> <li>• Solid - 无填充 - 没有元素填充颜色的实线边框 ( 此设置会阻止元素的填充不透明度；请参阅填充不透明度部分 )</li> </ul>
水平泳道	类型在您想要将元素划分为的水平段数中，以在水平边界中对系统中的元素进行分组 ( 例如，客户端、应用程序和数据库层可以在泳道中表示 )。 该字段默认为1。如果您打算使用自定义网格，请将其保留在此设置上。 泳道是系统的边界——你不能改变它们的相对高度。

<p>垂直泳道</p>	<p>类型在您想要将元素划分为的垂直段数中，以在垂直边界中对系统中的元素进行分组（例如，开始、进度和终止段）。</p> <p>该字段默认为“1”。如果您打算使用自定义网格，请将其保留在此设置上。</p> <p>泳道是系统的边界——你不能改变它们的相对宽度。</p>
<p>实例分类器</p>	<p>（在属性窗口上。）显示属性的分类器边界如果尚未指定，或者您想更改它，请单击  图标并使用“选择元素”对话框找到并选择所需的分类器。特征窗口也会显示。</p>
<p>多样性</p>	<p>（在属性窗口中。）显示可以存在于集合中的元素实例的数量；如果没有设置图形或您想更改它，请单击下拉箭头并选择适当的值。</p>

## 自定义系统-边界/自由形式

当您在“形状”字段中选择了“用户定义”选项之一时，您可以在系统的边界添加路径点，以向一个方向拖动以创建新形状。这有助于您创建不规则的形状，这些形状包含无法在简单的矩形或椭圆中捕获的分散元素。

“正交”变体可帮助您创建具有垂直和水平线的形状，而“自由形式”变体可帮助您创建对角线。

要在边缘设置航点：

- 在边缘的适当点上按 Shift+单击、Ctrl+单击或 Ctrl+Q

清除航路点：

- 按 Shift+单击、Ctrl+单击或 Ctrl+Q 就可以了

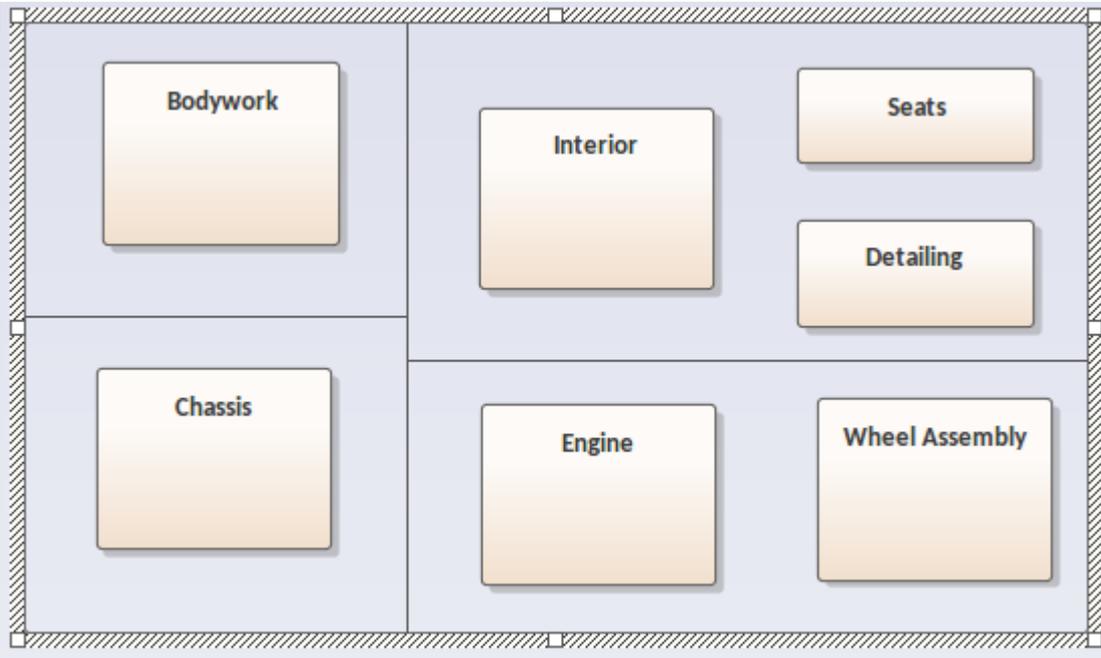
移动航点：

- 点击边界显示所有路点，将光标定位在需要的路点上，使其变为绿色，然后单击并拖动该路点；当您光标移动到边框上时，光标两侧的航点会变为绿色，您可以单击并拖动该边框段

如果您创建“自由形式”形状，然后将“形状”设置更改为“用户定义 - 正交”，系统会将所有对角线转换为垂直或水平线。然后，您可能需要调整形状以减少线条。如果您尝试沿对角线方向拖动正交航点，则水平线和垂直线会调整以在光标位置保持直角。

## 自定义系统边界- Custom Grid

当您选择此选项时，您可以使用与为边界边界设置的样式相同的样式在该区边界绘制水平线和垂直线。请记住，此选项是使用泳道的替代方法 - 您不能同时使用两者选项一起，并且泳道选项被禁用。



自定义线条链接现有线条，所以如果你有一个没有边界的元素，你可以在元素的边界之间画一条线（如插图中的垂直线），但如果你已经添加了单元，你可以画一条线a单元的垂直或水平边界之间（如图中的两条水平线）。这有助于您在本地添加更多非常规的元素边界，以创建诸如出现在业务模型画布中的效果。正如您可以在泳道之间拖动元素一样，您也可以在自定义单元之间拖动元素。

将自定义单元添加元素边界

1. 选择“用户定义 - 自定义网格”选项。
2. 返回图表。
3. 按住 Shift 键并按住鼠标左键，同时将光标拖过或向上边界元素。将显示A虚线指引线以显示将在何处创建线。
4. 当线条大致位于你想要的位置时，松开鼠标按钮。线变成实线。

您可以通过将光标悬停在线条上来移动线条，以便显示一个小的双头箭头，然后按住鼠标左键并根据需要移动鼠标。邻接线将延伸或缩回以保持连接。

要删除一条线，请右键单击它并选择“删除区域”上下文菜单选项。这与单元表中的合并非常相似。系统突出显示该线所界定的区域并提示您确认删除。单击是按钮。任何邻接线都将延伸到下一条垂直线。

如果您调整边界元素的大小，所有单元都会按比例调整大小。但是，如果您只是想为添加更多单元创建更多空间而不一定要更改单元内部的边界，请在调整元素大小时按住 Shift 键。只有那些连接到边界的线被延长；内部单元保持其大小。

## 填充不透明度

虽然某个边界通常包含许多其他元素，但可能有隐藏这些元素的原因以及完全显示它们的时间，或者可能只是表明它们在那里，这取决于图表的直接目的。您可以通过更改元素边界。

在设置不透明度之前，请检查元素是否具有填充颜色，并且“属性”对话框或属性“边界边界”选项是否设置为“实体 - 无填充”以外的值。

您可以使用以下两个弹出式元素工具栏中的图标设置不透明度：

- 单击元素边界  图标：



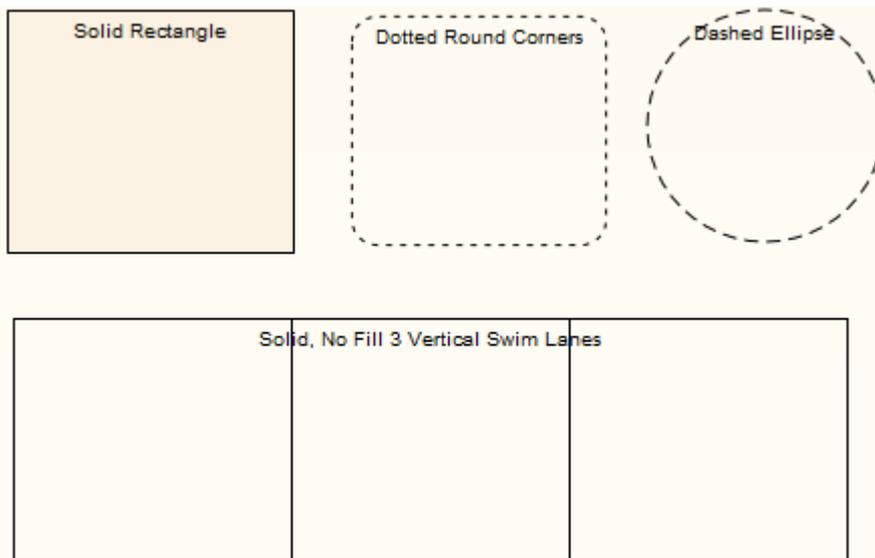
- 右键单击该边界并查看元素菜单上下文：

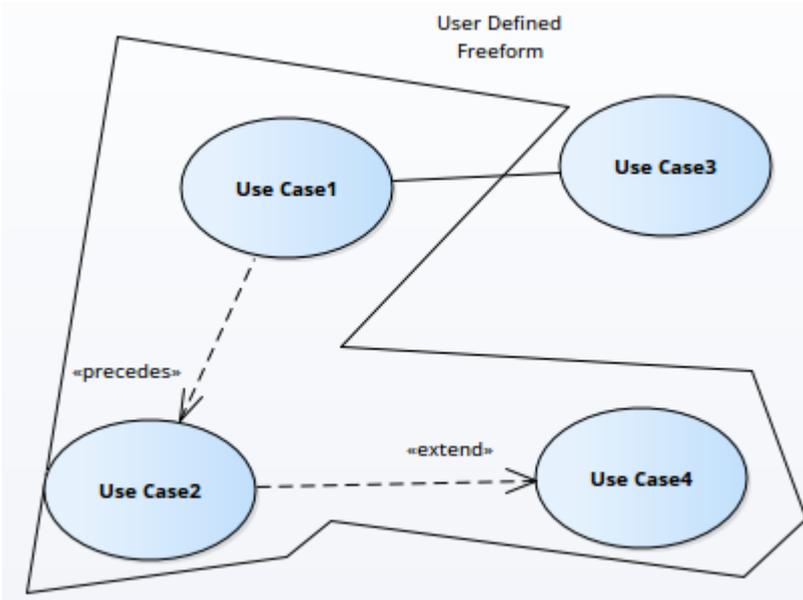
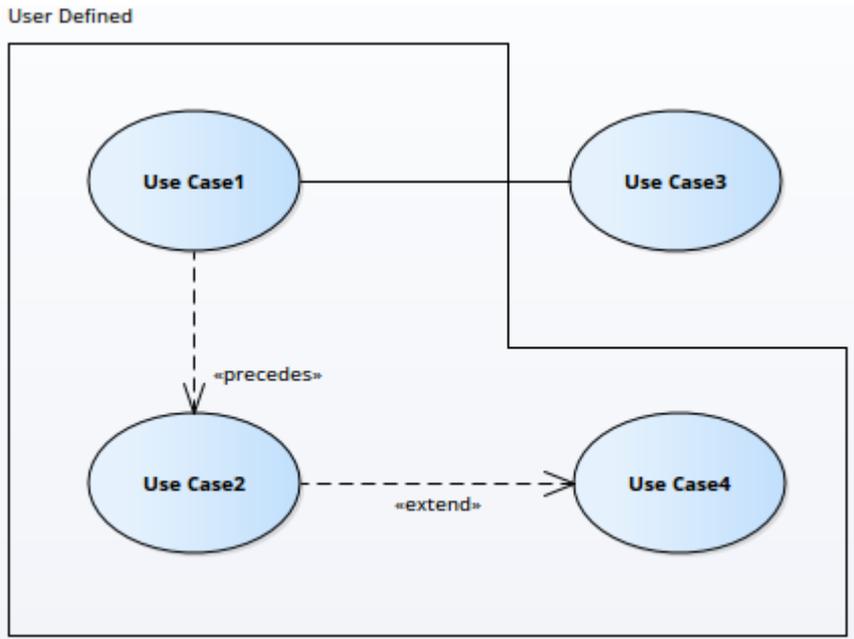


单击  图标并选择：

- 100% 的总不透明度，隐藏在后面和重叠或被边界覆盖的元素（您可以右键单击单个元素并选择 **Z-Order | 置顶** 选项以仅显示这些元素）
- 0% 表示无不透明度，其中不应用填充颜色，并且任何东西后面的边界都是完全可见的
- 75%、50% 或 25% 以设置适当的不透明度并使被覆盖的元素可见但过度着色

### 示例形状





### 注记

- 如果图表被锁定或用户没有更新图表的访问权限，则边界的图表特定选项（形状、边框样式、泳道计数）将被锁定；名称"字段可以更新
- 如果元素元素边界名称"字段；其他字段可以更改

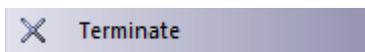
## 终止



### 描述

终止pseudostate表示一旦进入其pseudostate，状态机的执行就结束。

### 工具箱icon



# 触发器



## 描述

A 触发器表示一个事件，它启动了一个动作（并且可能来自于前一个动作的完成）。您最初以以下四种方式之一定义触发器：

- 作为一个转移的属性转移关系
- 作为接受行动的一个属性（在元素事件 属性”对话框的 触发器”选项卡上）
- 作为状态机库表中的事件
- 直接，作为一个触发器的元素，通过 新元素”对话框或图表工具箱 状态附加”页面）

当您保存触发器时，它会添加到浏览器窗口中父包的元素列表中。然后，您可以单击它并按 Ctrl+2 以显示元素的属性窗口，以查看并在需要时将其属性为一个而不是一个属性本身。作为触发器创建的简单事件保持为元素，而以其他触发器创建的是触发器的元素，在属性窗口中有一个 触发器”选项卡。

字段	行动
类型	如有必要，编辑触发器类型： <ul style="list-style-type: none"> <li>• 调用-指定事件为调用，通过调用操作向关联object发送消息</li> <li>• 更改-指定事件是一个 ChangeEvent，这表明转换是属性值更改的结果</li> <li>• 信号指定事件为SignalEvent，对应接收到一个异步信号实例</li> <li>• 时间-对应于 TimeEvent；它指定了一个时刻</li> </ul>
规格	输入引发简单的触发器，或单击  按钮并选择事件（取决于类型值）。
端口	单击添加按钮并从 选择端口”对话框中选择适当的端口。 <ul style="list-style-type: none"> <li>• 要使用 选择端口”对话框创建新的端口，触发器端口创建为类或部件的子元素</li> <li>• 要一次添加多个端口，请在选择每个端口时按 Ctrl</li> <li>• 要检查端口的确切位置，请右键单击端口名称选项并选择 在项目中查找浏览器”</li> </ul>

## 注记

- 您也可以将现有触发器元素拖到另一个图表上，尽管在那个时间元素的用途上下文
- 这个元素和简单操作不同，触发器操作是数据库中数据修改后自动执行的操作

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.300) 指出：

事件可能导致行为的执行（例如状态机中转换的效果活动的执行）。触发器指定可能触发行为执行A事件以及对事件的任何约束以过滤掉不感兴趣的事件。

# 用例

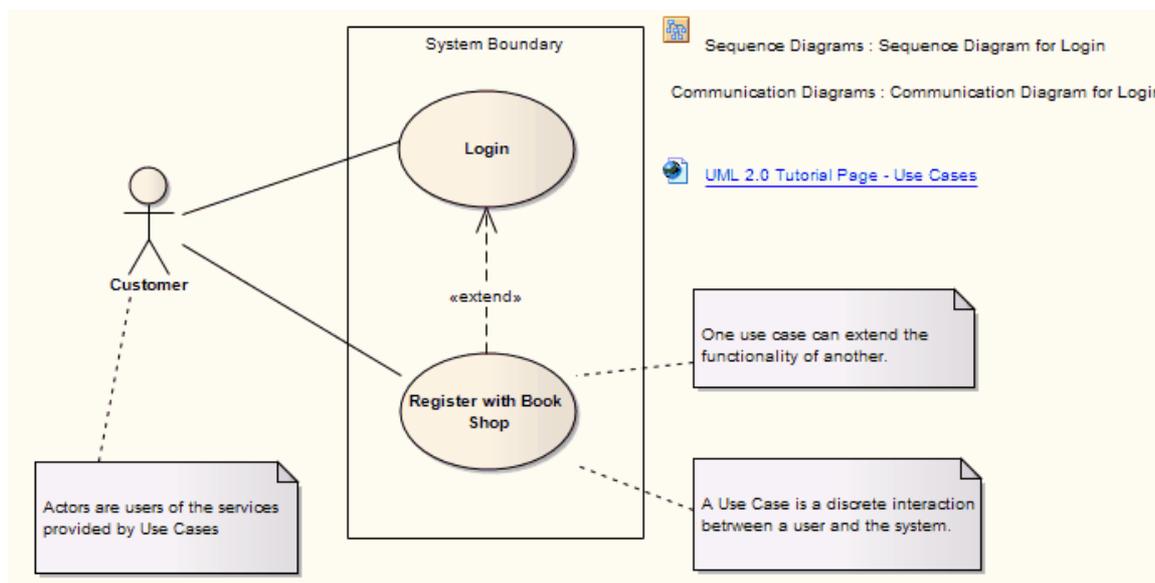


## 描述

用例是UML建模元素，它描述了建议系统的用户如何与系统交互以执行离散的工作单元。它描述并表示随着时间的推移对最终用户（人、机器或其他系统）有意义的单个交互，并且需要使系统处于完成状态：交互完成或回滚到初始状态。▲用例：

- 通常具有描述其运行的基本特征和规则的要求和约束
- 可以有一个关联的序列图来说明一段时间内的行为；谁对谁做什么，什么时候做
- 通常有与之相关的场景，描述随着时间的推移产生最终结果的工作流程；还启用了替代工作流（例如，捕获异常）

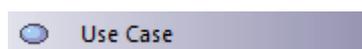
## 示例用例



如果扩展一个用例，你可以用用例扩展点来指定扩展点。要在图表上显示用例的属性、操作或约束，请使用矩形表示法。

Enterprise Architect还提供了两个典型的使用案例：测试用例和业务用例。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.649) 指出：

A用例指定了由其主体执行的一组动作，这会产生一个对每个主体的一个或多个参与者或其他利益相关者有价值的可观察结果。

# 用例扩展点

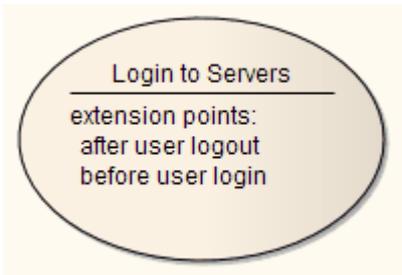
为一个用例定义的行为可以添加到另一个用例的行为；也就是说，第一个使用案例扩展了第二个。这在模型上由从第一个用例到第二个的扩展连接器表示。如果扩展行为在特定点生效，您可以将该点定义为扩展用例上的扩展点。扩展点的名称（描述）文本可以是非正式的或精确的，以适合定义扩展应用的行为点。A用例可以有多个扩展点，以允许不同的源使用案例扩展此目标用例，或者根据为扩展连接器定义的约束更改扩展行为的应用位置。连接器还标识有效的扩展点。

## 访问

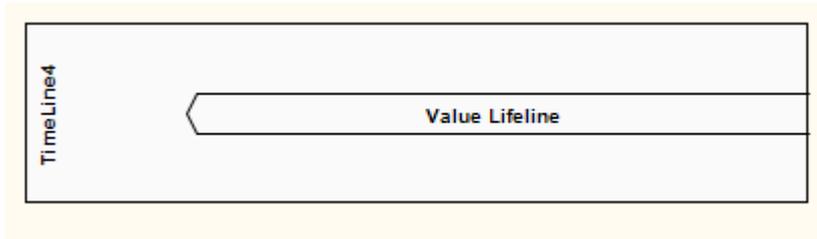
上下文菜单	上图   右键单击扩展用例元素   高级   编辑扩展点
-------	------------------------------

## 为一个用例添加扩展点

字段/按钮	行动
定义扩展点	列出当前为所选用用例定义的扩展点。
添加	单击此按钮可显示新扩展点名称的提示。 类型名称并单击确定按钮。该名称将添加到定义扩展点列表中。
编辑	单击现有扩展点并单击此按钮以显示更改所选扩展点名称的提示。 改写名称并单击确定按钮。名称在定义扩展点列表中更新。
消除	单击现有扩展点并单击此按钮可立即从定义扩展点列表中删除该名称。
确定	单击此按钮可保存对扩展点的所有更改，并关闭对话框。 您定义的扩展点在图中的用例元素上表示，如图所示。



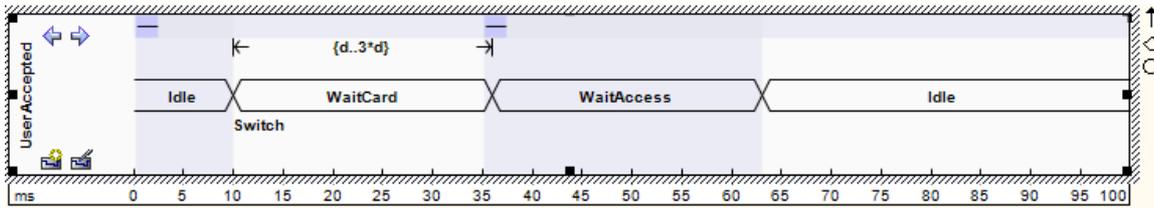
# 价值生命线



## 描述

生命线是object在一定时间内所走A路径，由 x 轴表示。有两种：值生命线（在此处定义）和状态生命线，均用于时序图中。

Value生命线A整个图表中显示生命线的状态，平行线表示稳定状态。线之间A交叉表示状态的过渡或状态。这是价值生命线的例子：



请参阅 OMG Unified Modeling Language规范 · ( 1 · 图 14.30 · p.520 。 )

## 转移点属性

A价值生命线由一组过渡点组成。可以使用以下属性定义每个转换点：

属性	描述
当时	指定状态更改的开始时间。
转移至	表示状态要改变的生命线。
事件	描述正在发生的事件。
时间限制	指状态在一条状态内发生变化所用的生命线，或消息传递所用的时间。
时间观察	提供有关状态更改或发送消息的时间的信息。
持续时间限制	与生命线在特定状态下的周期有关。约束可以由生命线内的状态变化或生命线收到消息引发。
持续时间观察	表示某一特定状态下生命线的间隔，从状态或消息接收开始。

## 示例属性

在示例图中，10ms 转换点具有以下属性：

属性	文本
在时间	10ms
转移至	等待卡
事件	转变
时间限制	—
时间观察	—
持续时间限制	d..3*d
持续时间观察	—

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.604) 指出：

将可连接元素的值显示为时间函数。值明确表示为文本。交叉反映了值发生变化的事件。

## 结构图表元素

本节详细描述了在Enterprise Architect中使用UML Structural图表建模时常用的元素。

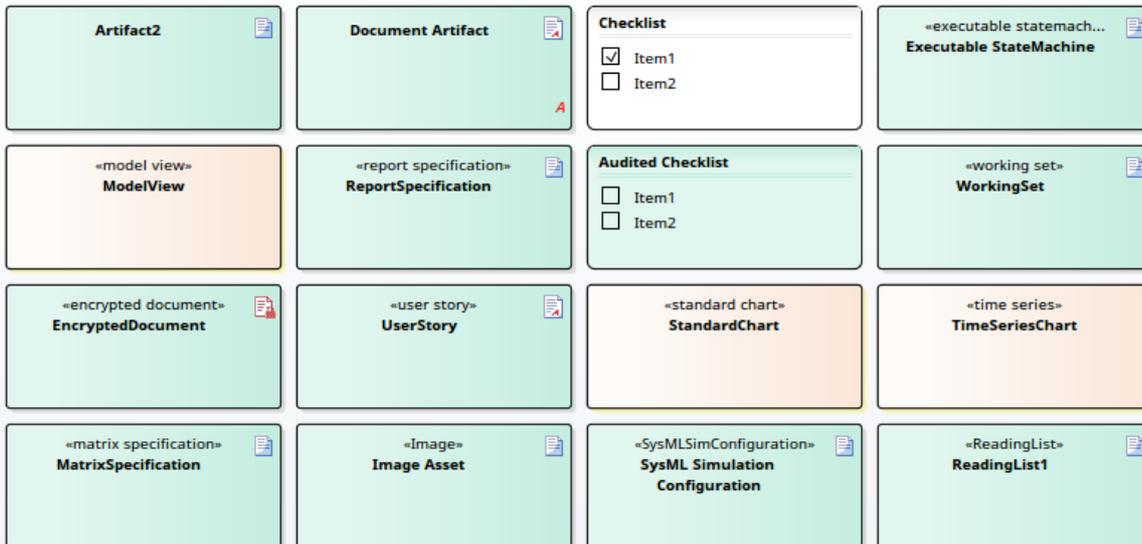
# 工件

工件是系统使用或产生的任何物理信息。在Enterprise Architect工件中，一个工件是由一个元素组成的，它可以有一个数量的内部工件系统代表一个特定的目的，包括在示例中显示的模型中的相关属性和结构。操作，并且可以根据它们所代表的object实例化或与其他工件相关联。

您可以通过从图表工具箱元素中工件一个工件图标来创建一个工具。公共元素”页面的通用工具工件”图标上，您可以选择一种类型的工具箱-当您工件要创建的图表类型时。

- ▷ Documents
- ▷ Governance
- ▷ Dynamic Viewpoints
- ▷ Charts
- ▷ Simulation

## 类型的工件



类型	描述
( 基地 ) 工件	<p>基础源德，在开发过程中使用A外部工件，以及开发过程中产生的内部工件，如模型、数据库库表、数据库可交付文件等。属性代表的工件列在元素窗口的“文件”选项卡上。</p> <p>要打开由元素代表的工件，请单击图表上的元素并按 Ctrl+E。每个文件都可以在图表视图工作区的单独选项卡上打开（如果文件可以在Enterprise Architect中打开）或在文件类型的默认窗口查看器/编辑器中打开（如果文件不能在Enterprise Architect中打开）。</p> <p>文件也可以从“文件”选项卡（在窗口默认编辑器中打开）单独启动，对于具有关联文件的任何其他类型的元素。</p>
文档工件	<p>文档工件是A工件“文档”的立体类型。您使用工件或部件创建、文档工件工具或工具箱，将其与图表文档的部署页面相关联。</p> <p>双击元素以显示链接文档编辑器。当您创建文档工件元素时，在图表的右下角会显示一个“A”符号。</p>

	
<p>检查清单工件</p>	<p>A检查清单，每个工件都有一个复选框，用作功能的功能您可以将每个复选框设置为默认选中或未选中，并将所有选中的选项设置为灰色或删除，以便未选中的选项更加突出。然后，最终用户可以使用元素来勾选已获得的项目或已完成的活动。</p> <p>A版本的工件-需求检查清单可在 扩展“需求工具箱中找到。这会包含良好需求的十个特征，供需求作者在检查他们的需求是否已设置为显示这些特征时勾选。</p> <div data-bbox="518 582 1173 878" style="border: 1px solid black; padding: 5px;"> <p><b>Stock Replenishment</b></p> <ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Check Stock in Storage</li> <li><input checked="" type="checkbox"/> Check Stock in Delivery</li> <li><input type="checkbox"/> Check Stock on Current Order</li> <li><input type="checkbox"/> Check Catalog for Minimum and Maximum Stock Holding</li> <li><input type="checkbox"/> Is more stock required?</li> <li><input type="checkbox"/> If stock required, draw up new Order</li> </ul> </div>
<p>审核清单工件</p>	<p>任何工件都与检查清单工件，只是它具有关联的 审核日志”页面，该页面标识了对清单进行的更改、更改的人员以及更改的内容。这是一个非常有用的项目管理工具，为清单的使用添加了跟踪和问责制。</p>
<p>加密文档工件</p>	<p>加密文档工件用于创建和保存自动加密的链接文档，并且在没有输入密码的情况下无法在Enterprise Architect中打开和自动解密。因此，您可以使用生成的工件来记录敏感信息，您可以通过分配密码来保护元素信息不被一般访问。</p> <p>当您从工具箱的 文档”页面将 加密文档”图标拖到图表上时，会显示一个提示以输入密码。输入此密码后，您可以创建链接文档。此后，当任何用户尝试打开文档时，都会显示相同的密码提示。如果用户没有提供您最初指定的密码，文档将无法打开，而如果提供了正确的密码，文档将被解密并打开，并且可以查看和编辑文本。</p> <p>您不能更改密码，也不能从链接中删除链接的工件（尽管您可以删除链接的工件本身）</p> <p>其它 功能”链接文档的其他功能 - 例如文档窗口或报告生成器 - 将忽略加密文档。</p> <p>加密文档在工具箱页面、图表和浏览器窗口中由红色 文档”图标  指示。</p>
<p>用户故事</p>	<p>A用户故事（文档中的工件文档）是一种使用上下文方法的方法，例如极端程序用例。在链接文档中，您定义了业务系统必须提供的功能；它以简单、简洁的格式捕获需求的 谁”、 什么”和 为什么”。用户故事工件文档工件的基础，提示您选择链接文档的模板</p>
<p>工作工件</p>	<p>A工作集，工件定义经常使用的工作集并打开各种窗口、图表，重新创建一个您可以使用的工作环境。</p> <ul style="list-style-type: none"> <li>• 要创建或修改工作集，请右键单击元素并选择 编辑工作集”选项</li> <li>• 要执行工作集以打开定义的窗口和视图并执行任何命令，请双击元素</li> </ul>
<p>定制库表工件</p>	<p>A定义库表工件生成的图表object以类似于电子表格的格式显示图表，在适用的地方提供有关元素、图表或项目管理的额外 非模型”信息。</p>

<p>标准图表</p>	<p>用于在您的模型中生成图表或图表方面A标准图表工件提供功能。它在元素属性"对话框的标准选项卡中添加了三个 图表详细信息"选项卡。</p> <p>将元素添加到图表后，双击它。元素的 属性"对话框会在 图表详细信息 -源"选项卡中自动打开。定义图表类型和数据源，然后继续定义您想要应用的任何过滤器，以及图表的显示方式。</p> <p>定义图表后，只要您打开父图表，它就会自动显示最新信息。</p>
<p>时间系列图表</p>	<p>时间序列图表工件A了属性随时间模型的线性图的功能。</p> <p>将元素添加到图表后，双击它。元素的 属性"对话框会在 图表详细信息 -源"选项卡中自动打开。定义要从中提取数据的包，以及对数据进行采样的时间间隔。然后继续定义图表的外观。</p> <p>定义图表后，只要您打开父图表，它就会自动显示最新信息。</p>
<p>模型视图</p>	<p>A模型视图工件了一个表格模型视图图表功能#</p> <p>使用您的模型中的数据片段，提取的自定义模型#</p> <p>搜索。</p> <p>将元素添加到图表后，双击它。元素的 属性"对话框会在 图表详细信息 -源"选项卡中自动打开。定义#</p> <p>搜索以提取和制表信息。</p> <p>定义图表后，只要您打开父图表，它就会自动显示最新信息。</p>
<p>报告规范</p>	<p>A工件报告定义。在图上创建好元素后，双击它，弹出 生成文档"对话框，在该对话框中输入报表参数并生成报表。</p> <p>创建报表规范后，每次工件元素的报表参数对话框，再次生成文档报表参数。您可以继续生成相同的报告，或在必要时更改参数。如果您更改参数，它们会重新显示，直到您再次更改它们。</p>
<p>矩阵规格</p>	<p>A矩阵规格工件关系矩阵配置文件在图表上创建元素后，双击它以显示 矩阵规范"对话框，在该对话框中创建配置文件定义。配置文件采用元素的名称。包中定义的工件文件独立于包含工件元素源和目标的包，因此可以指定除母包之外的对象。</p> <p>创建配置文件定义后，双击工件元素配置文件的的关系矩阵将显示应用的配置文件。</p> <p>要编辑配置文件，请右键单击工件文件并选择 文档  编辑矩阵配置文件'选项。</p>
<p>可执行状态机</p>	<p>- 一个可执行状态机工件生成、编译和执行状态机状态机的复杂代码。</p> <p>每个状态机都是类元素的子元素；当你从浏览器工件类元素上时，它就像一个部件一样被粘贴在浏览器工件。您可以将 - 并因此将多个类 - 粘贴到一个单独的工件中。</p> <p>设置好状态机可执行状态机，工件使用简单上下文选项来工件和执行执行代码生成中的所有执行工件。</p>
<p>业务流程仿真</p>	<p>在您的系统上安装并注册了业务流程BPSim 仿真引擎工具箱时， 业务流程仿真引擎仿真工具"页面工件出现。您使用工件业务流程仿真和作为容器的访问模型来模拟仿真配置，它定义了模拟过程中应用什么参数。</p> <p>右键单击元素并选择 配置 BPSim "上下文菜单选项。将显示配置 BPSim窗口。</p>
<p>图表结果图表</p>	<p>当图表BP的 工具"页面中安装并注册了您的系统上的图表自定义结果图表引擎和工具箱应用程序时，工件自定义结果图表BPSim 仿真引擎和图表工具箱工具已安装并注册。从反映工件模拟的选定结果生成图表：</p>

	<ul style="list-style-type: none"> <li>• 图表结果图表 - 生成一个图表，反映从一系列标准图表模拟中选择的结果</li> <li>• 图表自定义结果图表 - 生成反映一系列自定义图表模拟结果的图表</li> </ul> <p>至于其他维度的 Bar 工件，无论是图表图表都可以快速显示仿真结果，在 Line 图表变体中，二维 Bar 图表或 3 图表。在工件业务流程仿真中自动生成的结果中捕获了这两种工件运营的结果。</p>
<p>配置</p>	<p>此配置演示配置了对工件仿真窗口的访问，并包含特定的 SysML 仿真窗口。要访问配置仿真窗口，请双击配置配置或右键单击并选择 显示工件”选项。</p>
<p>图像资产</p>	<p>图像资产是用于在模型中存储图像的模型元素。您可以通过将 图像资产”图标拖到图表上并选择图像文件来创建它们，或者将图像文件从文件系统直接拖到图表上。Enterprise Architect 图像资产工件创建一个从文件中工件的图像，然后存储在该图像中。元素可以在图表上显示为图像或矩形元素；要在它们之间切换，请右键单击 object 并单击 显示拥有的图像”选项。</p> <p>当工件为图像时，您可以双击它以显示元素的属性对话框。如果工件- 将打开以元素格式显示的图像，则在默认图像查看器中的图像上双击它。</p> <p>通过插入指向图像资产模型图像元素的超链接，可以在文档中使用图像资产图像，就像使用来自图像管理器的图像一样。创建超链接时，系统会提示您选择要链接的 object 类型；单击 无素图像”并从列表中选择适当的图像资产元素。如果按住 Ctrl 并单击超链接，则图像将显示在图像的默认外部查看器中。生成报告时，使用其 拥有的图像”呈现超链接的图像资产元素。</p>
<p>阅读清单</p>	<p>阅读列表工件提供包含特定重要性信息的任务或过程的列表。目的是让每个元素的信息显示在文档窗口中，按照序列在阅读列表中的组织顺序。文档窗口提供了 下一个”和 上一个”选项来移动元素。</p>

# 工件文件

A文件工件元素工件一个外部文件相关联。您可以从文件本身元素工件

## 创造工件

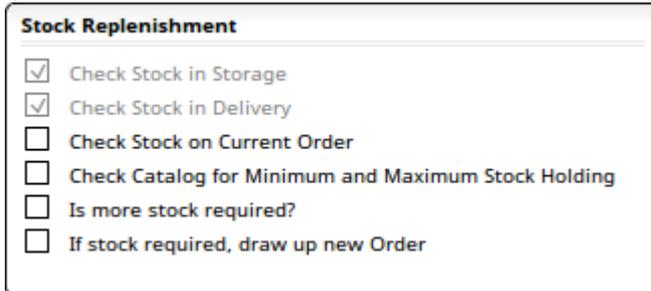
节	行动
1	在文件列表 ( 例如窗口 ) 或桌面上找到该文件。将文件拖到图表上。 A上下文菜单。
2	单击您需要的菜单选项： <ul style="list-style-type: none"> <li>• '超链接' - 在图表上创建一个超链接元素；您可以选择一个子选项来定义用户在双击超链接时是仅显示文件内容还是在适当的文件编辑器中打开它</li> <li>• '工件' - 在元素上创建工作件显示元素 属性"对话框，您可以在其中输入您需要的任何元素属性保存输入的数据并关闭 属性"对话框 ( 注记文件名变成元素名 ) 如果你双击工件 属性"对话框重新显示；单击 文件"选项卡以查看 文件"面板中列出的文件路径名，您可以从中在其注册的应用程序中启动它</li> <li>• '在工件中以元素工件名元素 该文件存储在您的模型中，但由已注册的文件类型的外部应用程序管理；如果您双击工件，文件将在其外部应用程序中打开 如果文件被更改，系统会提示您更新模型中的元素- 单击保存按钮更新元素，或单击丢弃按钮忽略更改</li> <li>• 'Insert' - ( 图形文件 ) 将文件作为边界元素插入到图表中；双击图像以边界 属性"对话框</li> </ul>

## 注记

- 此特征在Enterprise Architect的所有版本中都可用
- 可以使用类型为 无素图像"的超链接创建存储为工件的图像的超链接
- 对于窗口以外的操作系统；作为将文件拖到图表上的文件源，请使用：功能区>设置>用户工具>窗口浏览器。

## 使用检查清单和审核清单工件

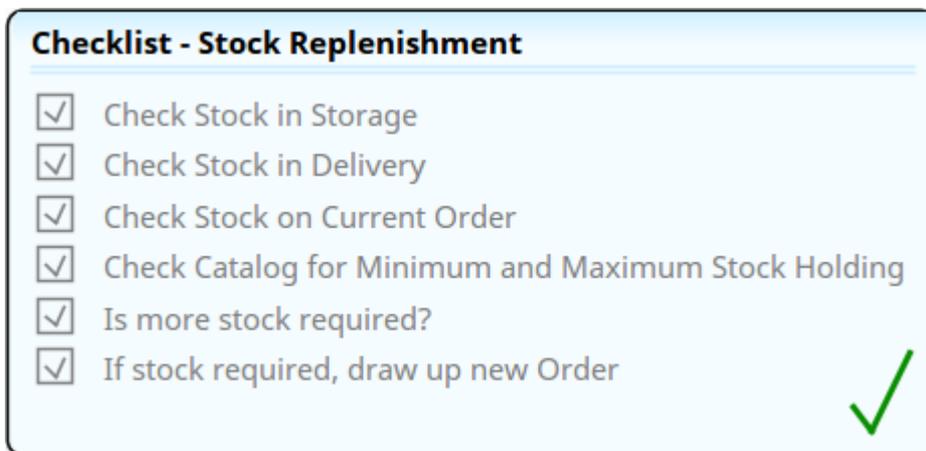
使用检查清单的任何工件，您可以创建检查清单元素和其他用户可以完成的任务，以确保在完成该object时已经解决了开发任务或object所需的方面。您可以针对每个项目选择一个复选框以指示该点已得到解决，并且您可以将清单配置为以灰色显示选定的项目或删除，以便未选定的项目更加明显。



您还可以在平面列表或编号或项目符号列表中显示项目，如果您在处理列表时不一定要检查项目，这会有所帮助。如果确实需要选择此类项目，您可以通过“检查清单项”对话框进行选择。

您也可以重新使用复选框 - 上下文中单个元素、图表中的多个元素以及图表本身的时间菜单都具有清除复选框的选项，以便可以再次使用检查表来重新启动流程。

当一个检查清单上的所有项目都被选中后，该元素的右下角会显示一个绿色箭头，表示检查清单已经完成。



审核清单是审核检查清单的工件，有一个关联的“审核记录”页面。它是项目管理的有用工具，提供功能责任和跟踪任务完成的功能。

您可以有一个必须按序列完成的清单的序列- 必须完成完成顺序中的第一个检查清单，然后才能启用下一个检查清单上的任何复选框以供选择。这个完成顺序是通过在检查清单元素之间创建依赖连接器来建立的。

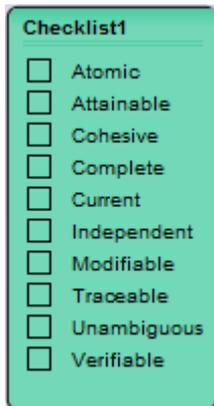
- Dependency 的目标检查清单元素成为先决条件，源检查清单成为依赖项
- 必须先选中先决条件检查清单中的所有项目，然后才能选择依赖检查清单中的任何项目
- 先决条件检查清单选择项目A而依赖检查清单已选择项目
- 清除一个先决条件检查清单也会清除任何依赖的清单（因为他们不能在先决条件检查清单被清除的情况下选择项目）
- 可以将新项目添加到先决条件检查清单中，但前提是它们在保存前设置为选中（对于第一个新项目会显示警告消息；此后新项目会自动设置为选中）
- 清除检查清单及其依赖项将影响层次结构中的所有检查清单，即使它们在当前包或图表之外

### 创建一个元素检查清单

您可以通过从工具箱的“治理”页面将“检查清单”图标图表图表上来创建一个空的检查清单元素。“治理”页面始终存在于每个工具箱的底部。



您还可以通过在工具箱的“扩展功能需求”页面中选择“需求检查清单”图标来创建即用型需求检查清单，方法是选择“需求检查图表”功能（在工具箱标题并指定“需求”）。



## 设置检查清单

要使用项目填充检查清单元素，请双击图表中的检查清单元素。将显示“检查清单项目”对话框，光标位于对话框顶部的字段中，准备接收第一个项目名称。

只需输入项目名称，然后按 **Enter** 键即可显示并移至下一个项目输入字段。

每个项目的左侧是一个复选框。如果您希望在使用元素时默认选中该项目，请单击此处的复选框。

对话框底部是三个字段：

- 显示 - 接受默认的“<无>”作为简单列表，或者单击下拉箭头并从复选框、项目符号或数字中选择要遵循的清单项目
- 选择样式 - 单击下拉箭头并选择是否对选中**复选框**（'<None>'）的项目不应用任何样式，或者显示带有一条线穿过它们的“选中”项目（“删除线”）或浅灰色（“灰色”）
- 间距 - 要么接受项目之间的单个间距的默认值，要么单击下拉箭头并选择项目之间更大的间距；项目内的行距保持为单个间距  
您还可以使用元素上下文列表上的“检查清单Spacing”选项更改间距

完成检查清单设置后，单击关闭按钮。

## 编辑检查清单

要更改检查清单元素中的文本、选择状态或序列顺序，请双击元素以显示“检查清单项目”对话框，然后单击要更改的项目。至：

- 更改项目的文本，只需再次单击并输入当前文本（或右键单击并选择“编辑项目”选项）
- 删除项目，右键单击它并选择“删除项目”选项
- 切换复选框的状态，单击复选框（或右键单击并选择“切换项目检查”选项）
- 将项目移动到项目序列中的不同位置，右键单击并选择“上移项目”或“下移项目”选项
- 将显示格式更改为简单列表、复选框、项目符号或数字，单击“显示”字段中的下拉箭头并选择适当的选项

- 更改清单中所有项目的选定项目样式，单击“选择样式”下拉箭头并选择适当的选项
  - 更改项目间距，单击“间距”字段中的下拉箭头并选择适当的行间距
- 单击关闭按钮以保存更改并关闭对话框。

## 让用户可以使用检查清单

要使其他用户可以使用检查清单，请在用户可以访问的图表上创建它。或者，将图表保存为用户可以绘制的模式，作为创建自己的检查清单图表的基础。

## 使用检查清单

要在带有复选框的检查清单元素上记录操作的完成或object的存在，只需单击相应的复选框以选中它。

如果您需要清除一个或多个复选框，您可以：

- 再次单击每个复选框，或
- 右键单击检查清单元素并选择“清理清单”选项（该元素中的所有复选框），或
- 选择多个检查清单元素，右键单击其中一个并选择“清理清单”选项（所有选定元素中的所有复选框），或
- 右键单击图表背景并选择“清除所有清单”选项（图表上所有清单中的所有复选框）

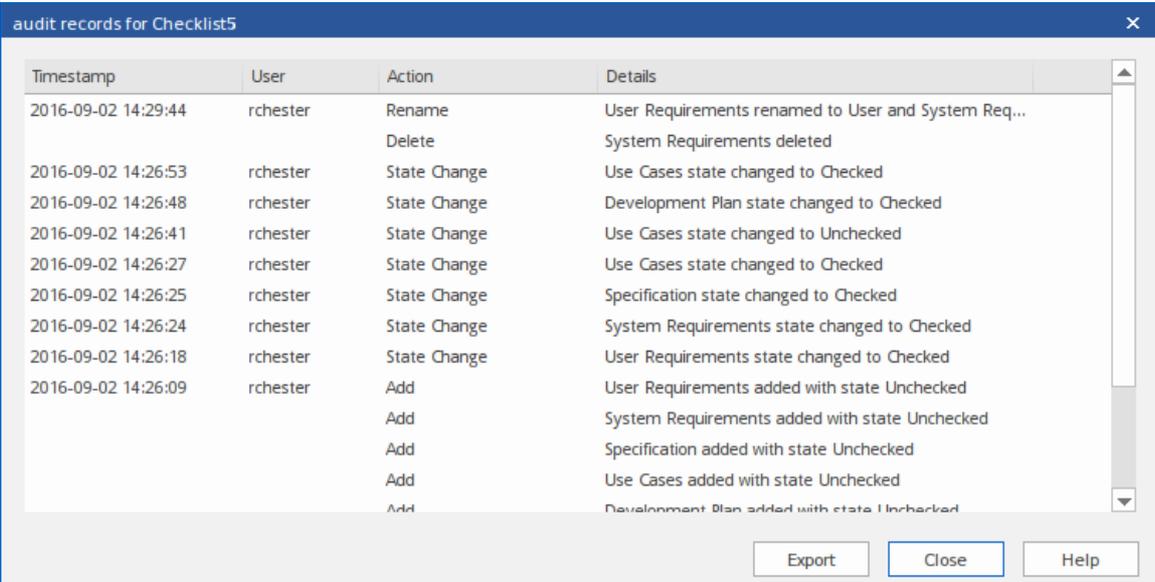
您可以将清除的检查清单重新用于另一个步骤、阶段或过程。

如果您已选择显示简单、项目符号或编号列表，并且您想暂时删除列表中的项目，您可以在“检查清单”对话框中通过选中每个项目对应的复选框来执行此操作。

您还可以使用元素检查清单上下文来更改您正在查看的项目的分隔；右键单击元素并选择您喜欢的行距。

## 审核清单

您可以设置审核清单，其他工作人员可以使用它，两者的方式与审核检查清单完全相同。但是，还有一个额外的上下文菜单选项 - “视图log” - 显示审核清单的“审核记录”页面。此“审核记录”页面仅为审核检查列表生成，是使用“工件记录”工具箱页面创建的。



Timestamp	User	Action	Details
2016-09-02 14:29:44	rchester	Rename	User Requirements renamed to User and System Req...
		Delete	System Requirements deleted
2016-09-02 14:26:53	rchester	State Change	Use Cases state changed to Checked
2016-09-02 14:26:48	rchester	State Change	Development Plan state changed to Checked
2016-09-02 14:26:41	rchester	State Change	Use Cases state changed to Unchecked
2016-09-02 14:26:27	rchester	State Change	Use Cases state changed to Checked
2016-09-02 14:26:25	rchester	State Change	Specification state changed to Checked
2016-09-02 14:26:24	rchester	State Change	System Requirements state changed to Checked
2016-09-02 14:26:18	rchester	State Change	User Requirements state changed to Checked
2016-09-02 14:26:09	rchester	Add	User Requirements added with state Unchecked
		Add	System Requirements added with state Unchecked
		Add	Specification added with state Unchecked
		Add	Use Cases added with state Unchecked
		Add	Development Plan added with state Unchecked

此页面显示对清单所做的每项更改、更改的日期和时间、更改的人员、更改的类型以及更改的内容。如图所示，这些更改包括在创建或更新清单时所做的更改以及用户选择或清除复选框所做的更改，包括：

- 添加检查清单项目（其中也记录了它们被设置的状态）
- 删除项目
- 重新设置每个项目的状态
- 完成一个检查清单
- 取消选择已完成的检查清单上的项目
- 使一个检查清单成为另一个检查清单的先决条件或依赖于另一个检查清单
- 重命名先决条件或依赖检查清单
- 使一个检查清单不再是另一个检查清单的先决条件或依赖项

同时提交的修改（例如当您单击确定按钮时）具有一个时间戳，如审核log图像中的“添加”更改所示。

如有必要，您可以将“审核记录”页面的全部内容导出到 .csv 文件。去做这个：

1. 点击导出按钮；'导出列表'对话框显示。
2. 浏览并选择存储文件的适当目录。
3. 输入文件的文件名，然后单击“保存”按钮。

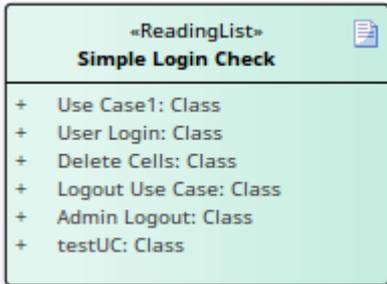
该文件被保存，然后可以在任何外部电子表格应用程序中打开，例如 Excel。

## 注记

- 与许多其他工件属性的对话框不同，可以在元素中设置一个优先级为中或低的清单；这有助于确保流程中的检查清单被标记为重要，例如在看板图中

# 使用阅读工件

阅读列表工件提供包含特定重要性信息的任务或过程的列表。目的是让每个元素的信息显示在“动态文档”视图中，其中序列在阅读列表中的组织顺序。“动态文档”视图提供“下一个”和“上一个”选项来移动元素。



## 设置阅读列表

任务	行动
创建工作件	将“阅读列表”图标从图形工具箱的“文档”页面图表工具箱上。双击元素以显示“属性”对话框并为元素命名；也许确定阅读清单的主题。
创建阅读列表	<p>在浏览器中，识别包含所需信息的每个元素到阅读列表元素并工件元素。元素被显示在工件序列中，它们被添加到工件中。</p> <p>元素是属性名的工件。如果要更改列出它们的序列：</p> <ol style="list-style-type: none"> <li>1. 单击此工件并按 Ctrl+5 特征窗口显示，显示“属性”页面。</li> <li>2. 右键单击要移动的属性，然后选择“上移”选项或“下移”选项。每次单击，该属性都会在列表中向上或向下移动一个空格。</li> </ol>

## 阅读阅读清单

要阅读由阅读列表图标标识的信息，您可以：

- 右键单击工件并选择“显示阅读列表”选项以显示“动态文档”视图，或
- 显示“动态文档”视图（发布 > 模型报告 > 预览模式）并单击浏览器窗口中的元素名称

阅读列表中第一个元素的信息显示在“动态文档”视图中。阅读完此信息后，单击 [阅读] 标题并选择“下一个”选项以显示阅读列表中下一个元素的信息。当您在列表中向下工作时，您还可以选择“上一个”选项以返回列表中更靠前的元素的信息。

“动态文档”视图提供了其他选项，例如，使用其他样式模板来显示信息，或更改显示大小。这些选项在您审阅阅读列表时可用。

# 文档工件



## 描述

文档工件是A工件“文档”的立体类型。您使用图表工具的公共、文档、部件、文档或部署页面图表文档工件工具箱并将其与 CSV 文件相关联。双击元素以显示链接文档编辑器。当您创建文档工件元素时，在图表的右下角会显示一个“A”符号。



## 工具箱icon



# 定制库表工件

以表格格式显示自定义数据的自定义库表工件与电子表格类似的图表object。例如：

Development Stages		A	B	1	2	3	4	5	6	7	8	9	C
A	Business requirements on Mobile Apps												
B	IT Maxim "Full Multi-Channel Capabilities"												
1	Generalize requirements	X	X		X				X		X		
2	Review by business			X									
3	Create prototype specification and project description				X		X						
4	Approval by IT management & procurement					X							
5	Implement prototype						X						
6	Review by business							X					
7	Create development guidelines							X	X		X		
8	Review by IT organization									X			
9	Approval of guidelines by IT management										X		

使用这个元素的好处包括：

- 准确地 在适用的地方提供有关元素、图表或项目管理的额外“非建模”信息，例如 SWAT分析或能力矩阵
- 以方便的人类可读和（如果合适的话）人类可编辑的格式提供此类信息
- 使用脚本和插件读取和更新数据的能力

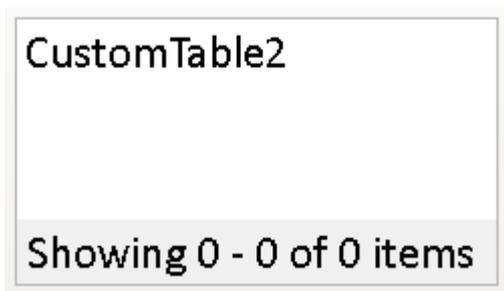
此特征在Enterprise Architect的企业统一版和终极版中可用，从 15.0 版开始。

## 创建自定义库表

从工具箱的“Dynamic图表”页面，将自定义库表图标拖到图表上。



库表元素在图表和浏览器窗口中创建。在图表上，单击元素将其选中，然后拖动元素的边框将其扩展为更舒适的大小以进行编辑。



## 使用自定义表

要修改自定义库表，必须先将其置于编辑模式。要开始编辑，请右键单击图表上的自定义库表元素并选择“编辑自定义库表”。或者，单击元素并按 Enter 键或 F2 键。

在编辑模式下，您可以修改和格式化表内容。

要退出编辑模式，请右键单击元素并选择“退出编辑”选项。或者，取消表元素（通过单击元素外部，或按“Esc”键）也将退出编辑模式。

退出编辑模式将自动保存您的更改。

1. 自定义库表数据内容将自动以XML格式保存到元素的“数据”属性中。
2. 自定义库表行/单元格式数据将自动以XML格式保存到'dataformat'属性中。

如果您通过直接编辑 XML 更新自定义库表元素的属性，您可以通过右键单击元素并选择“刷新自定义库表”选项来刷新图表中的元素。

手术	描述
定义网格大小	<p>新的自定义库表元素还没有定义网格大小。</p> <ol style="list-style-type: none"> <li>1. 右键单击元素正文中的空白区域，然后选择“设置网格大小...”选项。</li> <li>2. 在“设置网格大小”对话框中，输入所需的行数和列数。</li> <li>3. 点击确定按钮。</li> </ol> <p>列会自动调整大小以占据库表元素的可视区域。但是，行默认为单行高。</p>
添加更多列	<p>您有两个选项可以向表中添加更多列：</p> <ul style="list-style-type: none"> <li>• 在表的右侧添加一个新列 - 右键单击表并选择“添加列”选项</li> <li>• 在表中的特定点插入一列 - 右键单击一列并选择“插入列”选项，并根据需要选择“在选定列之前”或“在选定列之后”</li> </ul>
添加更多行	<p>您还有两个选项可将行添加到表中：</p> <ul style="list-style-type: none"> <li>• 在表底部添加一个新行——右键单击表并选择“添加行”选项；如果不滚动到该行可能不可见，但请注记元素右下角的“显示 x - n 个项目”计数器将递增</li> <li>• 在表中的特定点插入一行 - 右键单击一行并选择“插入行”选项，并根据需要选择“Above Selected Row”或“Below Selected Row”；插入行的默认高度为一行</li> </ul>
删除列或行	<p>右键单击列或行中的单元，然后选择“删除选定行”或“删除选定列”。</p> <p>您一次不能删除多个行或列，也不能删除包含合并单元的列或行（即使它们不包含行或列之外的单元）。</p>
复制内容	<p>当表格中有数据表，您可以选择将所选单元或整个表的内容复制到剪贴板，粘贴到外部电子表格工具或文本文件中。</p> <p>选择“复制到剪贴板”选项，然后选择“选定”或“全部”。</p>
更改列宽	<p>单击一列单元并将列标题的边框单元到所需的宽度，或右键单击库表元素并选择“设置列宽度”选项。（要选择多列，请在选择每一列时按 Ctrl 键。）</p> <p>如果您选择“设置列宽”选项，将显示“设置列宽”对话框。输入所需的宽度（以像素为单位），或单击箭头以每次单击将值增加或减少一个像素。输入宽度后点击确定按钮。</p> <p>注记：</p> <ul style="list-style-type: none"> <li>• 列宽受元素宽度的限制；您不能无限期地增加一列的宽度，因为宽度增加将在其中一列将在库表上失去其可见存在的点之前被阻止</li> <li>• 通过拖动标题边框手动设置列宽会等量调整任何未设置列的宽度；例如，在三列库表中，将一列宽度增加 8 个像素，其他两列的宽度各减少 4 个像素</li> <li>• 通过“设置列宽”对话框设置列宽会更改任何未设置列的宽度并手动设置等量的列</li> <li>• 调整库表元素的大小会按比例调整库表内的列宽</li> <li>• 文本字符串将在列的宽度内换行</li> </ul>
更改行高	<p>如果您想将一个或多个行的高度增加或减少一个行高，请右键单击该行中的一个单元”或“减少行行”选项。（要选择多行，请在选择每一行时按 Ctrl 键。）</p> <p>如果您想将一个或多个行的高度增加或减少几个行高，请右键单击所选中</p>

	<p>的 a 并选择 设置单元选项”。将显示 设置行线”对话框。</p> <p>以行为单位键入所需的高度，或单击箭头以每次单击将值增加或减少一行。单击确定按钮保存设置。</p> <p>注记：</p> <ul style="list-style-type: none"> <li>• 更改一或多行的高度不会影响库表中未选择行的高度</li> <li>• 增加一行的高度可以将其他行滚动到看不见的地方，超出库表元素的边界</li> <li>• 如果一个单元的文本超出单元的显示范围，文本会滚动到unit的顶部和底部单元；在这些情况下，将鼠标悬停在单元上以在弹出字段中显示全文，或向行添加更多高度以容纳文本</li> </ul>
<p>格式化网格</p>	<p>您可以执行操作来格式化单元在表中的外观，以及整个表的外观。右键单击表并选择：</p> <ul style="list-style-type: none"> <li>• Show Grid Lines - 隐藏或显示所有分隔单元的线；这不会隐藏使用 设置单元格边框”选项专门显示的那些行</li> <li>• 设置单元格边框 - 如果关闭 显示网格线”并且您已突出显示一个或多个单元，请选择所需单元边的选项以在单元上显示该单元：             <ul style="list-style-type: none"> <li>- 最佳</li> <li>- 正确的</li> <li>- 底部</li> <li>- 剩下</li> <li>- 重置默认值（隐藏当前选定单元的可见边框）</li> </ul> </li> </ul> <p>注记边框宽度固定为 1px。</p> <ul style="list-style-type: none"> <li>• 设置网格颜色 - 显示 选择项目颜色”对话框；单击 设置颜色”下拉箭头和调色板中的适当颜色，然后单击 确定”按钮将颜色应用于表中的所有单元边框（显示网格线”）或那些专门选择的for display（设置单元格边框）</li> <li>• Merge Selected - （如果您通过将光标扫过它们选择了两个或多个单元，则可用）单独的单元成为一个单元，格式为最上面和/或最左边的单元；选定的单元必须形成一个规则的块 - 你不能将两个在单元上的单元与一个在下一行的单元合并，单元形成一个矩形</li> <li>• 单元Selected - 先前合并的单元以其原始尺寸、颜色、文本和格式重新出现</li> </ul>
<p>将文本添加到单元格</p>	<p>双击单元并开始输入。</p> <p>或者，如果您已将文本复制到缓冲区中，请单击单元，右键单击并选择 粘贴”选项。请注记，格式（即使来自另一个表单元）不会在副本中传输。</p>
<p>格式文本</p>	<p>您可以执行许多操作来格式化一个单元的完成文本。这些选项不能对单元中的部分文本字符串进行操作。请注记，这些选项在单击关闭时单元。</p> <p>右键单击单元并选择所需的选项：</p> <ul style="list-style-type: none"> <li>• 文本水平对齐——选择合适的子选项使文本与单元的左、中或右单元（新单元默认为左对齐文本）</li> <li>• 垂直对齐文本-选择适当的子选项将文本与单元的顶部、中心或底部单元（新单元默认为居中对齐文本）</li> <li>• 设置文本颜色 - 显示 选择项目颜色”对话框；单击 设置颜色”下拉箭头和调色板中的适当颜色，然后单击 确定”按钮</li> <li>• 设置背景颜色 - （设置单元的单元，无论它是否包含文本）选择项目颜色”对话框显示；单击 设置颜色”下拉箭头和调色板中的适当颜色，然后单击 确定”按钮</li> <li>• 切换粗体文本- 文本变为粗体或恢复正常</li> </ul>

## 使用标记值

最初定义和填充自定义库表的最简单方法是使用上下文菜单选项。但是，如果要设置多个类似结构的库表，将 XML 从一个表的标记值复制到另一个表的标记值，或者添加脚本来读取和/或填充库表。每个标记值都是一个 <memo> 类型，可以包含很长的 XML 定义集合。

标记值	内容
数据	<p>当您建立网格时，基本结构在此标记值中定义在 XML 中，并且当您添加数据值时，它们将插入到结构定义的适当行中。例如：</p> <pre>&lt;?xml 版本=" 1 .0"?&gt; &lt;adhocTable&gt; &lt;表格&gt; &lt;行&gt; &lt;column&gt;标题 1&lt;/column&gt; &lt;column&gt;标题 2&lt;/column&gt; &lt;column&gt;标题 3&lt;/column&gt; &lt;column&gt;标题 4&lt;/column&gt; &lt;column&gt;标题 5&lt;/column&gt; &lt;/row&gt; &lt;行&gt; &lt;column&gt;行名&lt;/column&gt; &lt;列&gt;&lt;/列&gt; &lt;列&gt;&lt;/列&gt; &lt;列&gt;&lt;/列&gt; &lt;列&gt;&lt;/列&gt; &lt;/row&gt;</pre> <p>如果您想直接在这个标记值的 XML 中输入数据，编辑自定义库表“上下文菜单必须关闭（您已选择 退出编辑”选项），您应该选择 刷新自定义库表”上下文菜单选项经常更新表图像。</p>
数据格式	<p>当您定义网格的格式和外观时，该定义以 XML 格式存储在此标记值中。例如：</p> <pre>&lt;?xml 版本=" 1 .0"?&gt; &lt;数据格式&gt; &lt;风格&gt; &lt;网格行="8"列="5"&gt; &lt;gridcolor&gt;16646398&lt;/gridcolor&gt; &lt;/grid&gt; &lt;细胞&gt; &lt;单元行="0" col="0"&gt; &lt;bold&gt;真&lt;/bold&gt; &lt;txtcolor&gt;255&lt;/txtcolor&gt; &lt;边框&gt; &lt;左&gt;0&lt;/左&gt; &lt;top&gt;0&lt;/top&gt;</pre>

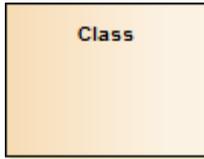
	<pre> &lt;right&gt;1&lt;/right&gt; &lt;底部&gt;1&lt;/底部&gt; &lt;/边界&gt; &lt;/细胞&gt; &lt;单元="0" col=" 1 "&gt; &lt;txcolor&gt;13434880&lt;/txcolor&gt; &lt;边框&gt; &lt;左&gt;0&lt;/左&gt; &lt;top&gt;0&lt;/top&gt; &lt;right&gt;1&lt;/right&gt; &lt;底部&gt;1&lt;/底部&gt; &lt;/边界&gt; &lt;/细胞&gt;                 </pre> <p>如果您想直接在标记值的XML中修改定义，编辑自定义库表“上下文菜单”必须关闭（您已选择“退出编辑”选项），您应该选择“刷新自定义库表”上下文菜单选项经常更新表图像。</p>
--	--

## 自定义库表脚本

您还可以将JavaScript脚本与您的自定义库表元素相关联。通常，脚本可能用于从表中读取数据或更新表中的数据。脚本保存在名为“脚本”的操作中。

行动	描述
定义一个脚本	要定义或编辑脚本： <ul style="list-style-type: none"> <li>• 单击表元素以选择它</li> <li>• 选择功能区选项“开发 &gt;源代码 &gt;行为” (Alt+7)</li> <li>• 在编辑器的左侧面板中选择名为“脚本”的操作</li> <li>• 在编辑器的右侧面板中输入脚本代码</li> </ul> 有关可用方法的详细信息，请参阅细节类帮助帮助。
运行脚本	如果已经为表定义了关联脚本，则可以在运行未处于编辑模式时右键单击表，然后选择“运行自定义库表脚本”选项来运行。

# 类



## 描述

A类是一种object类型的表示，它反映了系统内此类对象的结构和行为。它是一个创建实际运行实例的模板，尽管可以定义一个类来控制它自己的执行，也可以定义为一个模板或参数化类，指定必须由任何绑定类定义的参数。

A类有属性（数据）和方法（操作或行为）。类可以从父类继承特性并将行为委托给其他类。类模型通常描述系统的逻辑结构，并且是构建组件的构建块。

类的顶部显示与类关联的属性（或数据元素）。它们在运行时保存object的“状态”。如果信息保存到数据存储并可以重新加载，则称为“持久性”。下半部分包含类操作（或运行时的方法）。描述类提供给其他类的行为，以及它具有的内部行为（私有方法）。



类元素一般用在类图和复合结构图中。

Enterprise Architect还支持许多原型类元素来表示网页建模中的各种实体。A也可以与关联连接器集成以形成关联类类以允许关联连接器具有定义某些类型的UML关系的操作和属性。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, pp.194-195) 指出：

类的目的是指定对象的分类并指定表征这些对象的结构和行为的特征。

类是一种封装分类器，其特征是属性、操作、信号接收、端口和连接器。类的属性是类所拥有的属性。其中一些属性可能代表二进制关联的结束。类的对象必须包含属于类成员的每个属性的值，根据属性的特征，例如它的类型和多重性。

在类中实例化object时，对于类的每个具有指定默认值的属性，如果没有为实例化显式指定属性的初始值，则评估默认的 ValueSpecification 以设置属性的初始值为object。

一个类的操作可以在一个object上调用，给定一组特定的操作参数值，(…)。

A类不能访问另一个类的私有特征，或在不是其祖先的另一个类上受保护的属性。

A类充当其范围内定义的各种分类器的命名空间，包括类。嵌套分类器是包含类的成员。分类器嵌套用于隐藏信息的命名空间的成员。



## 活动课程



### 描述

一个 **Active**类表明，当被实例化时，类控制它自己的执行。它可以独立运行并定义自己的行为线程，而不是被其他对象调用或激活。

### 在Enterprise Architect中定义一个 **Active**类

节	行动
1	突出显示一个类，并显示其“属性”对话框（右键单击类并选择“属性 属性”选项）。
2	选择对话框右下方的“详细信息”选项卡。
3	选中“激活”复选框。
4	单击确定按钮保存更改。

### OMG UML规范

OMG Unified Modeling Language规范 (v2.5.1, p.438) 指出：

活动object是一个object，作为其创建的直接结果，它开始执行其分类器行为，并且在完成行为被执行或object被某个外部object终止之前不会停止。（这有时被称为“具有自己的控制线程的object”。）活动object响应来自其他对象的通信的点仅由活动object的行为决定，而不是由调用object决定。如果活动object的分类器行为完成，则终止该object。

# 参数化类 ( 模板 )

Enterprise Architect支持参数化 ( 模板 ) 类，这些类指定必须由任何绑定类定义的参数。

参数化类通常用 C++ 实现；Enterprise Architect为 C++ 导入并生成模板类。

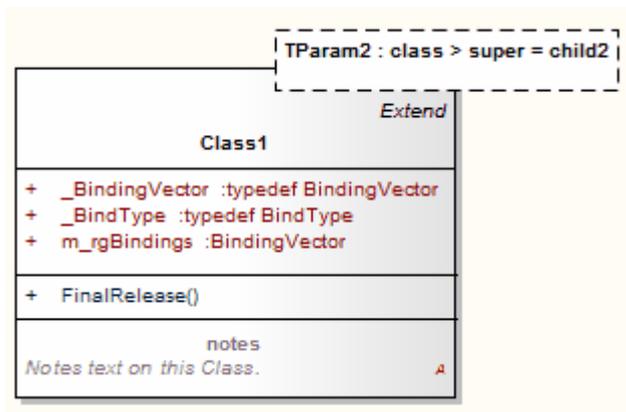
模板类的功能可以被任何绑定类重用。如果为参数指定了默认值，并且绑定类没有为该参数提供值，则使用默认值。

## 创建参数化类

节	行动
1	单击所需的类。
2	选择 设计>元素>管理>模板参数”功能区选项。 将显示 模板”对话框。
3	在 模板参数”面板中，单击 添加”按钮。 将显示 模板参数”对话框。
4	类型在参数的名称和类型中，如果需要，单击 约束”和 默认”字段后的  按钮，从 选择<项目>”对话框中选择所需的约束和默认类。 默认类可以是约束分类器或从约束分类器派生的任何类。

## 符号示例

在图表中，模板类在类的右上角以轮廓的形式显示参数。



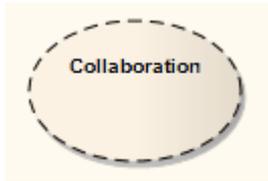
## OMG UML规范

OMG Unified Modeling Language规范 (v2.5. 1 , p. 622) 指出：

模板是A参数化元素，可用于使用模板关系生成其他模型元素。模板签名的模板参数指定将由绑定中的实际参数 ( 或默认参数 ) 替换的形式参数。



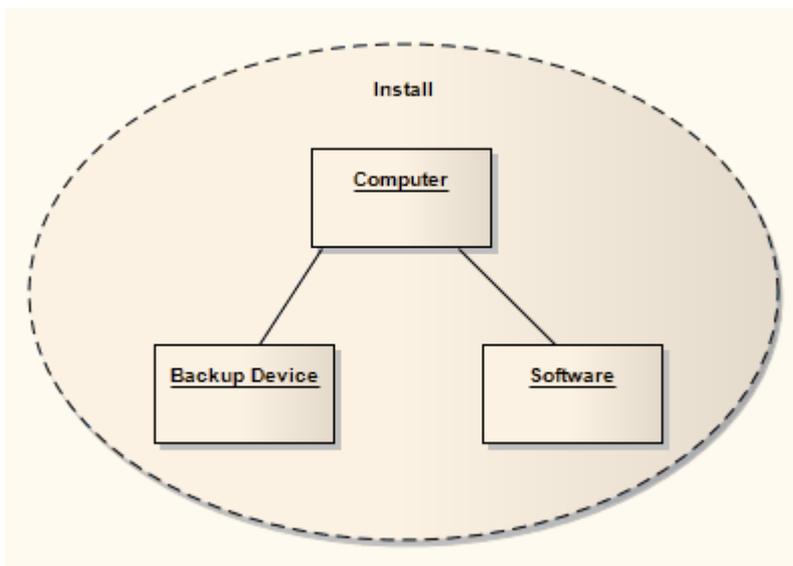
# 协作



## 描述

协作定义了A组协作角色及其连接器。这些用于在复合结构图中共同说明特定功能。协作应仅指定完成特定任务或函数所需A角色和属性。尽管在实践中，行为及其角色可能涉及许多相切的属性和属性，但隔离主要角色及其必要条件可以简化和阐明行为，并提供重用。A实现了协作模式以适用于各种情况。

此示例说明了一个协作安装，如图所示连接了三个角色（对象）。此协作的过程可以通过使用交互、添加通讯交互序列。



要了解在特定情况下引用协作，请参阅协作应用帮助。

Enterprise Architect支持一个刻板印象的协作来表示一个业务协作用例在业务建模中的实现。

## 工具箱icon

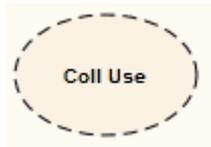


## OMG UML规范：

OMG Unified Modeling Language规范 · v2.5 · 1 , p.222) 指出：

A（协作）描述协作元素的结构，每个元素执行专门的函数，共同完成一些所需的功能。

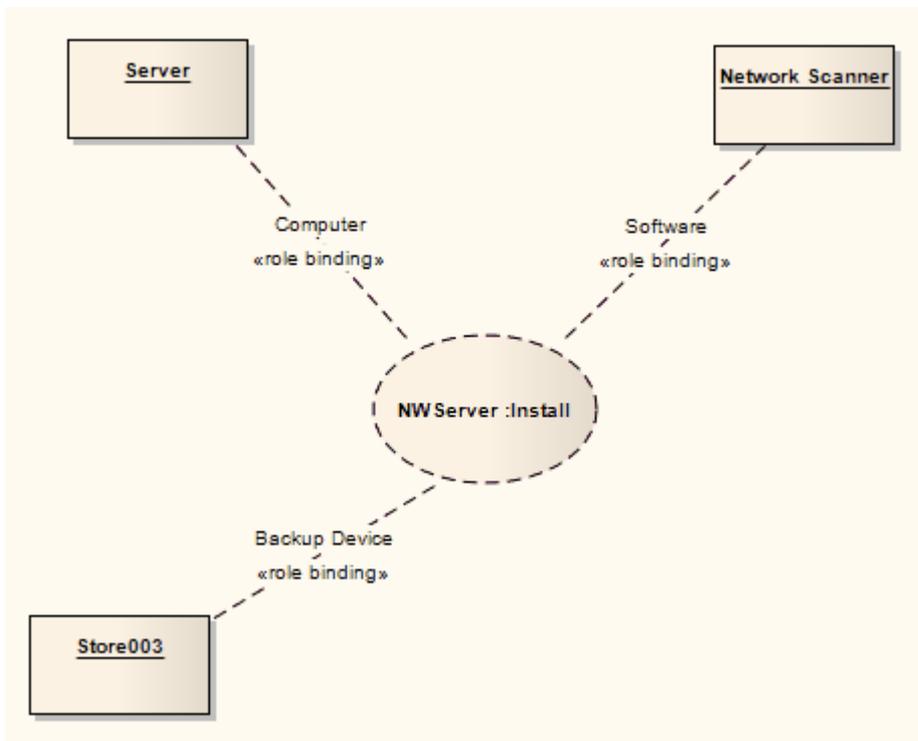
# 协作应用



## 描述

使用协作应用将协作定义的模式应用于特定情况，在复合结构图中。

此示例显示了“安装”的协作使用，用于定义网络扫描仪的安装过程。此过程可以通过附加到协作的交互来定义。（有关安装协作的表示，请参阅协作帮助。）



要创建协作应用，请将图标从复合的工具箱图表图表上。

## 工具箱icon



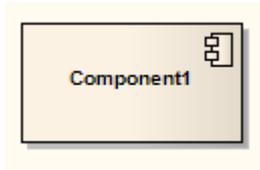
## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.216) 指出：

CollaborUse 表示A组协作的特定协作，用于解释一组元素之间的关系。Collaboration 通过将特定时间的协作上下文到可连接的协作或协作A协作角色，上下文了分类器描述的协作如何应用于给定的协作。可能存在与一个分类器相关的多个协作使用，每个协作使用不同地协作在一个分类器中。A定的协作角色或连接器可能涉及相

同或不同协作的多次使用。

# 部件



## 描述

部件是系统A模块化部分，其行为由其提供和所需的接口定义；组件的内部工作部件是不可见的，并且它的使用环境独立。源代码文件、DLL、Java bean 和其他定义系统的工件可以在组件中体现。

A部件可以由多个类或组件拼凑而成。随着较小的组件组合在一起以创建更大的组件，最终系统可以在部件图中以构建块样式进行建模。通过在离散组件中构建系统，数据和行为的本地化可以减少类和对象之间的依赖性，从而提供更健壮和可维护的设计。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范，v2.5，1，pp.208-209) 指出：

本子条款规定了一组可用于定义任意大小和复杂性的软件系统的结构。特别是，它将部件指定为具有定义明确的接口的模块化单元，可在其环境中替换。部件概念解决了基于组件的开发和基于组件的系统结构领域，其中在整个开发生命周期中对部件进行建模，并在部署和运行时逐步细化。

基于组件的开发的一个重要方面是重用以前构建的组件。A部件总是可以被认为是系统或子系统中的一个自治单元。它有一个或多个提供的和/或需要的接口（可能通过端口暴露），并且它的内部是隐藏的，并且除了由它的接口提供之外是不可访问的。尽管它可能依赖于所需的接口方面的其他元素，但部件是封装的，并且它的依赖关系被设计为可以尽可能独立地处理它。因此，组件和子系统可以通过将它们连接（“接线”）在一起来灵活地重用和替换。

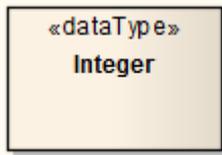
自治和重用方面也在部署时扩展到组件。实现部件的工件旨在能够独立部署和重新部署，例如更新现有系统。

OMG Unified Modeling Language规范，v2.5，1，p.224) 指出：

部件表示系统A模块化部分，它封装了其内容，并且其表现形式在其环境中是可替换的。

组件A提供的和所需的接口定义其行为。因此，组件作为一种类型，其一致性由这些提供和必需的接口定义（包括它们的静态和动态语义）。

# 数据类型



## 描述

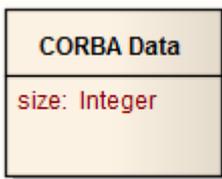
数据类型是A特定的分类器，类似于类，只是数据类型不能拥有子数据类型，并且数据类型的实例仅由它们的值标识。例如Person类的实例是 Helen object，而整数数据类型的实例是 12。

数据类型实例的所有副本，以及具有相同值的该数据类型的任何实例，都被认为是同一个实例。也就是说，Helen 的实例不一定是同一个 Helen，但所有 12 都是同一个 12。例如，表盘上的 12 与一年中的月数完全相同。

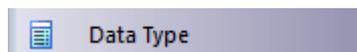
如果结构相同且相应属性的值相同，则具有属性的数据类型的实例（即结构化数据类型的实例）被认为是相同的。如果数据类型具有属性，则该数据类型的实例包含与属性匹配的属性值。

数据类型A典型用途是表示编程语言原始类型或 CORBA 基本类型。例如，整数和string类型通常被视为数据类型。

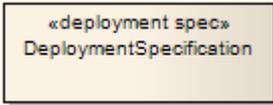
数据类型由A关键字 «dataType» 的矩形表示，或者，当它被（例如）属性引用时，由包含数据类型名称的string表示，如下所示：



## 工具箱icon



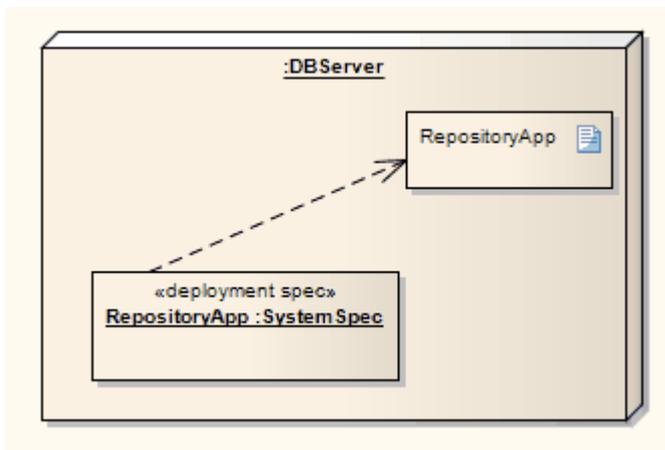
# 部署规范



## 描述

A部署参数指导部署规范的部署和大多数软件技术的工件。规范列出了必须定义才能进行部署A属性，如部署图所示。本规范的一个实例指定了参数的值；可以为多个工件实例化单个规范。

这些规范可以通过某些组件配置文件进行扩展。配置文件可能添加到部署规范的标准标记值的示例是带有标记值 {thread, process, none} 的«concurrencyMode» 或带有标记值 {transaction, 部署规范, none} 的«transactionMode»。此示例描述了部署在服务器节点上的仓库RepositoryApp，按照RepositoryApp 的规范，从部署规范工件实例化。



## 工具箱icon

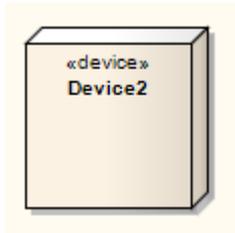


## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.662) 指出：

A部署规范，指定部署在工件上的一组确定执行参数的组件的属性。部署规范可以针对特定类型A容器。一个工件化或实现部署属性的部署规范是部署描述符。

# 设备



## 描述

设备是具有处理能力的工件资源，可以部署用于执行，如部署图中A电子资源。复杂设备可以由其他设备组成；也就是说，设备可以是嵌套元素，其中物理机器通过名称空间所有权或通过设备类型的属性分解为其元素。

## 工具箱icon

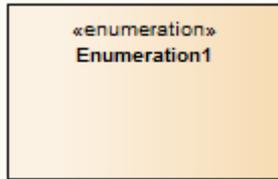


## OMG UML规范：

OMG Unified Modeling Language规范，v2.5。1, p.663) 指出：

设备是具有处理能力A物理计算资源，可以在其上部署工件以供执行。设备可能很复杂（即，它们可能由其他设备组成）。

# 枚举



## 描述

枚举是一种数据类型，其实例可以是许多用户定义的枚举文字中的任何一种。可以在其他包或配置文件中扩展适用的枚举文字集。您可以在类或包图表中创建枚举，并在使用工具箱的元模型和配置文件页面开发的图表中。

## 工具箱icon

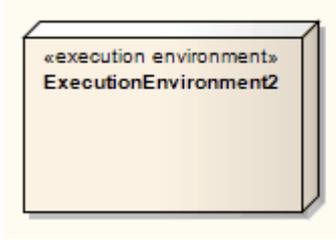


## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.175) 指出：

枚举是一种数据类型，其值在模型中枚举为 EnumerationLiterals。

# 执行环境



## 描述

一个提供特定类型的组件执行环境的执行环境是部署在其上的一个节点，该执行环境是以工件的形式执行的。这在部署图中进行了描述。

执行环境可以嵌套；例如，数据库执行环境可以嵌套在操作系统执行环境中。然后将适当类型的组件部署到特定的执行环境节点。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.664) 指出：

执行环境是一个节点，它为以可执行工件的形式部署在其上的特定类型的组件提供执行环境。

OMG Unified Modeling Language规范 (v2.5.1, p.658) 还指出：

通常，节点通过节点上定义的组合关系分配给一些，通常是更高级别的节点或一般节点系统。

ExecutionEnvironment 可以嵌套（例如，数据库 ExecutionEnvironment 可能嵌套在操作系统

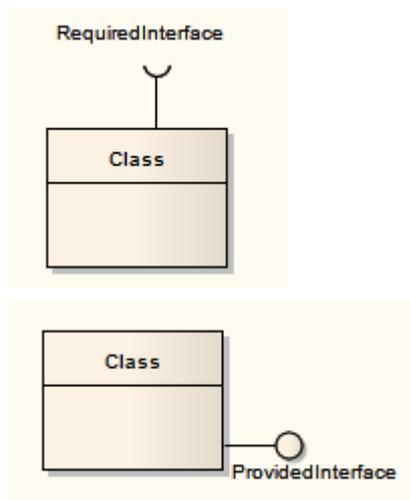
ExecutionEnvironment 中）。ExecutionEnvironment 可能具有系统级服务的显式接口，可以由部署的元素调用。

在这种情况下，软件 ExecutionEnvironment 服务应该被显式地建模。然后使用部署关系将适当类型的应用程序

组件部署到特定的 ExecutionEnvironment 节点或 DeployedArtifacts 的 Manifestations 关系。对于每个组件部署，

这些服务的各个方面可以由特定类型的执行环境的部署属性中的属性确定。

## 曝露接口



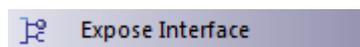
### 描述

曝露接口元素是一种图形方法，用于在部件或复合结构图中描述部件、类或部件的所需或提供的接口。它只是标识元素提供或需要接口的事实；要描述提供的接口被另一个元素使用或所需的接口由另一个元素提供的事实，请使用组装连接器。

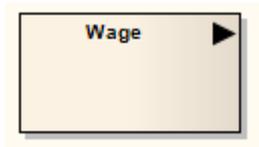
曝露接口元素必须附加到类或部件元素，它成为类或元件的部件元素；它不能独立存在。您可以将多个曝露元素附加到另一种元素上。

当您创建曝露接口元素时，会显示一个对话框，您可以在其中输入元素的名称并指定它是代表必需的接口还是提供的接口。

### 工具箱icon



# 信息项

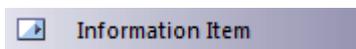


## 描述

信息项元素表示数据的抽象，该数据可以在两个对象之间传递。术语“信息项”也更松散地应用于任何分类器，它表示可以在两个对象之间传达的数据类型的更具体标识。

两个对象之间的项（任何一种）的传递和实现由一个信息流连接器表示。

## 工具箱icon



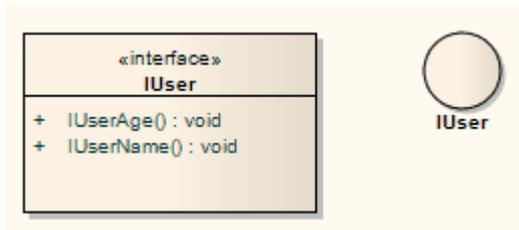
## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.673) 指出：

InformationItem 表示可以以非常抽象的方式从源流向目标的多种信息。它们代表可能在系统内移动的信息类型，但不详细说明传输信息的细节。传输信息的细节是最终定义信息项的其他分类器的范围。因此，InformationItems 不能被实例化，并且它们本身不具有特征、概括或关联。

InformationItems 的一个重要用途是在早期设计阶段表示信息，可能在最终定义它们的详细建模决策之前。InformationItems 的另一个目的是以不太精确但可能更一般和可交流的方式抽象复杂模型的部分。

# 接口



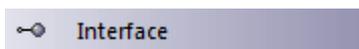
## 描述

An接口是实现者同意满足的行为规范（或合同）。通过实现接口，可以保证类支持所需的行为，这使系统能够以相同的方式处理不相关的元素；即通过通用接口。您还可以在复合结构图中使用接口。

接口的绘制方式与类类似，并指定了操作，如下所示。它们也可以绘制为一个没有详细操作的圆圈 - 右键单击元素并选择“使用圆形符号”选项以在样式之间切换。实现接口的连接器接口为圆形，绘制为实线没有目标箭头。

一个接口不能被实例化（也就是说，你不能从一个接口创建一个object）。您必须创建一个“实现”接口规范的类，并在类主体中为每个接口操作放置操作。然后，您可以实例化类。

## 工具箱icon



## OMG UML规范：

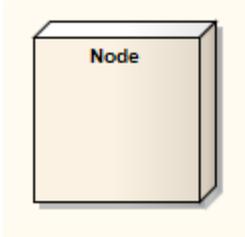
OMG Unified Modeling Language规范 (v2.5. 1 , p.171) 指出：

An接口是一种分类器，表示一组公共特征和义务的声明，它们共同构成一个连贯的服务。接口指定合同；任何实现接口的分类器实例都应履行该合同。与接口相关的义务是约束（例如前置条件和后置条件）或协议规范的形式，这可能会对通过接口的交互施加排序限制。接口可能不会被实例化。相反，接口规范由BehavioredClassifier实现或实现，这意味着 BehavioredClassifier 呈现符合接口规范的公共外观。

笔记。A给定的 BehavioredClassifier 可以实现多个接口，并且一个接口可以由许多不同的 BehavioredClassifier 实现。接口提供了一种划分和表征公共特征组和实现行为分类器应具备的义务的方法。

An接口没有具体说明它是如何实现的，而只是说明实现 BehavioredClassifiers 需要支持什么。也就是说，此类行为分类器应提供一个公共外观，该外观由属性、操作和符合接口的外部可观察行为组成。

# 节点



## 描述

节点是部署系统A物理设备，例如工作组服务器或工作站。节点A托管组件和其他可执行代码，它们又可以连接到特定的进程或执行空间。典型的节点是客户端工作站、应用服务器、大型机、路由器和终端服务器。

节点在部署图中用于模型系统的部署，并说明已实施工件的物理分配。它们还用于 Web 建模，来自工具箱中的专用 Web 建模页面。

## 工具箱icon



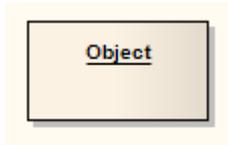
## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.658) 指出：

节点关系是计算资源，可以A部署在其上工件执行。对于高级建模应用程序，节点可能具有由复杂定义的嵌套结构，并且可能相互连接以表示特定情况。Nodes的内部结构只能由其他Nodes组成。除了参与节点之外，节点还获得从部署在它们上的清单的关系中工件的相关元素。

节点可以进一步细分为设备和执行环境。设备代表物理机器组件。ExecutionEnvironments 代表应用程序组件在执行时可能需要的标准软件系统。例如，特定的配置文件可能会为诸如«OS»、«工作流引擎»、«数据库系统»和«J2EE 容器»之类的执行环境定义原型。

# 物件

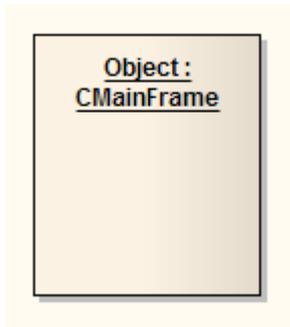


## 描述

物件是类在运行时的特定实例。例如，车牌为 AAA-001 的汽车是具有车牌号属性的一般类汽车的一个实例。分析中经常使用对象来表示任何业务中存在的大量工件和项目，例如纸张、传真和信息。要模型对象在运行时的变化行为，请使用运行时状态。

在分析的早期，对象可用于快速捕获系统域内、物件、复合结构或通讯图中的所有相关事物。随着模型的进展，这些分析对象被细化为通用类，从这些类中可以派生实例以表示常见的业务项目。一旦定义了类，就可以输入对象；也就是说，它们可以有一个分类器集来指示它们的基本类型 - 请参阅分类器和实例主题。

Enterprise Architect还支持许多原型物件元素来表示业务建模中的各种实体。



## 工具箱icon



# 运行时状态

在运行时，一个物件实例的属性可以具有特定的值，或者存在于特定的状态中。要模型对象在运行时的变化行为，请使用从 选择 <Item>”对话框和运行时状态或运行状态中选择的实例值。

通常对已经具有分类器集的对象的行为感兴趣。您可以从分类器的属性列表中选择并为您的物件实例应用特定值。如果分类器有一个子状态机，它的状态会传播到一个可以定义物件运行时状态的列表。

## 示例

此示例为列出的变量定义运行时值，这些变量是实例的账户项分类器的属性。



## 访问

功能区	>开始>应用>设计>属性，点击图中的物件或浏览器窗口>运行状态 设计>元素>编辑器>属性，点击图中的物件或浏览器窗口>运行状态
上下文菜单	在图表或浏览器窗口中，右键单击 物件 特征 ”设置运行状态选项
键盘快捷键	Ctrl+Shift+R Ctrl+2 > 点击图中的一个物件或浏览器窗口 >运行状态

## 将运行时状态实例变量添加到一个物件

在属性窗口或 < object名称>：特征”对话框中， 运行状态”页面列出了从物件元素的分类器继承的任何变量。这些继承的变量最初没有值并且处于非活动状态。您可以为它们激活和定义运行状态，或者您可以右键单击 < object名称>：特征”对话框并选择 隐藏继承变量”选项以将它们从视图中隐藏。

节	行动
1	在 变量”字段中： <ul style="list-style-type: none"> <li>• 用新变量的名称改写新变量文本，或</li> <li>• 单击继承变量的名称以激活</li> </ul>
2	在 运营商”字段中，单击下拉箭头并选择符合运行状态值的运营商。运营商包括： <ul style="list-style-type: none"> <li>• 空白 (无操作员)</li> <li>• !=</li> <li>• &lt;</li> <li>• &lt;=</li> </ul>

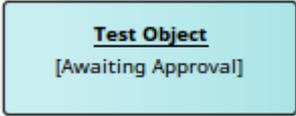
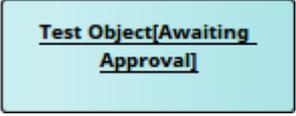
	<ul style="list-style-type: none"><li>• &lt;&gt;</li><li>• =</li><li>• =&gt;</li><li>• &gt;</li></ul>
3	在“值”字段中，输入变量运行状态的值。
4	如有必要，请输入一些解释性注记。
5	单击或添加下一个变量，或单击关闭按钮保存更改。

## 删除物件的运行时状态变量

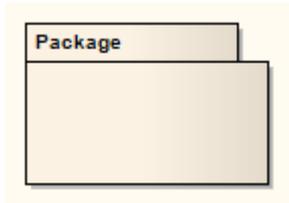
节	行动
1	在“变量”字段中，右键单击要删除的变量并选择“删除”选项。 (或者，单击变量并按 Ctrl+Del。)
2	单击关闭按钮。

# 物件状态

## 设置类实例的物件状态

节	行动
1	<p>右键单击图表中所需的物件并选择 高级 设置物件状态'选项。</p> <p>将显示 设置实例状态"对话框。</p>
2	<p>在 状态"字段中，键入所需的状况（例如 等待批准"）或从状态列表中选择一。</p> <p>状态"字段的下拉列表填充有：</p> <ul style="list-style-type: none"> <li>• object的分类器拥有的任何状况</li> <li>• object的分类器的任何超类拥有的任何状况</li> <li>• 任何状态机拥有的状态机拥有的object的分类器</li> <li>• object分类器的任何超类拥有的状态机拥有的任何状况</li> </ul>
3	<p>状态"字段下方是 " 实例名称合并状况"复选框，默认为未选中。任何一个：</p> <ul style="list-style-type: none"> <li>• 忽略复选框以在object名称下方的元素上显示object状况</li> </ul>  <ul style="list-style-type: none"> <li>• 或者选中复选框以将元素上的object状况显示为object名称的后缀</li> </ul> 
4	<p>单击确定按钮以应用状况。</p> <p>object现在以您选择的格式显示运行时状况。</p>

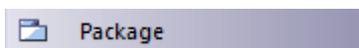
# 包



## 描述

A包是一个名称空间，也是一个可以包含在其他包的名称空间中的元素。A包可以拥有或与其他包合并，并且它的元素可以导入到一个包的命名空间中。除了在浏览器窗口中使用包来组织项目内容外，您还可以将包拖到图表工作区（大多数图表类型，标准和扩展）以进行结构或关系描述，包括包导入或合并。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , pp.241-242) 指出：

包是其成员A命名空间，其中包括通过 `packagedElement` 关联的那些元素（据说是拥有或包含的）和那些导入的。A包定义可以通过合并包含的元素来扩展其他包的内容。

A包可以定义为一个模板并绑定到其他模板：请参阅子条款 7.3，模板，以获取更多信息。可以指定 URI 以提供包的唯一标识符。在UML中，除了配置文件 (...) 之外，对此没有预先确定的用途。例如，它可以被模型管理功能用于模型识别。

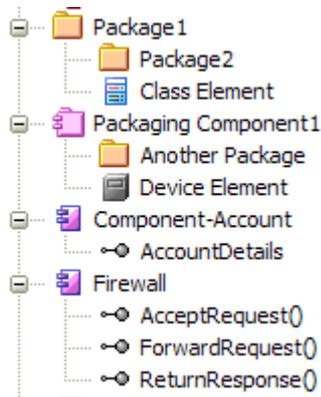
# 包装部件



## 描述

Packaging部件是A在图表中看起来与部件非常相似但在浏览器窗口中表现为包的元素（也就是说，它可以是版本控制的并且可以包含其他包和元素）。它通常用于部件图中。

在浏览器窗口中，三个元素显示如下：



部件元素不能包含子包或包装组件。

## 工具箱icon



# 部件



## 描述

部件是类或接口的运行时实例。可以使用以下符号为部件指定多重性：

$(x\{...y\})$

其中  $x$  指定创建复合结构时的初始或设置的实例数， $y$  表示任何时候的最大实例数。

部件用于表达复合结构或建模模式，可以由各种对象调用以实现特定目的。在说明结构的组成时，可以将 **Parts** 作为其他 **Parts** 的属性嵌入。当作为属性嵌入时，部件可以由一个实心轮廓包围，表示周围的部件按部件拥有部件。或者，一个虚线轮廓表示该属性被周围的部件引用和使用，但不包含在其中。

## 工具箱icon



# 添加属性值

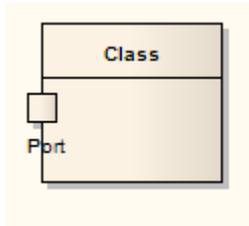
## 将属性值变量添加到部件

A属性值的部件类似于下图：



节	行动
1	右键单击部件并选择 特征 设置属性值的选项（或按 Ctrl+Shift+R）。 将显示 设置属性值”对话框。
2	在 变量”字段中，单击下拉箭头并选择变量，或输入新变量名称。
3	设置运算符和值，并可选择键入注记。
4	点击确定按钮保存变量。

# 端口



## 描述

端口定义分类器与其环境之间的交互。可以使用接口元素来描述控制这种交互的接口。任何连接到端口的连接器都必须提供所需的接口，如果已定义。端口的边界可以出现在包含的部件类或复合元素上。

A端口一个类型化的结构特征或属性，其包含分类器端口图表。通常在类、物件图表和复合结构图表中创建。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p. 230) 指出：

A是端口分类器的一个属性，它指定封装分类器与其环境之间或封装分类器（行为）与其内部部分之间的不同交互点。端口通过连接器连接到属性器，通过连接器可以发出请求调用 BehavioralFeatures。A可以指定端口向其环境提供（提供）的服务以及 EncapsulatedClassifier 期望（要求）其环境的服务。端口可能有A关联的 ProtocolStateMachine。

# 添加一个端口到一个元素

## 向元素添加新端口

使用以下步骤之一：

节	行动
1	点击工具箱“复合”页面中的端口符号，将符号拖到（或点击）目标主机元素。 这会在光标位置附近的边界上创建一个无类型的简单端口。
2	在浏览器窗口中合适的类、部件或复合元素的上下文菜单上，选择“添加端口”选项。
3	将合适的分类器从浏览器窗口拖到图表上的类或部件上。 将显示A提示以在光标位置添加类型化端口或部件。 新的端口由原来的拖拽分类器键入。
4	使用“端口”子菜单（在图表上，右键单击元素 新元素）向当前选定的元素添加新的端口。

## 继承并重新定义端口

A端口是复合分类器（例如部件）的可重新定义和可重复使用的属性。部件可以从其父A继承端口；如果组件的父级拥有端口，则当您打开组件的“部件”选项部件并选中“显示已继承/已继承”复选框时，将端口该交互的特征窗口。

如果你想在组件上显示一个部件的端口，特征窗口提供了两个选项：

- 公开继承的端口- 勾选“显示拥有/继承”复选框，创建一个只读副本到该端口；这对于在父元素中定义端口操作的子元素中端口交互很方便
- 重新定义继承的端口- 右键单击端口并选择“重新定义”选项，以创建端口的可编辑端口；当孩子在元素端口上放置额外的限制或行为时，这很有用

这也适用于从已实现接口端口的组件，以及部件分类组件端口的组件部件。

## 端口 as Owners of Parts

如果将端口键入到具有 Parts 的类中，则端口可以作为这些 Parts 的所有者显示在图表上。要做到这一点：

- 右键单击端口|高级|端口大小可定制，然后放大端口或
- 右键单击端口|特征>部件/属性：选择要显示的部件：选择“显示拥有/继承”

当您希望连接器连接到端口的内部结构时，此特征端口。

## 属性窗口 - 属性，重新定义/子集

端口和部件的属性窗口有一个“属性”选项卡——它定义了元素的类型、初始值和多重性——以及一个标识元素的重新定义和子集属性和定义限定词的“重新定义/子集”选项卡。

您可以通过单击限定词按钮来设置限定词，以显示“限定词”对话框。您可以通过单击相应的添加按钮来添加重新定义和子集属性，以显示“选择属性”对话框。

# 原始



## 描述

原始元素标识A预定义的数据类型，没有任何相关的子结构（也就是说，它在上下文的UML中没有部分）。它可以被视为一种概念数据类型。原始元素可用于支持元对象功能（MOF）规范。

## 工具箱icon

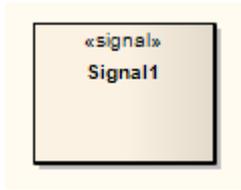


## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.168) 指出：

PrimitiveType 定义了A预定义的 DataType，没有任何子结构。PrimitiveType 可能具有在UML之外定义A代数和操作，例如，在数学上。PrimitiveType 的运行时实例是对应于在UML之外定义的数学元素（例如，整数）的值。

# 信号



## 描述

信号是对象之间通信A发送请求实例的规范，通常在类或包图中。接收object处理由其信号接收指定的接收请求实例。发送请求携带的数据表示为信号的信号。A信号的定义独立于处理信号出现的分类器。

接收被定义为接收object A特征，从元素信号派生而来。接收接受信号的名称和信号的属性作为它的参数。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.178) 指出：

A信号是对象之间的一种通信规范，其中在接收器中异步触发反应而无需回复。

OMG Unified Modeling Language规范 (v2.5. 1 , p.178) 还指出：

接收是声明分类器准备对接收到信号做出反应A声明。

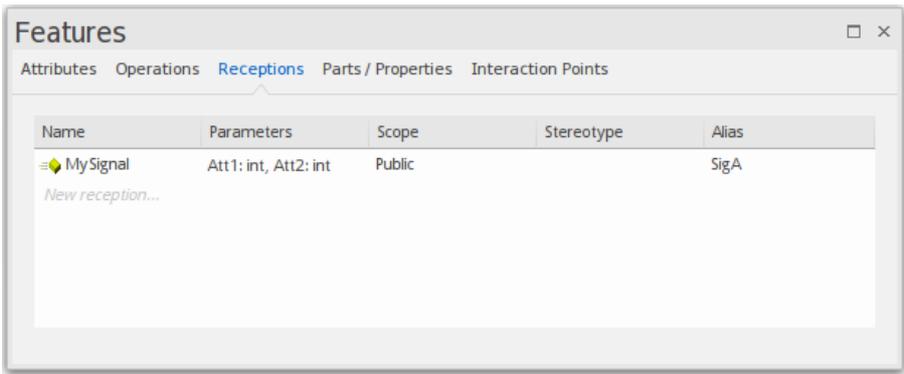
# 接待

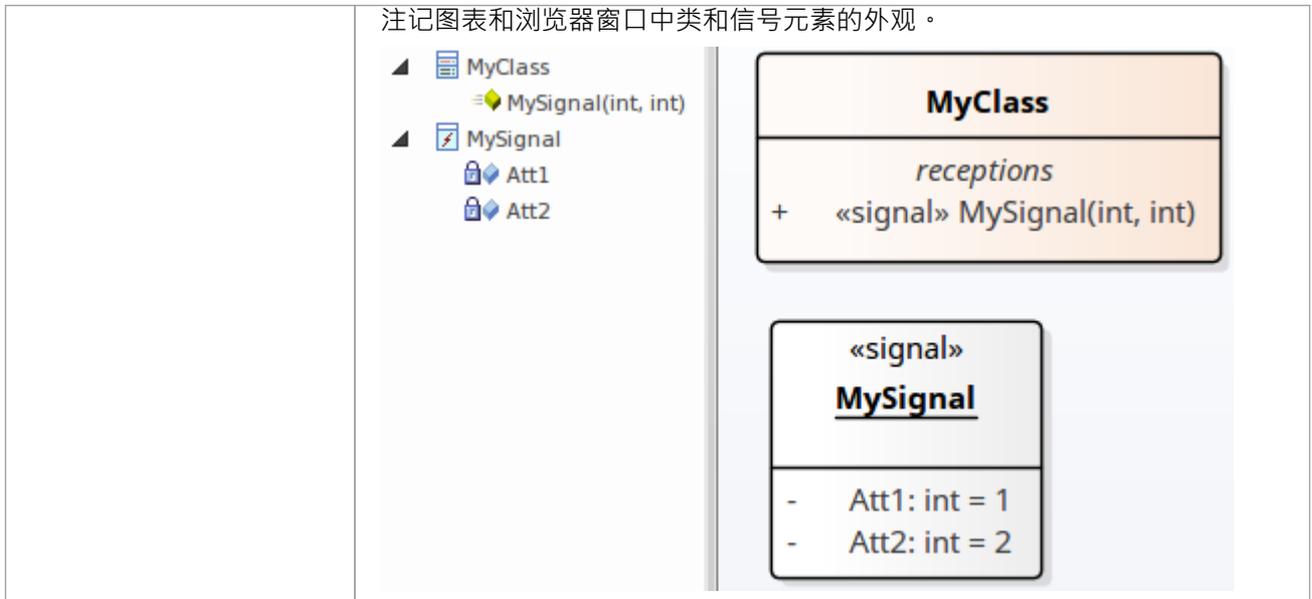
A接收是一个分类器的一个特征，并且是从一个信号元素派生而来的；它以与操作相同的元素处理信号接收信号的信号。



A信号元素就无法创建接收，该信号元素要么已经存在，要么作为创建接收的步骤而创建。Reception的名称取自信号元素的名称，信号的属性定义了信号的参数。对信号的任何更改都会反映在接收中，反之亦然。

## 创建接待处

节	行动
1	在浏览器窗口或图表中创建信号元素。
2	在图表上创建或拖动类或接口元素。
3	<p>任何一个：</p> <ul style="list-style-type: none"> <li>• 将信号元素从浏览器窗口拖到类或接口元素上，然后转到第 8 步，或</li> <li>• 右键单击类元素并选择 特征 信号接收”选项</li> </ul> <p>特征窗口显示在 信号接收”页面。</p> 
4	单击 名称”列中新接收文本右侧的  按钮。将显示 选择信号”对话框（选择 <item>”对话框的变体）。
5	浏览并单击信号元素。（如果你一开始没有创建一个信号元素，你可以使用 加新按钮来创建一个新的信号元素现在。）
6	点击确定按钮。信号元素的名称和属性用于生成 Reception 的名称和参数。
7	单击关闭按钮。
8	



## OMG UML规范

OMG Unified Modeling Language规范 · ( 1 · pp.447 - 448 ) :

接收是声明分类器准备对接收到的信号作出反应A声明。接收指定A信号并指定预期的行为响应。处理信号的细节由与接收相关的行为或分类器本身指定。 ...信号接收使用与关键字<signal>的操作相同的符号表示。

# 属性信号接收窗口

停靠的属性窗口提供了一种方便且直接的方式来查看和编辑建模对象的常见属性，包括特征，例如信号接收。当您在浏览器窗口或图表中单击 **Reception** 时，属性窗口立即显示属性的属性，您可以在同一窗口中快速显示其他类型object的属性 - 元素、属性、连接器和图表 - 无需必须打开和关闭单独的对话框。

信号接收的属性窗口具有三个选项卡：

- 接待处，您可以在其中定义一般设置
- 行为，您在其上定义接收信号时要采取的行为
- 重新定义，您可以在其中设置发生的任何目标操作的重新定义，并检查异常

有关“行为”和“重新定义”选项卡的信息，请参阅操作行为和重新定义操作和选择异常帮助帮助。

## 访问

功能区	>开始>应用>设计>属性；单击图表或浏览器窗口中的接收设计>元素>编辑器>属性，单击图表或浏览器窗口中的Reception
键盘快捷键	Ctrl+2 或 Ctrl+Enter，然后单击图表或浏览器窗口中的 Reception

## 接待处 - 常规设置

字段	描述
名称	显示接收的名称，即派生它的信号元素的名称。您不能更改此字段中的名称。如果信号的名称发生更改，则接收名称会自动更新以匹配。
参数	显示信号元素的属性的名称和类型，接收参数从这些属性派生。您只能通过更新信号属性来更改参数。
范围	显示从信号元素派生的接收范围。如有必要，您可以更改此范围 - 单击字段末尾的下拉箭头并从列表中选择新值。
构造型	( 可选 ) 如果您需要对 Reception 进行分组或专门化，您可以在此字段中键入适当构造型的名称。
别名	( 可选 ) 如有必要，您可以输入接待处的替代名称或参考。
并发	默认为顺序。如果要将其更改为“守卫”或“同步”，请单击下拉箭头并选择适当的值。
抽象的	识别接收是否是抽象的。
静止的	指示接收是否是静态成员的标志；要更改此标志，请根据需要在字段中键入“False”或“True”。



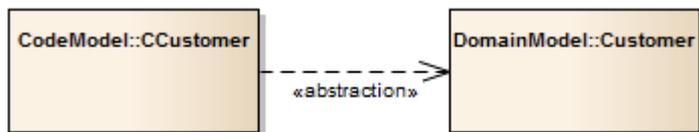
# UML连接器

连接器将元素链接在一起，通常在图表上表示为显示元素如何相互关联的线条。与自然语言进行比较，如果元素是名词，则连接词是描述名词如何相互关联的动词。

UML有多种连接器类型，用于表达所涉及的模型元素之间关系的性质。一些连接器，如关联定义了结构关系，而其他连接器，如控件，定义了时间的流逝。每种连接器类型都有一个符号，可帮助建模者识别连接器并理解其用途。

可以在各种窗口中查看连接器，例如关系窗口、层次结构窗口、关系矩阵、检验员窗口的“详细信息”选项卡和元素的“属性”对话框。

## 抽象



## 描述

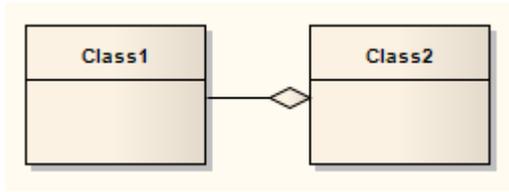
“抽象”是代表同一概念的两个元素之间的关系，无论是在不同的抽象级别还是从不同的观点。该图显示来自不同模型（领域模型和代码模型）的两个不同客户类别代表相同的概念。

“抽象”关系是“依赖”关系的子类型。

## 工具箱icon



## 聚合

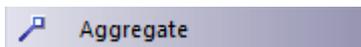


### 描述

聚合连接器是一种显示元素包含其他元素或由其他元素组成的关联类型。它用于类模型、包模型和物件模型中，以显示如何从一组更简单的元素（组件；例如，车轮、轮胎、电机等组成的汽车）构建更复杂的元素（聚合体）。

A更强的聚合形式，称为复合聚合，用于表示整体对其部分的所有权。该部分一次只能属于一个复合聚合。如果组合被删除，它的所有部分都会随之删除。

### 工具箱icon



## 更改聚合连接器Form

在您的建模中，当您创建聚合关系时，它默认为关系的弱（共享）形式，由空心菱形头表示。您可以将其更改为强形式（组合），由纯黑色菱形头表示。

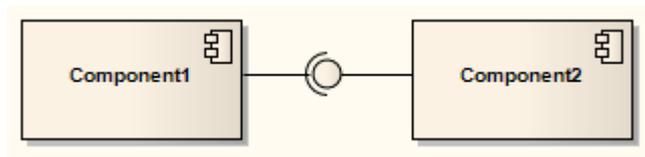
### 更改聚合连接器的形式从弱到强

节	行动
1	右键单击聚合连接器以显示上下文菜单。
2	选择设置聚合为复合。 菱形显示为实心。

### 注记

- 如果连接器已经是强（组合）连接器，则上下文菜单选项将更改为“设置聚合为共享”

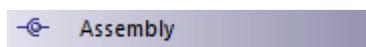
## 组装



### 描述

组装连接器将组件所需的接口 (Component1) 与另一个组件 (Component2) 提供的接口连接起来，通常在部件图中。

### 工具箱icon

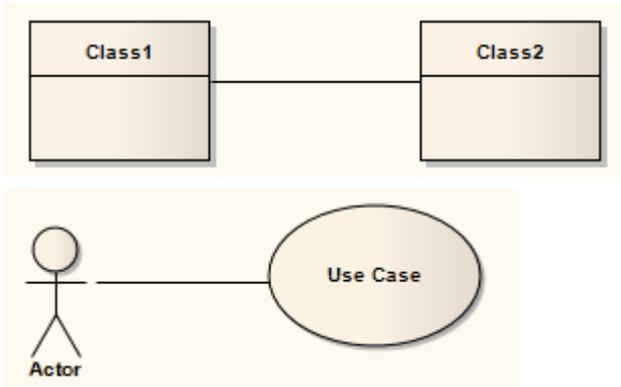


### OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.210) 指出：

部件连接器中的程序集连接器的执行时间部件是请求（信号和操作调用）沿着连接器的实例传播。

# 关联



## 描述

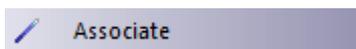
关联意味着两个模型元素具有关系，通常实现为一个或两个类中的实例变量。连接器可以在每一端包括命名角色、多重性、方向和约束。您还可以通过向连接器上的名称标签添加名称方向指示箭头来指示阅读方向（请参阅管理物件标签帮助帮助），并为绑定类和参数化类之间的关联连接器定义模板绑定参数。

关联充当为实例元素的结构提供可能的分类器的键，并在关联中的源SysML块元素上自动生成属性（部件）元素。

当为类图生成代码时，关联成为目标类中的成员变量。该关系还用于包物件、通讯、数据建模和部署图。

“关联”是两个元素之间的一般关系类型；要在一个关联中连接两个以上的元素，可以使用 N-Ary 关联元素。关联连接器也可以与类元素集成以形成关联类，以允许连接器具有定义某些类型的 UML 关系的操作和属性。

## 工具箱 icon



## OMG UML 规范：

OMG Unified Modeling Language 规范 (v2.5.1, p.199) 指出：

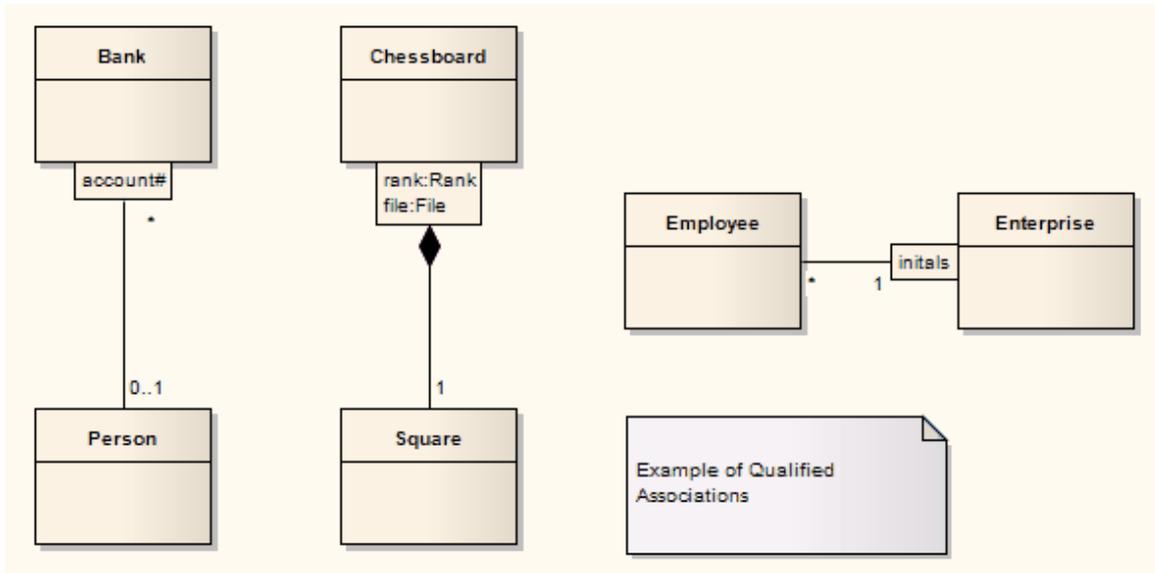
关联指定可以在类型化实例之间发生的语义关系。它至少有两个由属性表示的属性，每个都有 end 的类型。关联的多个末端可能具有相同的类型。

关联声明其类型符合或实现相关类型的实例之间可以存在链接。链接是 A 元组，关联的每个 memberEnd 都有一个值，其中每个值都是一个实例，其类型符合或实现末尾的类型。

# 限定词

限定词是关联部件、组件、端口或限制两个分类器或对象之间关系性质的属性的有序属性集。您在“定义词”对话框中定义一个限定词，通过单击关联、部件端口或属性“属性”对话框中“词”字段末尾的  按钮来显示该限定词。

## 例子



## 注记

- 在“属性”对话框的“限定符”字段中键入多个限定词时，用分号分隔它们；然后每个限定符显示在单独的行上；例如，在图中，限定符 'rank:Rank;file:文件' 已呈现为两行，在 ; 处有一个换行符。
- 您可以在“首选项”对话框的“图表”页面中启用或禁用限定符矩形（选择“开始”>外观>首选项>首选项>图表“功能区选项”）- 如果禁用，则使用旧样式文本限定词；不建议禁用限定词，因为它们是UML的组成部分
- 您可以在“首选项”对话框的“链接”页面中启用或禁用限定符矩形上的温和阴影

## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.129) 指出：

限定符相A限定端的实例声明关联实例集的分区（限定实例位于限定符所附的末端）。A限定词实例包括每个限定符属性的一个值。给定一个限定的物件和一个定义限定词实例，关联另一端的对象数量受声明的多重性限制。在多重性为0..1的常见情况下，限定符值相对于限定的物件是唯一的，并且最多指定一个关联object。通常在一般性0..\*的情况下，关联实例的集合被划分为子集，每个子集由给定的限定词实例选择。在多重性1或0..1的情况下，限定符具有语义和实现后果。在多重性0..\*的情况下，它没有真正的语义后果，但提出了一种便于轻松访问由给定限定符值链接的关联实例集的实现。

# 定义限定词对话框

定义限定词“关联用 定义连接连接器端、端口、部件或属性的定义限定词。

## 常规选项卡

审阅、编辑或完成表中所示的字段。

要更改限定词在 限定词”面板列表中的位置，请单击向上滚动按钮或向下滚动按钮（“手形”按钮）。

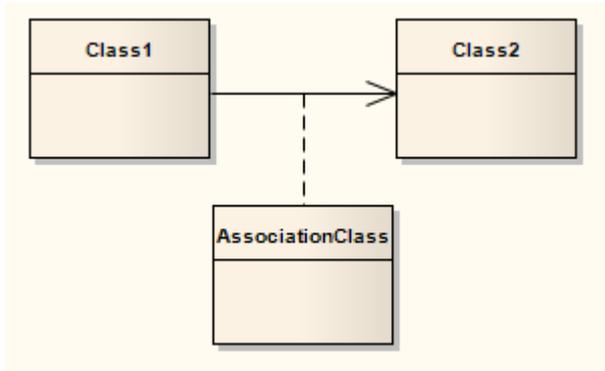
字段	行动
名称	显示限定符的名称。 对于新的限定符，输入名称（不带空格）。
别名	显示限定符的可选别名。 如有必要，输入新别名。
类型	显示限定符类型。 类型可以由代码语言（数据类型）或分类器元素定义。单击下拉箭头时，列表中的值集会提供适当的数据类型。  要选择或定义可能的分类器，请单击列表中的“选择类型”选项，或单击  按钮以显示“选择<项目>”对话框。 要添加可在此列表中显示的新代码语言数据类型，请参阅数据类型主题。
范围	将限定符定义为 <b>Public</b> 、 <b>Protected</b> 、 <b>Private</b> 或包。 如有必要，单击下拉箭头并选择不同的范围。
构造型	定义限定符的可选构造型。 如有必要，键入不同的原型名称或单击下拉箭头并选择一个原型。
衍生的	表示限定符是计算值。 如果选中此复选框，则元素上的限定符名称具有派生符号 (/) 作为前缀。
静止的	指示 <b>Qualifier</b> 是静态成员。
常量	表明 <b>Qualifier</b> 是一个常数。
最初的	显示一个可选的初始值。 如有必要，输入新的初始值。
注记	输入与限定词相关的任何自由文本注记。 您可以使用字段顶部的注记工具栏格式化注记文本。

## 详细信息选项卡

使用“详细信息”选项卡来模型选定限定符的其他属性，例如其属性多重性、重新属性和子集属性。  
在“常规”选项卡上选择一个限定符，然后如本表所示审阅、编辑或完成“详细信息”选项卡字段。

字段	行动
下限	定义集合中允许的元素数量的下限。
上限	定义集合中允许的元素数量的上限。
允许重复	表示允许重复。 映射到UML属性，值为 FALSE。
多重性是有序的	表示集合是有序的。
重新定义属性	审阅的重新定义属性。 通过单击“添加”按钮以显示“选择属性”对话框来添加重新定义的属性。
子集属性	审阅的子集属性。 通过单击“添加”属性以显示“选择属性”对话框来添加子集属性。

# 关联类



## 描述

关联类是使关联具有属性和操作（特征）的UML构造。这导致了具有关联和类特征的混合关系。

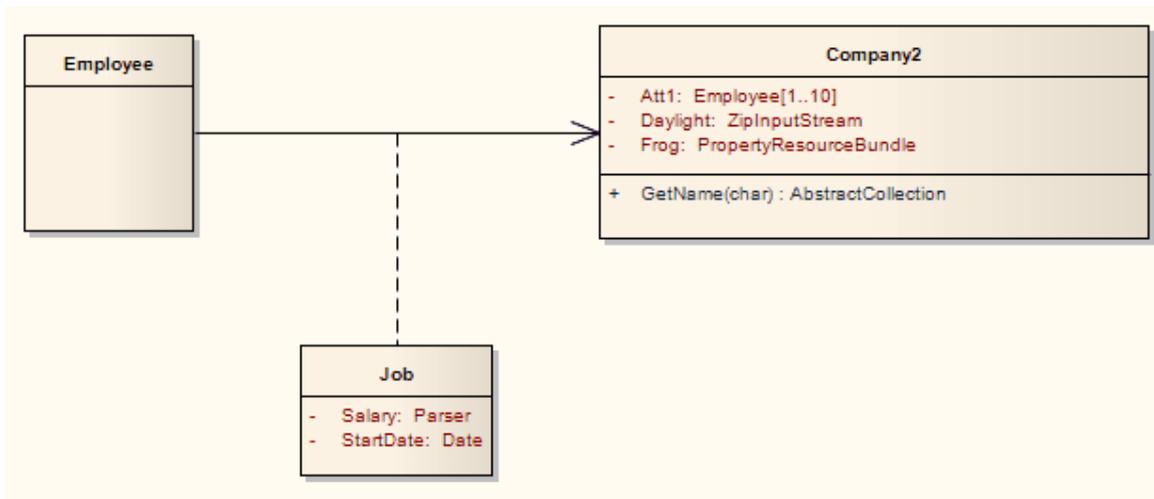
当您添加关联类连接时，Enterprise Architect还会创建一个自动连接到关联的类。当您隐藏或删除关联时，类也被隐藏或删除。

要将关联类添加到类或部署图中，请单击工具箱中的“关联类”图标。单击并按住图表中的源object，同时将线拖动到目标元素，然后释放鼠标按钮。Enterprise Architect绘制连接器并添加类，然后提示您添加类名。请记住，类和连接器的名称相同。您还可以将新类连接到现有关联。

通过选择关联连接器上的“查找关联类”上下文菜单选项，您可以在浏览器窗口中突出显示关联类的类部分。

## 示例

该图说明了模型元素之间的关联类。注记从类到关联的虚线。您不能移动或删除此行。

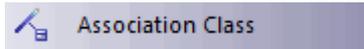


## 注记

- 如果您将带有形状脚本的构造型应用于关联类，请注意形状脚本适用于类部分和关联部分；因此，您可能必须在 shape main 中包含测试元素类型的逻辑，以便您可以为类和关联提供单独的绘图说明

- 这种逻辑在以下情况下是不必要的：
  - 形状源或形状目标，它们被类忽略，或者
  - 装饰形状，被关联连接器忽略
- 如果您将类与关联连接器分离，则两个部分都将保留其形状脚本，直到移除构造型

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.200) 指出：

AssociationClass 是一个关联的声明，它具有自己的一组特征。AssociationClass 既是一个关联类又是一个类，并保留了两者的静态和动态语义。一个AssociationClass描述了一组对象，每个对象都共享AssociationClass所包含的相同特征、约束和语义作为一种类，并对应于一个将AssociationClass实例化为一种关联的唯一链接。

# 将新类连接到现有关联

## 将类连接到关联

节	行动
1	在包含要连接的关联的图表中创建一个类。
2	右键单击新类并选择“高级 关联类”菜单选项。 将显示“创建关联类”对话框。
3	选择要连接的连接器的。
4	点击确定按钮。

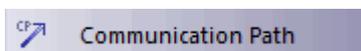
## 通讯路径



### 描述

通讯路径定义A两个 DeploymentTargets 能够交换信号和消息的路径。通讯路径是关联的特殊化。A DeploymentT 部署的目标，部署在一个部署工件中，可以是一个节点属性InstanceSpecSpec。

### 工具箱icon

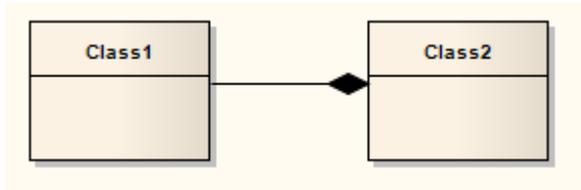


### OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.661) 指出：

通信路径是两个部署目标之间A关联，它们能够通过它交换信号和消息。

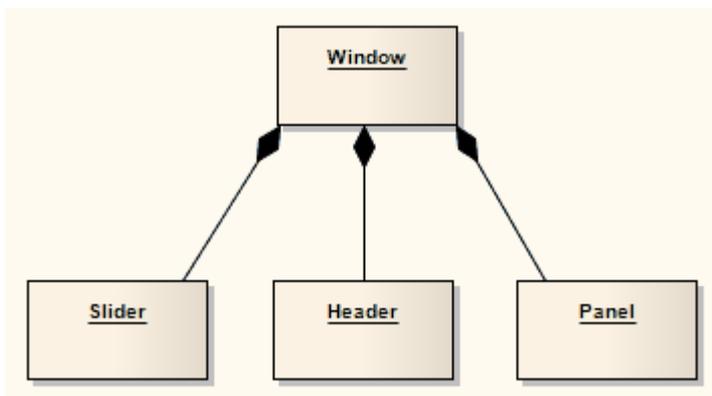
## 组合



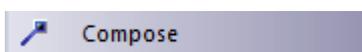
### 方向：

组合A描述由较小组件组成的元素，通常在类或包图中。A组件 - 或部分实例 - 一次最多可以包含在一个组合中。如果一个作品被删除，通常它的所有部分都会随之删除；但是，可以从合成中单独删除一部分，而不必删除整个合成。组合是传递的、不对称的关系并且可以是递归的。

### 示例



### 工具箱icon



### OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.112) 指出：

复合聚合是一种强大的聚合形式，它要求一个部分object一次最多包含在一个复合object中。如果删除了复合object，则它的所有作为对象的零件实例都将随之删除。

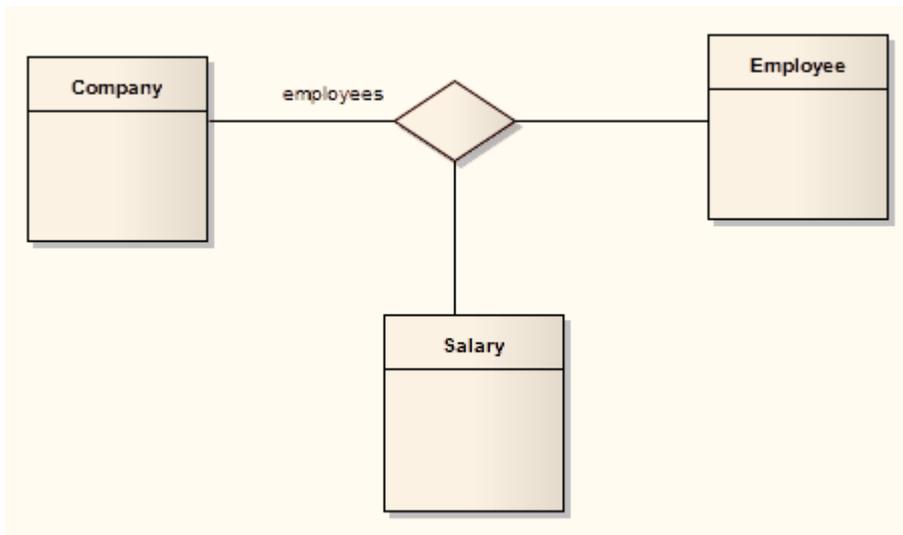
组合可以在具有传递删除特征的有向无环图中链接；即删除图中某一部分的object，也会导致删除该对象下方子图的所有object。有意未指定复合聚合的精确生命周期语义。故意未定义创建组合对象的顺序和方式。当容器或部件由 DataType 键入时，复合聚合的语义没有被特意指定。

## N-Ary关联



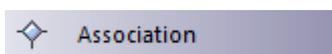
### 描述

n-模型关联元素用于模拟三个或更多元素之间的复杂关系，通常在类图中。它不是一种常用的设备，但可以在几个元素之间存在依赖关系的情况下使用，效果很好。它通常与关联连接器一起使用，但关系可以包括其他类型的连接器。



在此示例中，公司、员工和薪水之间存在关系。

### 工具箱icon

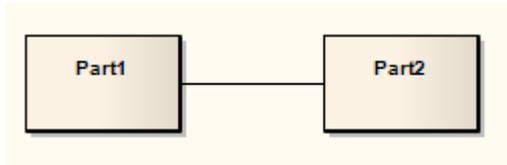


### OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.201) 指出：

任何关联都可以绘制为菱形（大于一条线上的终止符），每个关联成员用一条实线End 将菱形连接到作为端的类型的分类器。多于两端的关联只能这样绘制。

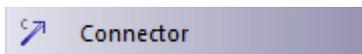
## 连接器



### 描述

连接器说明了部件之间的通信链接以实现结构的目的，通常在复合结构图中。每个连接器端都是不同的，控制与其连接元素相关的通信。这些元素可以定义指定此行为的约束。连接器可以有多样性。

### 工具箱icon

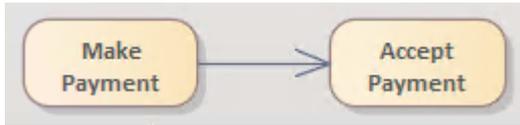


### OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.227) 指出：

A连接器指定了能够在两个或多个实例之间进行通信的链接。与指定关联分类器的任何实例之间的链接的关联相反，连接器指定仅播放连接部分的实例之间的链接。

# 控件



## 描述

控件是一个连接活动图中两个节点的连接器，对活动转换进行建模。控件Flow 连接器在活动节点之间桥接流，一旦源节点完成后将流引导到目标节点。

流控件和物件流可以定义一个守卫条件和一个权重条件。

守卫条件定义了一个条件，A条件在控制通过该活动边缘之前必须为True。A实际的例子是两个或多个活动边缘（控件流）从一个决策元素中退出。每个流都应该有一个独立于其他流的守卫条件条件，并定义在什么条件下采用哪条边。控件的“属性”对话框使您能够在控件流和物件流上设置守卫条件条件。

权重定义A在遍历该边时可以沿控件或物件流连接流动的令牌的数量。重量也可以在控件流和物件流的“属性”对话框中定义。

## 工具箱icon

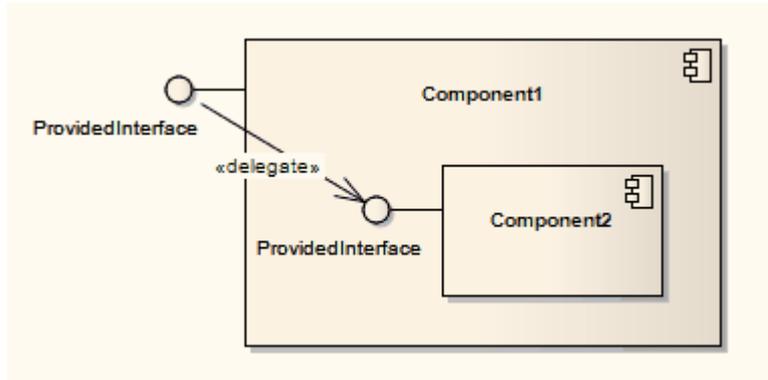


## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.376) 指出：

ControlFlow 是A仅传递控制令牌（以及建模者指定的一些object令牌，...）的ActivityEdge。ControlFlows 用于显式序列ActivityNode 的执行，因为目标ActivityNode 无法接收控制令牌并开始执行，直到源ActivityNode 完成执行并产生令牌。

# 代表



## 描述

A委托连接器在一个部件上定义了一个组件的外部端口图和部件图。使用委托连接器通过外部接口连接的委托将系统的内部工作连接到外部世界。

## 工具箱icon



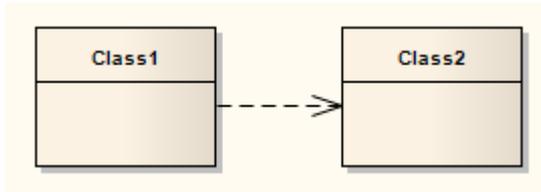
## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.191) 指出：

委托连接器是将端口链接到所属封装分类器中的角色A连接器。它代表请求的转发（操作调用和信号）。到达具有一个或多个属性或端口的属性连接器的端口A请求将被传递到那些目标以进行处理。

委托连接器可用于模型行为的分层分解，其中由封装分类器提供的服务最终可能由嵌套在其中的多个级别的服务来实现。

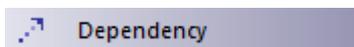
# 依赖



## 描述

依赖关系用于模拟模型元素在用例、活动和结构图中，甚至模型本身之间的广泛依赖关系。您可以从工具箱的公共页面创建依赖关系。UML 2.1中定义的 Dependencies包有许多派生类，例如实现、部署和使用。一旦你创建了一个依赖，你可以通过应用一个专门的原型来进一步细化它的含义。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.42) 指出：

依赖关系是表示单个模型元素或一组模型元素需要其他元素模型来规范或实现的关系。这意味着客户端元素的元素完成语义要么在语义上要么在结构上依赖于供应商的定义。

# 应用构造型

本主题定义如何将构造型应用于依赖关系。

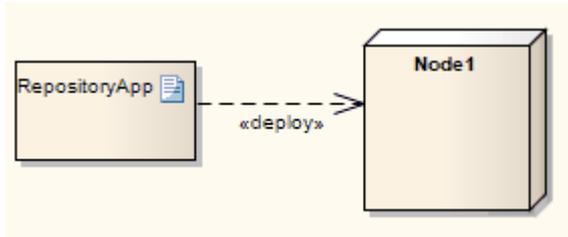
## 应用构造型

节	行动
1	选择要更改的依赖关系。
2	右键单击连接器并选择“依赖属性”选项。 将显示“依赖属性”对话框。
3	在“构造型”字段中，输入所需的构造型名称或单击下拉箭头并从列表中选择构造型。
4	点击确定按钮。

## 或者

右键单击依赖关系并选择“高级|依赖构造型”选项，然后从较短的标准构造型列表中进行选择。

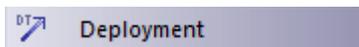
## 部署



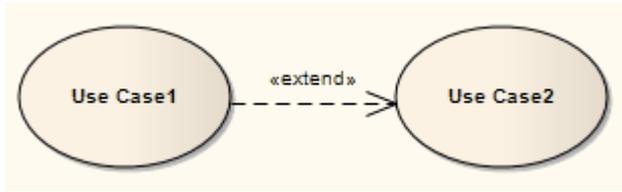
### 描述

A Deployment 类型的工件关系是一个部署关系到一个节点或一个可执行目标的部署，通常在一个 Deployment 图中。可以在类型和实例级别部署。在类型级别，将为节点的每个实例进行部署。还可以为节点的实例指定部署，以便节点的实例可以具有不同的部署工件。使用定义为部件的节点建模的复合结构，部件也可以用作部署关系的目標。

### 工具箱icon



# 延长



## 描述

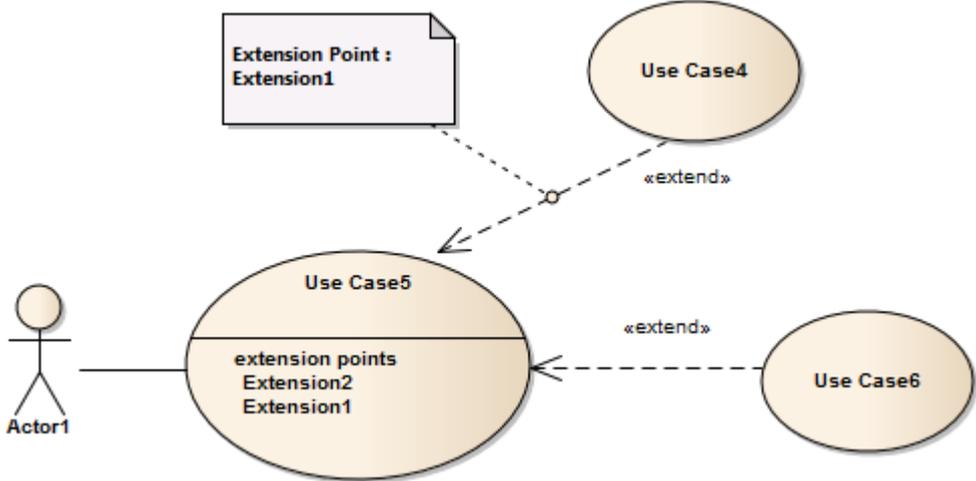
Extend 连接器用于指示一个元素扩展了另一个元素的行为，主要在用例模型中，其中一个用例（可选）扩展了另一个用例的行为。扩展用例通常表示与扩展用例的行为集成的替代流，在扩展点标识的元素内的行为流中的特定点处。扩展点由文本string表示，例如“启动时”或“建立连接之前”。

A用例可以有多个扩展点，并且可以扩展或被多个其他用例扩展。扩展使用案例、扩展使用用例和扩展应用点之间的精确关系可以在扩展关系上标识，如图所示。

## 识别扩展点

节	行动
1	<p>右键单击扩展连接器并选择 高级 扩展点  设置扩展点'选项。 显示 无素用途”对话框，列出目标中当前定义的扩展点用例元素。</p>
2	<p>单击源用例作用的扩展点，然后单击打开按钮。 对话框关闭，Extend 连接器在中点显示一个小圆圈，并带有注释链接到用于标识所选扩展点的注记元素。</p> <p>( 注记最初可能不会显示在扩展连接器附近 - 检查图表的左上角并将注记拖到您希望它占据的位置。 ) 使用这些相同的步骤来更改注记中标识的扩展点。</p>

### 显示/隐藏扩展点注记

节	行动
1	<p>右键单击扩展连接器并选择 高级 扩展点  显示扩展点'选项。</p> <p>如果在选定的延伸连接器上识别出任何扩展点，它们将如图所示显示。</p>
2	<p>右键单击扩展连接器并取消选择 高级 扩展点  显示扩展点'选项。</p> <p>在选定的延伸连接器上标识的任何扩展点都被隐藏，如下所示：</p> 

### 工具箱icon



### 注记

- Extend 连接器与扩展连接器不同，后者在配置文件图中用于指示一个构造型元素扩展一个元类或另一个构造型元素；两种连接器外观不同

### OMG UML规范：

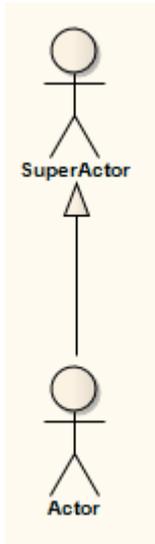
OMG Unified Modeling Language规范 (v2.5. 1 , pp.640-641) 指出：

扩展是从扩展用例（扩展）到扩展用例（extendedCase）的关系，它指定扩展用例中定义的行为如何以及何时可以插入到扩展用例中定义的行为中。扩展发生在扩展用例中定义的一个或多个特定扩展点。当有一些额外的行为可能有条件地添加到一个或多个 UseCases 中定义的行为时，应使用 Extend。扩展用例的定义独立于扩展用例，并且其意义独立于扩展用例。另一方面，扩展用例通常定义本身不一定有意义的行为。相反，扩展用例定

义了一组模块化行为增量，这些增量在特定条件下增强了扩展用例的执行。

笔记。同一个扩展用例可以扩展多个用例。此外，一个扩展用例本身可以被扩展。

# 概括

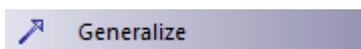


## 描述

A概括用来表示继承。从特定分类器到一般分类器，泛化的含义是源继承了目标的特征。它通常用于类、部件、物件、包、用例和需求图中。

您还可以为绑定类和参数化类之间的连接或定义模板概括参数。

## 工具箱icon

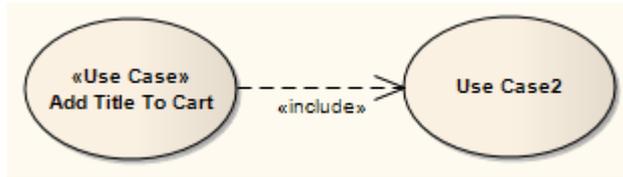


## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.138) 指出：

A概括一个更一般的分类器和一个更具体的分类器之间的分类关系。特定分类器的每个实例也是一般分类器的一个实例。特定分类器继承了更一般分类器的特征。A概括特定的分类器。

# 包括



## 描述

Include 连接表明源元素包含目标元素的功能。在用例模型中使用包含连接来反映一个用例包含另一个用例的行为。使用包含关系以避免在许多使用案例中具有相同的行为子集；这类似于类模型中使用的委托。

## 工具箱icon



## OMG UML规范：

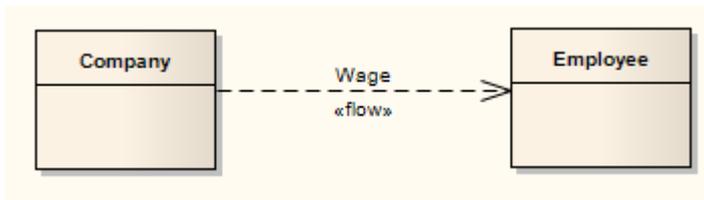
OMG Unified Modeling Language规范 (v2.5. 1 , p.641) 指出：

Include是两个UseCases之间的DirectedRelationship，表示将被包含用例（加法）的行为插入到包含用例例（includeCase）的行为中。它也是上下文的一种，因此它可以在其拥有的时间用例（包括Case）中具有名称。包含用例可能取决于执行包含用例所产生的变化。包含用例必须可用于包含用例的行为才能被完整描述。

Include关系旨在在两个或多个 UseCases 的行为有共同部分时使用。然后将这个公共部分提取到一个单独的用例中，以包含在所有具有该部分公共的基本用例中。由于 Include关系的主要用途是重复使用公共部分，所以保留在基本用例中的内容通常不是完成本身，而是依赖于包含的部分才有意义。这反映在关系的方向上，表明基础用例取决于加法，反之则不然。

在恢复执行包含用例用例之前，包含用例的所有行为都在包含用例中的单个位置执行。

# 信息流



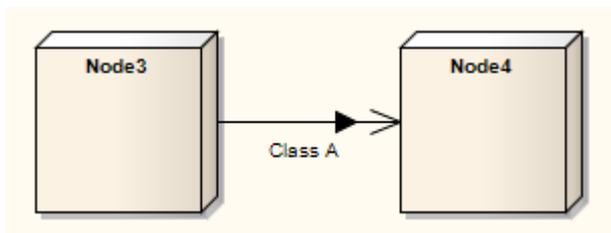
## 描述

信息流表示任何图表中两个元素之间的信息项（信息项元素或分类器）的流动。该连接器可从以下渠道获得：

- 工具箱的“公共”页面
- 每个快速链接菜单，以及
- 自动同时直接定义信息项实现

当您创建信息流连接器时，Enterprise Architect会自动提示您确定要传达的信息项。

您可以在相同的两个元素之间拥有多个信息流连接器，以识别在不同条件下哪些项目在元素之间流动。连接器可以单向流动：

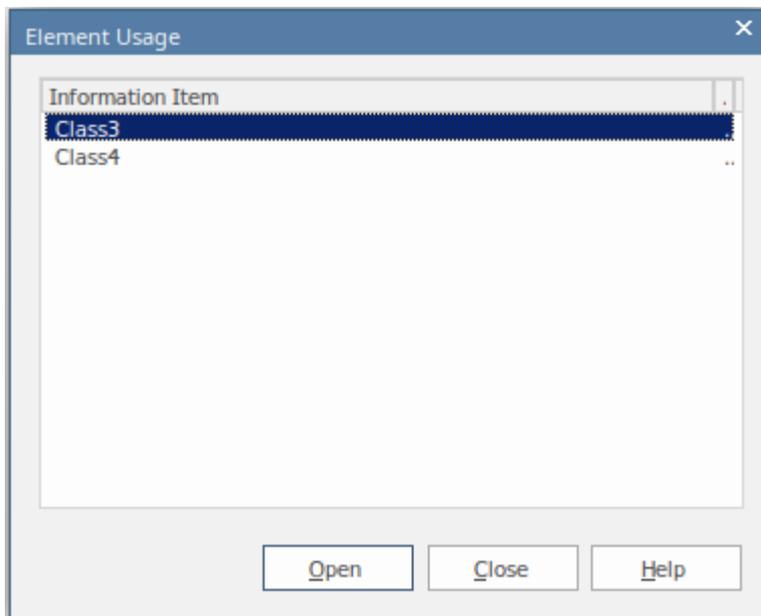


它们也可以定义为沿相反方向流动：



有关更多详细信息，请参阅使用信息流帮助主题。

您可以通过右键单击连接器并选择“查找传送项”选项来定位任何信息流中传送的项目。



## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.669) 指出：

信息流以一般的方式描述信息通过系统的循环。它们没有具体说明信息的性质、传递信息的机制、交换顺序或任何控制条件。在更详细的建模过程中，可以添加表示和实现链接，以指定哪些模型元素实现了信息流并显示信息是如何传递的。

OMG Unified Modeling Language规范 (v2.5. 1 , p.670) 还指出：

InformationItem 表示可以以非常抽象的方式从源流向目标的多种信息。它们代表可能在系统内移动的信息类型，但不详细说明传输信息的细节。传输信息的细节是最终定义信息项的其他分类器的范围。

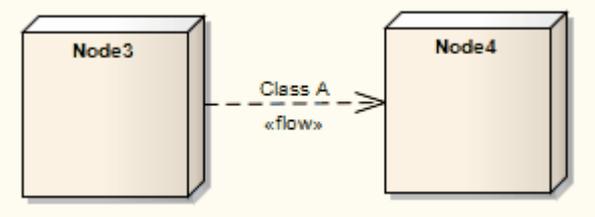
# 使用信息流

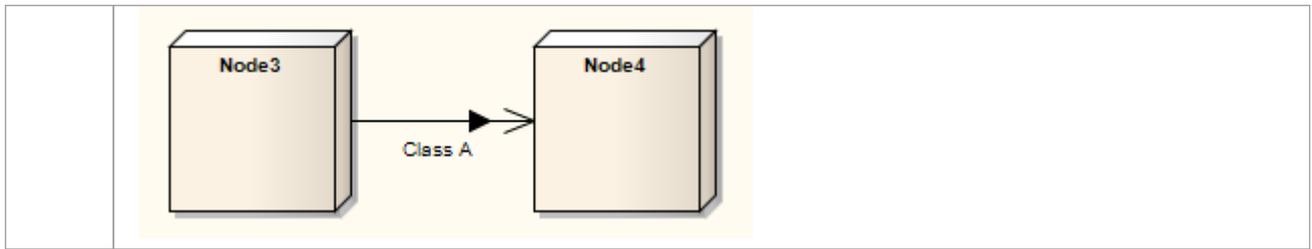
当您在图表上的两个元素之间拖动信息流连接器时，Enterprise Architect会自动提示您识别所传达的信息项。

您还可以在直接定义信息流实现的同时自动创建信息流，就像在序列图上的信息上所做的那样。

如果您倾向于在一个工作会话中、分类器之间以及端口和/或部件之间立即创建、填充和实现信息流，您将欣赏这样一个事实，即您现在可以在“信息项”上快速且完全地设置信息流Conveyed”对话框，使用“Add Realizing关系”复选框。

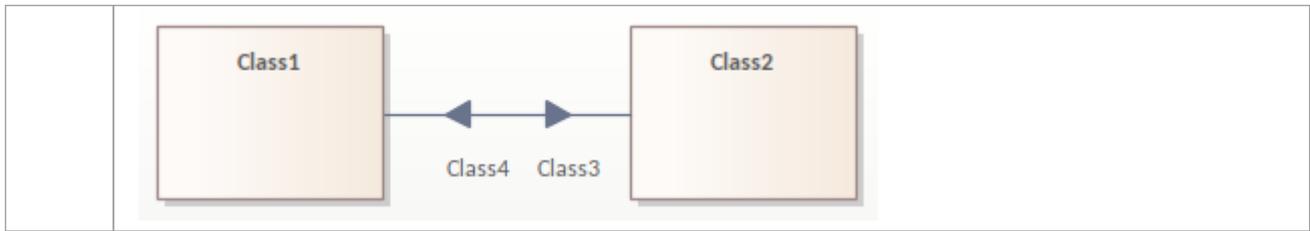
## 创建和实现信息流

节	行动
1	打开一个图表并添加两个元素（例如，部署图表上的节点）。
2	单击工具箱“公共”页面中的信息流连接器，并在两个元素之间拖动光标。 将显示“信息项传送”对话框。
3	将分类器或信息项元素到信息流中。如果您现在无法完成，您可以稍后返回信息流并添加剩余的信息项。
4	<p>如果您在两个分类器元素（例如类、部件或用例）之间或在端口和/或部件之间创建信息流，则启用“添加实现关系”复选框（尽管不适用于其他组合）。</p> <p>如果您已完成向流程分配信息项，请选中此复选框并单击确定按钮：</p> <ul style="list-style-type: none"> <li>对于两个 Classifier 之间的关系，两个元素之间会自动创建一个新的关联连接器，实现信息流</li> <li>对于端口和/或Parts之间的关系，在两个元素之间自动创建一个新的Connector类型的连接器，实现信息流</li> </ul> <p>在这两种情况下，请参阅步骤 7 中的图示。</p> <p>否则，单击确定按钮关闭对话框。该图现在类似于此示例：</p> 
5	<p>如果您已将信息项分配给流，请在相同的两个元素之间添加另一个连接器（例如，通讯路径连接器）。</p> <p>如果您还没有完成分配信息项，您可以稍后返回到实现，或者使用“已实现的信息流”对话框，或者 - 对于分类器之间和端口和/或部件之间的流 - 返回到“已传达的信息项”对话框并选择“添加实现关系”复选框。</p>
6	右键单击连接器并选择“高级 “信息流已实现”选项。 将显示“已实现信息流”对话框。
7	勾选已实现流程中每个所需信息项的复选框，然后单击确定按钮。 连接器现在类似于此示例，其中黑色三角形表示信息流连接器的存在和方向：



### 创建双向信息流

节	行动
1	<p>打开一个图表并添加四个元素（例如四个类）。</p>
2	<p>使用快速连接器，将一条信息流从 Class1 拖到 Class2 中。 信息项传送对话框出现。</p> <ul style="list-style-type: none"> <li>单击添加按钮并选择 Class3。</li> <li>确保未勾选“添加实现关系”选项。</li> <li>点击确定。</li> </ul> <p>将连接器从中心移开，以确保下一个连接器的可见性。</p> <p>A UML diagram showing two classes, Class1 and Class2, as rectangles. A dashed arrow labeled '«flow»' points from Class1 to Class2, with 'Class3' written below it.</p>
3	<p>使用快速连接器，将一个信息流从 Class2 拖到 Class1 。</p> <p>信息项传送对话框出现。</p> <ul style="list-style-type: none"> <li>单击添加按钮并选择 Class4。</li> <li>确保未勾选“添加实现关系”选项。</li> <li>点击确定。</li> </ul> <p>将连接器从中心移开，以确保下一个连接器的可见性。</p> <p>A UML diagram showing two classes, Class1 and Class2, as rectangles. Two dashed arrows labeled '«flow»' connect them: one from Class2 to Class1 labeled 'Class4', and one from Class1 to Class2 labeled 'Class3'.</p>
4	<p>使用快速连接器，在 Class1 和 Class2 之间拖动关联。</p> <ul style="list-style-type: none"> <li>右键单击关联并选择：高级 实现信息流</li> <li>在 <i>Information Flows Realized</i> 对话框中，选择 Class3 和 Class4 并单击确定按钮。</li> </ul>



## 注记

- 一旦实现了信息项信息流，您将无法直接访问“信息传递”对话框；要在连接器上添加或删除信息项，请在“信息项已实现”对话框中“取消实现”连接器
- 如果元素之间有多个信息流连接器，它们构成同一个组合连接器的一部分；您可以通过“信息项已实现”对话框再次单独处理它们
- 如果图表中有信息流用作模式的源，则不会将“Information项”和“Information Flows Realized”数据复制到模型中
- 您可以在浏览器窗口中，使用连接器上的“查找传递的项”上下文菜单选项，在信息流连接器上定位分类器或信息元素

## 传达有关流程的信息

当您在两个元素之间创建信息流连接器时，Enterprise Architect会自动提示您指定在此流上传送哪些信息项或分类器。如果您没有立即意识到其现有信息项的信息流，您可以更改和/或添加到稍后传达的信息项。菜单路径可帮助您返回到不完整的信息流，但流程步骤适用于新的和未完成的流。

### 访问

上下文菜单	右击信息流连接器 高级 传达的信息项
图表	从浏览器、图表工具箱或图表本身拖动分类器并将其放到信息流连接器上，以将该分类器添加为传送项目

### 指定信息流上传达的信息项

节	行动
1	在“信息项传送”对话框中，单击“添加”按钮。 将显示“选择分类器”对话框。
2	浏览或搜索所需的信息项或分类器元素或元素，并根据需要选择它们。 如果您不想在“选择分类器”对话框中保留所选项目，请按 Ctrl 并单击该项目。
3	单击确定按钮返回到“信息项传送”对话框。 您选择的每个信息项都列在对话框中的单独一行上。
4	如果您不想保留此对话框中的选定项目，请单击它并单击“删除”按钮。 对于分类器之间或端口和/或部件之间的链接，如果信息流已完成，则可以通过选择“添加实现关系”选项来创建实现连接器。对于其他类型元素之间的信息流，可以单独实现。 单击确定按钮关闭对话框并在信息流连接器标签上显示选定的信息项元素名称。

# 实现信息流

创建UML信息流连接器后，您可能希望：

- 在信息流连接器上实现一个或多个现有流
- 在信息流连接器上编辑现有流

您可能还想在非信息流连接器上创建和实现信息流，例如序列图上的信息。您可以使用 **已实现信息流** 对话框执行这些操作，该对话框显示可在所选连接器上实现的所有现有流。

对于分类器之间或端口信息和/或路径之间的关系，在完成将流项分配给信息流后，您可以通过在 **信息项传送** 对话框中选择 **添加实现关系** 选项来自动实现它。

## 访问

上下文菜单	右击连接器 高级 实现的信息流
-------	-----------------

## 审阅Item Flows on an信息流连接器

手术	行动
选定连接器上的实现信息流	选择每个所需流程的复选框，然后单击确定按钮。
取消流程的实现	取消选中相应流程的复选框，然后单击确定按钮。
更改信息流上传达的分类器或信息项元素	单击相应的信息项行，然后单击其右侧的  按钮。 选择分类器”对话框显示： <ul style="list-style-type: none"> <li>• 单击单个项目以将其选中</li> <li>• Ctrl+单击几个项目中的每一个以将它们全部选中，或者</li> <li>• Ctrl+单击选定的项目以取消选择它</li> </ul> 单击确定按钮返回 <b>已实现信息流</b> 对话框，如果需要，实现已更改的流。
直接在新连接器上创建已实现的信息流	1. 右键单击连接器并选择 <b>已实现信息流</b> 选项。 2. 单击 <i>Click to create new information flow...</i> 文本。将显示 <b>选择分类器</b> 对话框。 3. 选择所需的分类器或信息项元素，然后单击确定按钮返回 <b>已实现信息流</b> 对话框；选定的元素首先在对话框中列出，并勾选了激活复选框。 4. 单击确定按钮返回图表；连接器现在显示为已实现的信息流，其中选定的分类器或信息项元素在连接器标签中命名。

## 注记

- 如果有多个信息流并且您没有实现所有它们，则那些没有实现的由信息流连接器的单独的信息项

Conveyed迭代表示；您只能在原始连接器上实现这些流，此时流在该原始连接器上表示

- 如果您意识到所有的流程，它们会组合在一条连接线上
- 如果您在连接器上实现了信息流，您可以使用“查找传送项”上下文菜单选项在浏览器窗口中定位相应的信息流项
- 您仅在UML连接器上实现信息流；您无法在 ArchiMate 连接器上实现信息流，因此没有为它们提供菜单选项

# 中断流



## 描述

Interrupt Flow 是一个连接，用于定义 Exception Handler 和 Interruptible 活动区域的连接器的两个UML概念。中断流是一种活动边缘。它通常用于活动图表中，对活动过渡进行建模。

## 工具箱icon

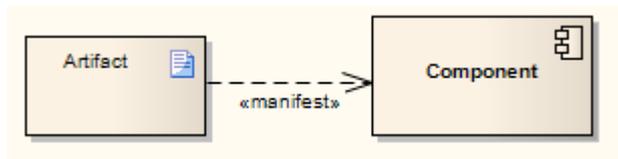


## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.375) 指出：

ActivityEdge 是两个 ActivityNode 之间的有向连接，令牌可以沿着它流动，从源ActivityNode 到目标 ActivityNode。

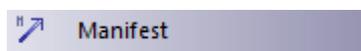
## 显现



## 描述

A元素关系表明了Enterprise Architect工件源模型的最佳体现元素，通常在部件和部署图中。可以添加到构造型类型以对模型的表现类型进行分类

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.657) 指出：

一个工件体现或表现出许多模型。工件拥有每个Manifests，代表了一些PackageableElement的使用情况。

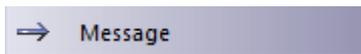
Profiles可以扩展 Manifestation关系以指示特定形式的实施例。例如， “生成的”和 “可能是两个自定义代码”，用于不同的类中工件的 “自定义代码”。

# 信息

消息指示元素之间的信息流或控制转换。消息可以在交互图、序列图、通讯图、系统视图图中使用。如果在类或分类器实例之间，相关的操作列表可用于指定事件。

移动信息可能会破坏图表上其他特征的组织。为避免这种情况并仅移动信息，请在移动信息时按 Alt。

## 工具箱icon



## OMG UML规范：

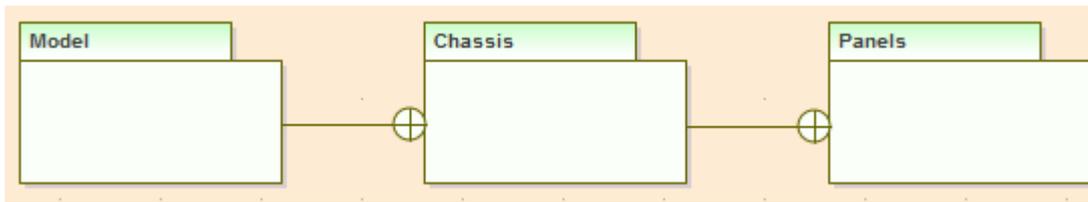
OMG Unified Modeling Language规范 (v2.5.1, p.623) 指出：

信息定义了交互之间A特定通信。

OMG Unified Modeling Language规范 (v2.5.1, p.574) 还指出：

信息的签名指的是操作或信号。信息的名称必须与引用的操作或信号的名称相同。

## 嵌套



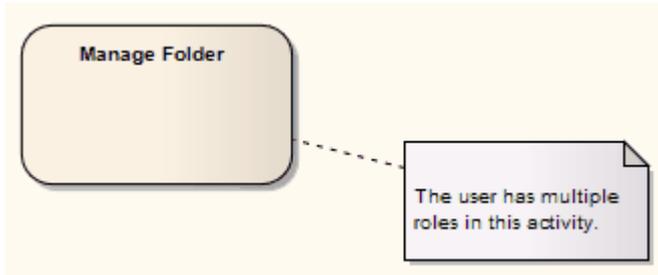
### 描述

嵌套连接器的其他元素是一种替代图形，而不是表示包含或嵌套元素内的元素。最适合用来展示嵌套的包包了。

### 工具箱icon



## 注释链接

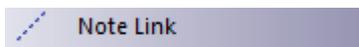


### 描述

注释链接连接器A注释注记连接到任何其他类型的一个或多个其他元素。

注记和注释链接都可以在公共页面的任何类别的工具箱中找到。您还可以从UML元素工具栏中选择它们。

### 工具箱icon

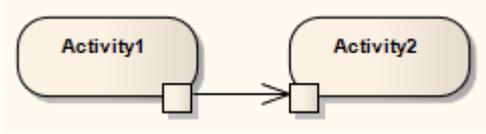


### OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.22) 指出：

注解A为右上角弯曲的矩形（这也称为“注解符号”）。矩形包含注解的注解。每个 annotatedElement 的连接由单独的虚线显示。连接注记符号和annotatedElement(s)的虚线如果从上下文很清楚，或者在此图中不重要，则可能会被抑制。

# 物件流



## 描述

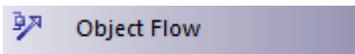
物件流用于活动图和状态机图。当在活动图中使用时，一个物件流连接两个元素，特定数据通过它，模拟一个活动过渡。要使用物件流查看示例活动图，请参阅活动图表主题活动图物件流。

在状态机图中，物件流是状态流或转换的规范。它意味着在运行时在元素之间传递一个物件实例。

您可以从工具箱的“状态”或“活动”页面插入工具箱物件流，也可以从位于标题工具栏中的所有关系的下拉列表中插入物件流。您还可以通过选择连接“属性”对话框上的“对象流”复选框来修改与物件流的转换连接。

有关在物件流上设置 Guards 和物件的信息，请参阅控件流主题。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.376) 指出：

ObjectFlow 是一个 ActivityEdge，它可以让Object令牌沿它传递。对象流模型对象节点之间的值流。令牌以与源提供的顺序相同的顺序提供给目标 ActivityNode。如果同时提供多个代币，则这些代币的提供顺序与从源一次提供一个的顺序相同。如果源是一个指定了 ordering 的物件节点，则源中的令牌以该顺序提供给 ObjectFlow，因此，ObjectFlow 以相同的顺序提供给目标

# 活动图表中的物件流

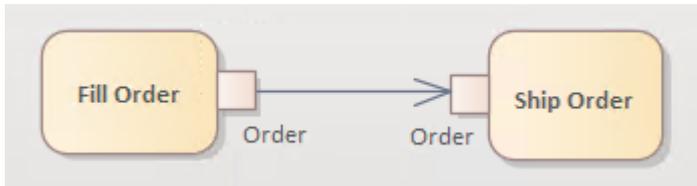
在活动图中，有多种方法可以定义对象之间的数据流。

该图描述了两个操作（“填写订单”和“发货订单”）之间的简单物件流，这两个操作均访问订单信息。



参见 OMG Unified Modeling Language规范，（v2.5.1，图 12.110，第 391 页。）

可以使用此格式细化对通过两个物件流连接到活动的数据object订单的明确描述。这里使用行动销来反映订单。



参见 OMG Unified Modeling Language规范，（v2.5.1，图 12.110，第 391 页。）

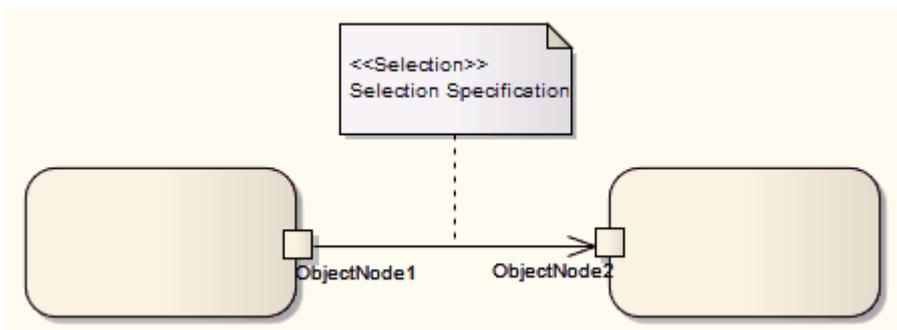
该图是两个动作之间多个物件流交换数据的示例。



参见 OMG Unified Modeling Language规范，（v2.5.1，图 12.111，第 391 页。）

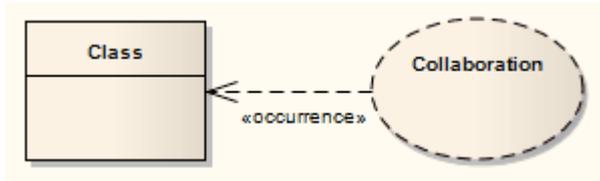
选择行为和转换行为共同构成一种查询，可以指定物件流数据访问的性质。选择行为确定哪些对象会受到连接的影响。然后，变换行为可能会进一步指定与选定object有关的属性的值。

通过将注记附加到物件流可以定义选择和转换行为。为此，请右键单击物件流并选择“附加注记或约束”选项。如果行为适用于多个流，A会列出图表中您可以选择附加注记的其他流。为了符合UML 2，请在行为前面加上符号“选择”或“转换”。



参见 OMG Unified Modeling Language规范，（v2.5.1，图 12.112，第 392 页。）

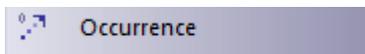
## 发生



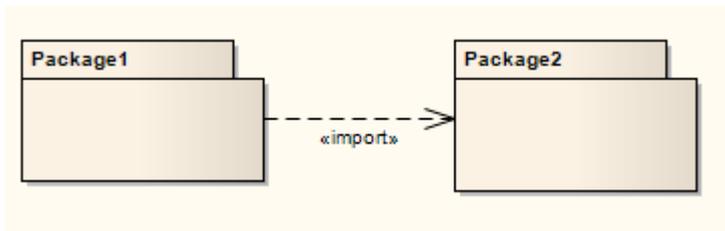
## 描述

一个协作关系表示一个类图代表一个分类器，在一个复合结构图中。一个协作连接器是从协作器到分类器的。

## 工具箱icon



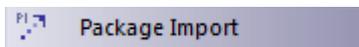
## 导入包



### 描述

导入包关系是从A源包到一个内容要被导入的包。无法导入目标包的私有成员。关系通常用在包图中。

### 工具箱icon

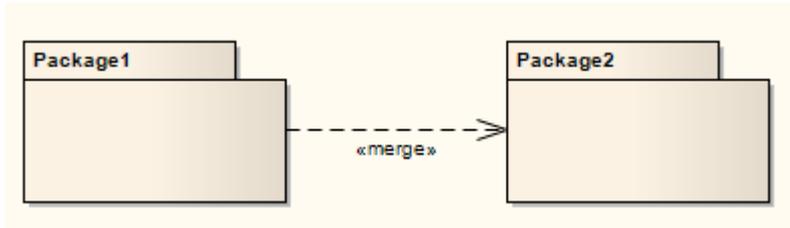


### OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , pp.28-29) 指出：

PackageImport 是A导入的命名空间和一个包之间的有向关系，表示导入的命名空间将包的成员的名称添加到它自己的命名空间中。从概念上讲，包导入相当于对导入的命名空间的每个单独成员都有一个 ElementImport，除非有单独定义的 ElementImport。如果有一个元素的 ElementImport，那么这优先于通过 PackageImport 可能导入的相同元素。

# 包合并



## 描述

在包图中，包合并表示两个包之间的关系，目标包的内容与源包的内容合并。不合并目标包的私有内容。包 Merge 的适用性解决了多个包包含相同名称的元素的情况，代表相同的事物。A 合并其合并包中的所有匹配元素，以及它们的关系和行为。注记，包本质上是对所有匹配元素进行概括和重新定义，但合并包包独立元素表示仍然存在且不受影响。

包合并 Enterprise Architect 中用于图形目的，但会创建应用于相关包的有序包关系（可以在包的“属性”对话框中的“链接”选项卡下看到）。这种关系可以反映在 XMI 导出或 Enterprise Architect Automation 接口脚本中，用于代码生成或其他模型驱动架构 (MDA) 利益。

合并关系有助于反映现有架构包含涉及类似元素的功能的情况，这些元素被合并到一个正在开发的包架构中。合并不影响合并对象，支持产品级联的常见情况。

## 工具箱 icon



## OMG UML 规范：

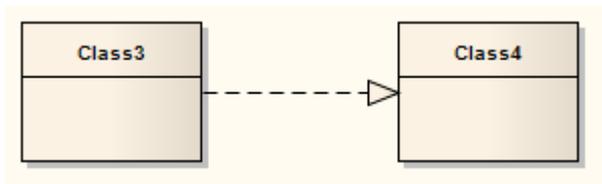
OMG Unified Modeling Language 规范 (v2.5.1, p.242) 指出：

PackageMerge 是两个包之间 A 有向关系，表示目标合并包的内容根据下面定义的一组规则组合到源接收包中。它与源元素的概念非常相似，概括在概念上将目标元素的特征添加到其自身的特征中，从而形成一个结合了两者特征的元素。正如子类通常不使用其继承的特征来描述一样，接收包通常不会使用来自其合并包的合并元素来描述。

此外，与合并概括，包本身（可能不会直接或间接地）合并。

此功能旨在在不同包中定义的元素具有相同名称并旨在表示相同概念时使用。A 给定的基本概念可以被合并用于不同的目的，每个目的都定义在一个单独的接收包中。通过选择不同的接收包，可以获得针对特定目的的概念的自定义定义。

## 实现



## 描述

源object实现或实现其目标object A实现连接器用于用例、部件或需求图中，以表示模型中的可追溯性和完整性。A业务流程或需求是由一个或多个使用案例来实现的，这些案例又是由一些类来实现的，而这些类又是由一个部件来实现的，等等。在您的系统设计中映射需求、类等，直至建模抽象级别，确保系统的大图记住并反映约束和定义它的所有小图和细节。

您还可以为绑定类和参数化类之间的实现连接器定义模板绑定参数。

## 工具箱icon

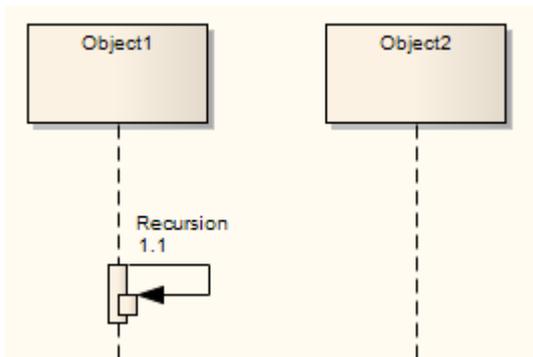


## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, pp.38-39) 指出：

实现是两组 NamedElements 之间的一种专门的抽象依赖关系，一组表示规范（供应商），另一组表示该规范的实现（客户端）。实现可用于模型的逐步细化、优化、转换、模板、模型综合、框架组合等。实现意味着客户端集合是供应商集合的实现，作为规范。实现”的含义并没有严格的定义，而是在一定的建模上下文上暗示了一种更精细或更精细的形式。可以发

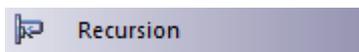
# 递归



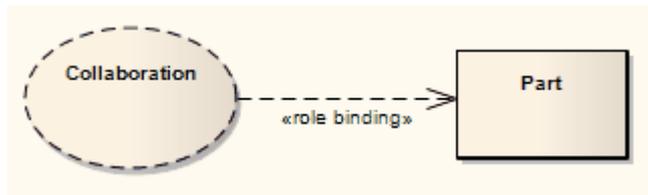
## 描述

递归是序列图中用于指示递归函数的A信息。

## 工具箱icon



## 角色捆绑

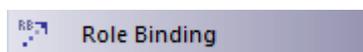


### 描述

角色捆绑是协作应用内部角色与实现特定情况所需的各个部分之间的映射，通常在复合结构图中。关联的部件可以定义属性以使绑定发生，并且协作发生。

角色捆绑连接器在A和分类器履行角色之间协作，协作器的内部绑定角色标记在分类器端。

### 工具箱icon

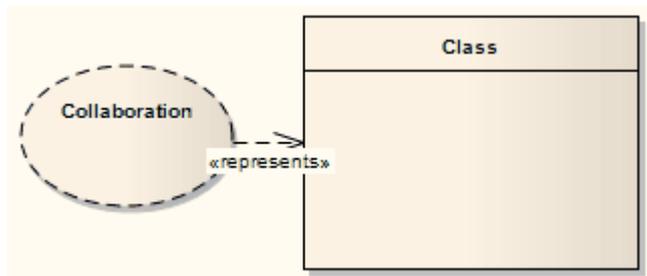


### OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.216) 指出：

角色绑定是使用 CollaborationUse 拥有的依赖项实现的。协作中的每个协作协作都受不同的依赖关系约束，并且是其供应商。Dependency 的客户端是一个 ConnectableElement，它以某种方式与上下文分类器相关联：它可能是时间分类器的直接协作上下文，或者是通过来自上下文分类器的某些引用集可到达的元素。这些角色绑定指示时间分类器中哪个上下文在协作元素中扮演哪个协作协作

## 代表



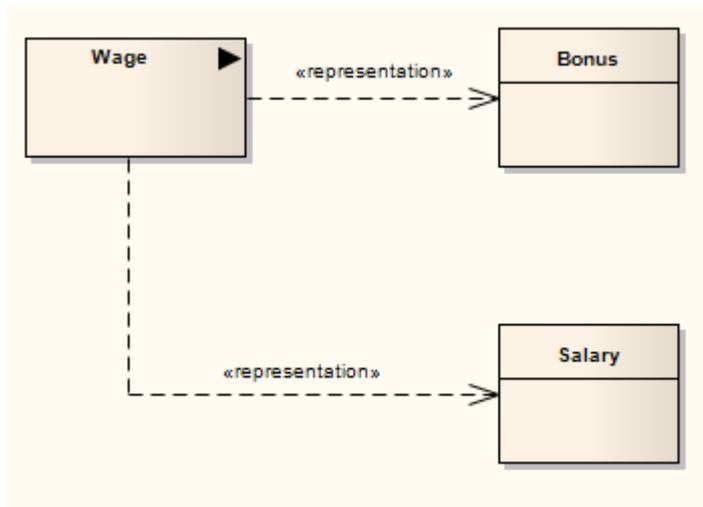
## 描述

代表连接器表示在协作分类器中使用了 a。通常在复合结构图中使用。连接器是从协作绘制到其拥有的分类器。

## 工具箱icon



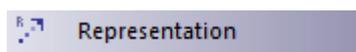
## 表示



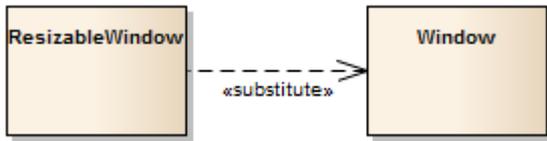
## 描述

表示关系是依赖关系的一种特殊化，它连接信息项元素，这些元素在模型中表示相同的想法，通常在分析图中。例如，“Bonus”和“Salary”都是信息项“Wage”的表示。

## 工具箱icon



## 替代



## 描述

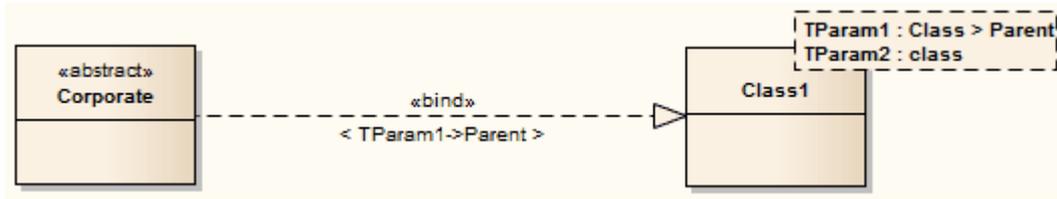
Substitution 是两个 Classifier 之间的关系，表示替换的分类器符合合约分类器指定的合约。这意味着替换分类器的实例是运行时可替换的，其中合约分类器的实例是预期的。在该示例中，名为 **ResizableWindow** 的类有一个到名为 **Window** 的类的替换连接器，这意味着无论您在哪里被要求提供一个窗口，您都可以使用一个可调整大小的窗口。

替换关系是依赖关系的子类型。

## 工具箱icon



# 模板捆绑



## 描述

在绑定类和参数化类之间创建一个模板捆绑连接器。然后，您在该连接器上定义绑定表达式。但是，如果绑定类需要一个概括或关联关系与参数化类，则可以在该关系上定义绑定表达式。

您可以使用以下方法创建模板捆绑连接器：

- 工具箱类关系页面的图表捆绑工具箱
- 源类元素旁边的快速链接器箭头
- 绑定类元素的“模板”对话框；在这里，您可以通过单击“捆绑”面板下的“添加”按钮，指定连接器类型，然后从“选择 <Item>”对话框中选择目标参数化类来创建模板捆绑关系

这些方法中的每一个都创建连接器本身。对于前两种方法，然后单击连接器使其成为属性窗口的聚焦，在该窗口上选择“捆绑”选项卡以将参数替换定义为绑定表达式。第三种方法自动将您带到“属性”对话框上的同一选项卡。

## 工具箱icon



## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.24) 指出：

模板是A参数化的元素...用于使用模板关系生成其他模型元素。模板签名的模板参数指定将由绑定中的实际参数（或默认参数）替换的形式参数。

# 参数替换

一旦存在模板捆绑（或其他绑定）关系，您可以添加参数替换以在绑定表达式中标识被替换的形式参数以及替换它们的实际参数。

## 访问

使用本表中列出的任何方法，显示连接器的“属性”对话框或属性窗口的“捆绑”页面。

功能区	开始>应用>设计>属性>点击连接器>捆绑(属性窗口) 设计>元素>编辑>属性>点击连接器>捆绑(属性窗口)
上下文菜单	上图  右击连接器 属性>捆绑('属性'对话框)
键盘快捷键	Ctrl+2 > 点击连接器 >捆绑(属性窗口)
其它	上图  双击连接器>捆绑 ('属性'对话框)

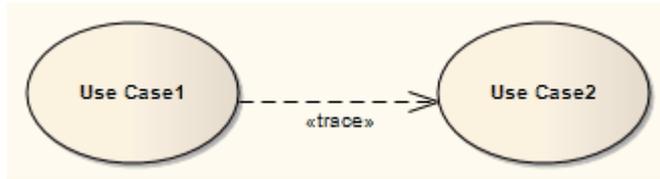
## 定义参数替换

目标“字段标识目标参数化类。

节	行动
1	单击“参数替换”面板下方的添加按钮。 面板中的下一个可用行已启用编辑，并且单词“<none>”显示在“正在”列中。
2	单击该字段，然后单击现在显示的下拉箭头。 目标A显示的模板参数类；单击所需的参数。
3	单击参数对应的“实际”字段中的  按钮。 如果模板参数： <ul style="list-style-type: none"> <li>没有约束，一个简短的时间上下文菜单显示提供选择在“实际”字段中键入自由文本值，或从“选择分类器”对话框中选择一个分类器</li> <li>定义了一个约束，“选择分类器”对话框会自动显示，显示可用的分类器</li> </ul>
4	找到并选择所需的分类器以替换绑定表达式中的参数。 如果您未定义实际分类器并且模板参数定义了默认值，则表达式中将使用该默认值。
5	要编辑现有的参数替换，请单击它们并按照步骤 3 和 4 中的说明进行所需的更改。
6	单击应用按钮和/或确定按钮。 参数替换显示为连接器下方的标签。



## 跟踪



### 描述

跟踪关系是抽象的特化，连接模型元素或跨模型表示相同概念的元素集。跟踪通常用于跟踪需求和模型的变化，通常在可追溯性图中，或在类、用例、物件或复合结构图中。

由于两个方向都可能发生变化，因此通常忽略此跟踪的顺序。关系的属性可以指定跟踪映射，但跟踪通常是双向的、非正式的并且很少可计算。

### 工具箱icon

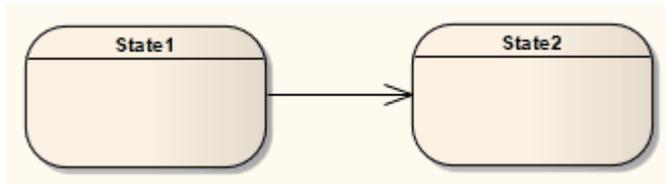


### OMG UML规范：

OMG Unified Modeling Language规范 (v2.5. 1 , p.682) 指出：

指定在不同模型中表示相同概念的模型元素或模型元素集之间的跟踪关系。跟踪主要用于跨模型跟踪需求和更改。由于模型的变化可能发生在两个方向上，因此通常可以忽略依赖的方向性。映射指定了两者之间的关系，但它很少可计算并且通常是非正式的。

# 转移



## 描述

如果需要在状态机图中定义从一个状态到另一个状态的逻辑移动，可以拖动一个转移从工具箱连接到图表上。你控制转移通过连接器 属性”对话框。

字段	行动
守卫条件	事件被调度之后，对应的转移之前要计算的表达式中的类型转移被触发。 如果当时守卫是真的，转移已启用；否则，它被禁用。
影响是一种行为	将 影响”字段从自由文本字段转换为特定活动或行为的定义。 将显示 选择 <项目>”对话框，提示您从模型中选择活动或行为元素。
影响	任何一个： <ul style="list-style-type: none"> <li>• 类型转移影响的描述转移，或者</li> <li>• 如果您选择了 影响是一种行为”复选框，请选择在转移期间要执行的活动或行为转移 (若要随后更改此设置，请单击  按钮以重新显示 选择 &lt;Item&gt;”对话框)</li> </ul>
触发器名称	指定触发器的名称；任何一个： <ul style="list-style-type: none"> <li>• 类型名称，或</li> <li>• 从 Select &lt;Item&gt; 对话框中选择模型中的现有触发器，通过单击  按钮显示该对话框</li> </ul>
触发器类型	指定触发器类型： <ul style="list-style-type: none"> <li>• 调用-指定事件为调用，通过调用操作向关联object发送消息</li> <li>• 更改-指定事件是一个 ChangeEvent，这表明转换是属性值更改的结果</li> <li>• 信号指定事件为SignalEvent，对应接收到一个异步信号实例</li> <li>• 时间-对应于 TimeEvent；它指定了一个时刻</li> </ul> 状态机的代码生成需要四种类型中的任何一种的规范值。

规格	指定引发转移的事件转移 ;任何一个： <ul style="list-style-type: none"> <li>• 类型事件（时间或变化），或</li> <li>• 使用 选择&lt;项目&gt;“对话框在模型中选择现有规范，通过单击  按钮显示该对话框</li> </ul>
新的	清除准备开始定义新触发器的字段。
节省	保存新创建或编辑的触发器。
删除	从列表中删除选定的触发器。
<触发器列表>	列出现有的触发器，它们可能有也可能没有名称和类型，并且可以包括在旧模型中创建的触发器。

### 注记

- 分叉和汇合段既不能有触发器也不能有守卫
- 您可以使用转移上的“触发器”选项识别隐藏的触发器并在浏览器窗口中定位触发器转移连接器上下文菜单；如果转移存在一个触发器转移它会立即在浏览器窗口中突出显示，如果存在多个触发器，则会显示“无素用途”对话框 - 选择所需的触发器并单击“打开”按钮以突出显示浏览器窗口中的触发器
- 您可以将自转移定义为内部转移，并表示状态元素的一个隔间中的连接器及其属性

### 工具箱icon



### OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.359) 指出：

#### A转移

表示恰好一个源顶点和正好一个目标顶点之间的弧（源和目标可能是同一个顶点）。它可以形成复合转换的一部分，将状态机从一个稳定状态配置带到另一个，表示状态机对触发它的事件的发生的完整响应。

## 内部转移

如果需要定义内部转移

在一个状态下，您可以通过创建外部自转换连接器（其中源和目标是相同的状态）然后更改连接器的“种类”属性来实现。然后将自转换连接器从图中移除，内部转移

显示在状态元素里面的一个隔间里。

### 定义内部转移

节	行动
1	在浏览器窗口中，双击包含状态元素的状态机图将其打开。
2	在状态元素上，创建一个转移连接器从元素发出和终止于元素（一个'自我转移'）。 在图表工具箱中，选择转移连接器，然后单击并释放状态元素。
3	右键单击连接器并选择“属性”选项以显示“属性”对话框。
4	选择“约束”选项卡并为转移定义任何守卫、效果和触发器转移。
5	选择“常规”选项卡，然后选择子选项卡“高级”。单击类型属性值字段中的下拉箭头并选择“内部”。
6	点击确定按钮。转移与内部活动（ <code>exit/</code> 、 <code>do/</code> 、 <code>entry/</code> ）显示在同一隔间中。  <pre> State1 entry / Init exit / finalize TIME_TICK [i&lt;0] /   this.xOffset +=2; TIME_TICK [i&gt;0] /   this.xOffset += 5; </pre>

### 注记

- 查看或编辑内部转移的属性转移

, 双击状态的条目

- 如果您需要多个内部转换，包括那些具有触发器但不同保护的转换，您可以使用每个转移单独创建它们转移拥有自己的警卫
- 您可以通过单击状态元素、在“内部触发器”选项卡上显示特征窗口、右键单击选项卡并选择“新内部触发器”选项来创建进一步的转换和内部触发器

## OMG UML规范：

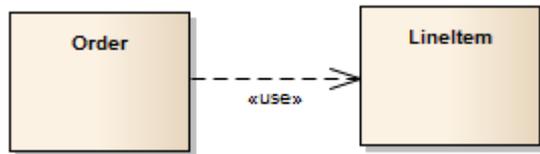
OMG UML规范 ( UML Superstructure Specification, v2.4. 1 , p.362 ) 指出：

[ A TransitionKind of internal ] 暗示着转移

· 如果触发，则在不退出或进入源状态的情况下发生（即，它不会导致状态变化）。这意味着不会调用源状态的进入或退出条件。内部转移

即使区域位于一个或多个嵌套在关联状态中的区域中，也可以采用

## 用途



## 描述

“用途”是一个类图关系，其中一个元素需要另一个元素才能完全实现或操作。示例图显示了类的完整实现需要类。

“用途”关系是“依赖”关系的子类型。

## 工具箱icon

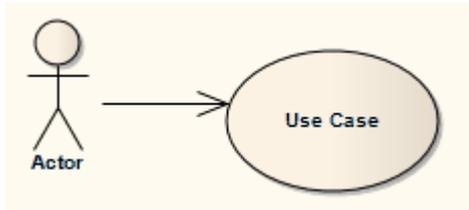


## OMG UML规范：

OMG Unified Modeling Language规范 (v2.5.1, p.38) 指出：

用途是一种依赖关系，其中A NamedElement 需要另一个 NamedElement ( 或一组 NamedElements ) 才能完全实现或操作。用途不指定客户如何使用供应商，除了供应商被客户的定义或实现使用这一事实。

## 使用



### 描述

A使用关系一个元素需要另一个元素来执行某些交互。使用关系没有指定如何使用目标供应商，除了源客户端在定义或实现中使用它。

您通常使用示例图中的使用关系来模型用例如何使用系统功能（使用案例）。

### 注记

- 在参与者和用例之间建立关联更为常见（也是正确的UML）
- 用于类图中的用途关系是一种不同的关系

### 工具箱icon



# UML构造型

UML支持构造型，这是一种用于逻辑扩展或更改基本UML模型元素的含义、显示、特征或语法的内置机制。您可以将构造型应用于一系列模型元素类型，包括：

- 元素（例如类和对象）
- 关系（例如依赖关系和关联）
- 关联结束
- 属性和操作
- 操作参数

不同的模型元素具有与之相关的不同刻板印象。您可以通过三种不同的方式创建和使用自己的刻板印象：

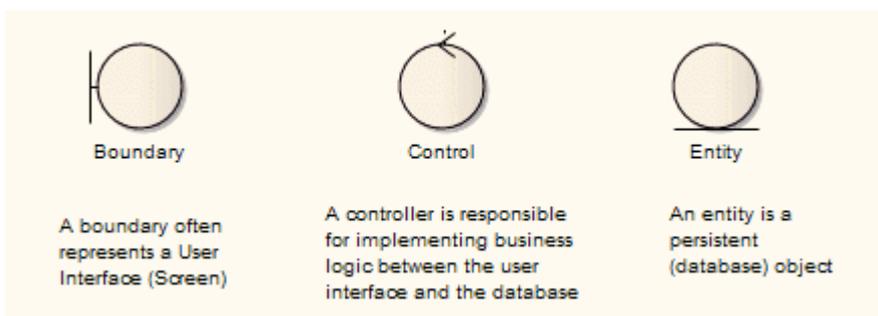
- 根据基本的UML模型元素类型创建新的object类型，作为配置文件的一部分导入到任何模型中，并通过图表工具箱；Enterprise Architect中已经提供的扩展元素类型的示例包括库表元素（它是一个原型类元素）和边界、控件和实体元素（它是一个物件元素）
- 自定义特定类型的模型元素实例的外观或属性；这些构造型仅通过object的“属性”对话框在创建它们的模型中应用，尽管您可以在模型之间传输自定义构造型定义作为参考
- 作为元素上的一个简单标签，用于标识该元素所代表的object的角色或性质

有关构造型的进一步定义，请参阅OMG UML规范（UML上层结构规范，v2.1.1，第18.3.8节，第667-672页）。

在构造型不影响外观的情况下，它通常由基本UML object形状上的名称指示。在这个例子中，«myStereotype2»是原型名称。一些内置的刻板印象也用图标表示；见构造型可见性。



在构造型导致元素以不同方式绘制或用于定义新类型的object的情况下，元素形状可以完全不同，如三个稳健性图构造型所示：



您可以通过将构造型与元文件（图像文件）和填充、边框和文本颜色或定义object的形状、尺寸和文本的形状形状脚本相关联来应用新的外观或形状。

# 应用构造型

在建模过程中，您可能会决定现有object需要构造型。Enterprise Architect允许将新的原型应用于对象本身或与其他原型结合。您可以通过以下位置的“构造型”字段执行此操作：

- object的“属性”对话框或
- 属性窗口

## 访问

使用此处概述的方法之一打开属性窗口或“属性”对话框，然后单击“构造型”字段右侧的按钮，并使用“<object名称>的构造型”对话框（构造型选择器）。

功能区	探索>门户>窗口>属性>属性 >开始>所有窗口>属性>常规>属性 设计>包>管理>属性>通用
上下文菜单	右键单击包或元素 属性>常规
键盘快捷键	Alt+Enter（“属性”对话框） Shift+Enter（“属性”对话框） Ctrl+2（属性窗口）

# 构造型选择器

如果您想从多个来源（例如Profiles或定制的构造型列表）将多个构造型应用于UML object，您可以从 构造型 for < object名称>”对话框中选择构造型。此对话框还可以帮助您识别现有的、有效的个人刻板印象，并创建新的刻板印象。在这一点上，新的刻板印象是简单的标签；如果您希望它们对object施加影响，请在“UML类型”对话框的 构造型”选项卡上找到它们并定义影响。

## 访问

其它	通过单击object 属性”对话框或属性窗口中 构造型”  旁边的浏览图标，显示 < object名称>的构造构造型”对话框。
----	--

## 选择构造型以应用或删除

字段/按钮	行动
蓝图	<p>单击下拉箭头并选择一个蓝图名称，以将可供选择的构造型限制为该蓝图下可用的构造型。</p> <p>如果您想从任何蓝图中检查模型中的原型，请单击按钮并选择 全部”。</p> <p>构造型”列列出了可用的构造型；单击每个原型的复选框以进行选择。</p> <p>当您最初打开 &lt; object名称&gt;的构造型”对话框时，字段中的蓝图名称是应用程序屏幕右上角的  选项&lt;透视名称&gt;图标中显示的蓝图。在 构造型for &lt; object名称&gt;”对话框中更改蓝图不会更改图标上的 全局”蓝图。如果您想将 构造型for &lt; object名称&gt;”对话框蓝图设置为全局蓝图，只需单击下拉箭头并选择 活动”。</p>
配置文件	<p>单击下拉箭头并选择所需的构造型源- 例如，集成的MDG 技术或基本 EAUML，或为您的自定义构造型列表选择空白行。</p> <p>该字段默认为最后选择的配置文件（如果它在当前设置的蓝图），或者，如果元素已经具有原型，则为该原型的配置文件。</p>
构造型	<p>选中每个所需构造型对应的复选框。</p> <p>如果您不再想使用构造型，请取消选中该复选框。</p>
适用于	显示选定构造型分配给的object类型。
新的	单击此按钮可创建一个新的（但未定义的）构造型。为原型名称显示A提示。
确定	单击此按钮以应用选择。
取消	单击此按钮可取消任何选择并关闭对话框。

## 注记

- 如果您选择了多个构造型，属性对话框会在构造型字段的单独行中列出它们
- 图表中object的构造型外观受图表属性对话框中构造型可见性设置的影响

# 构造型能见度

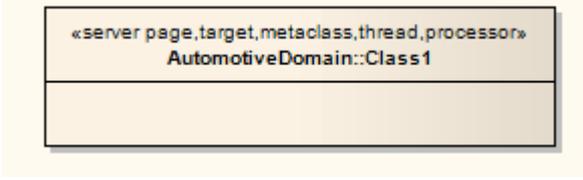
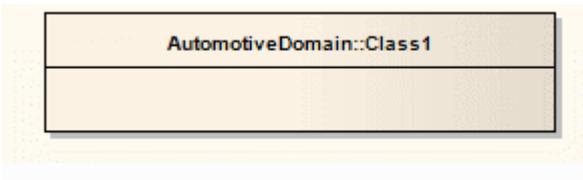
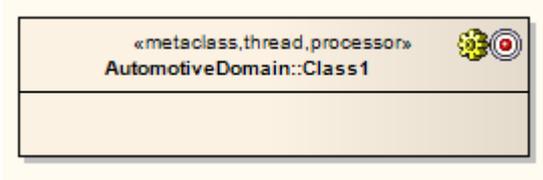
当您将一个或多个构造型应用于object时，该object在图表中的显示默认为在 guillemets (« ») 内的string中显示构造型名称；多个名称用逗号分隔。一些刻板印象与显示在元素右上角的小图标相关联；这些图标是系统内置的，不能删除或添加。在这两种情况下，您都可以使用图表的“属性”对话框来修改图表中文本或图标原型指示器的可见性。

## 访问

显示图表的“属性”对话框，然后显示“元素”选项卡或“特征”选项卡，以分别选择元素或特征的刻板印象的可见性。

功能区	设计>图表>管理>属性>选择“元素”或“特征”选项卡
上下文菜单	右键单击图表背景 属性>选择“元素”或“特征”选项卡
其它	双击图表背景 > 选择“元素”或“特征”选项卡

## 设置构造型可见性选项

字段/按钮	行动
显示元素构造型	<p>选中“元素”选项卡上的此复选框以显示当前图表中的所有元素刻板印象和关键字；例如（未选择“使用构造型”图标）：</p>  <p>取消选中此复选框可隐藏所有元素原型名称、图标和关键字。</p> 
使用构造型图标	<p>选中“元素”选项卡上的此复选框以显示图标而不是文本，对于那些定义了图标的元素原型。</p> <p>没有关联图标的构造型仍然由构造型名称表示；例如。</p>  <p>图标代表原型«服务器页面»和«目标»。</p>

显示构造型	选中“特征”选项卡上的此复选框以显示当前图表中的所有属性和操作构造型。此选项不影响元素原型的显示。
-------	---

## 注记

- 在浏览器窗口中，object名称前面是 guillemets 中的构造型名称，多个名称由第一个构造型名称后跟省略号 (...) 表示；您可以通过取消选择浏览器窗口“显示构造型”复选框来隐藏构造型名称（开始>外观>首选项>首选项>常规）

# 标准构造型

此表标识了 EABase 基础模型中提供的标准构造型，每个都用 guillemets («») 括起来。

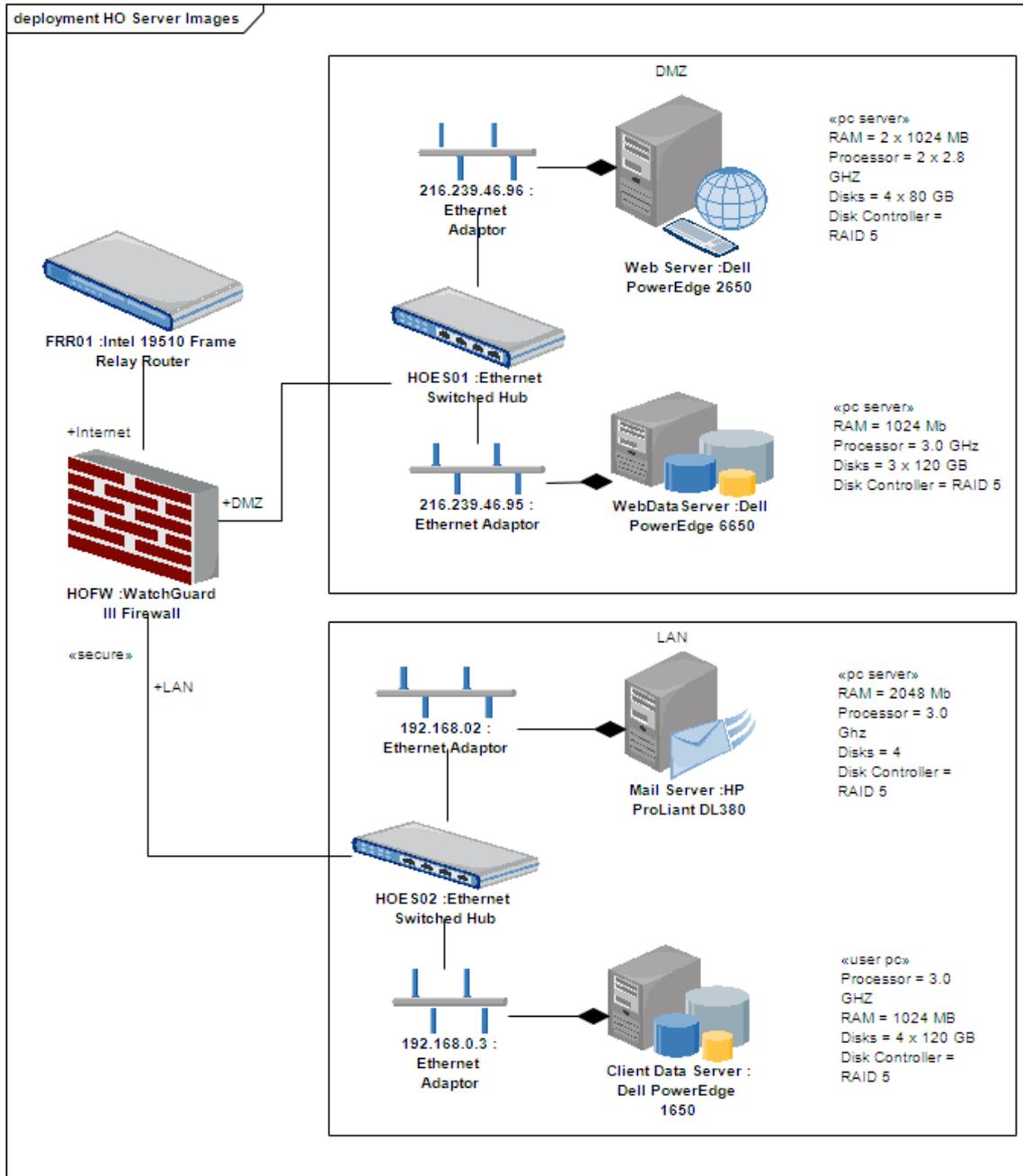
## 构造型

构造型	基类
使用权"	依赖
变得"	流动
称呼"	用途
复制"	流动
创造"	信息
派生"	抽象
破坏"	信息
文档"	抽象
«可执行»	抽象
正面"	包
文件"	抽象
框架"	包
«朋友»	依赖
全球的"	关联结束
执行"	类
执行"	概括
进口"	依赖
«实例化»	用途
«不变的»	约束
图书馆"	抽象
当地的"	关联结束

«元类»	类
范围"	关联结束
«后置条件»	约束
«电源类型»	类
前提"	约束
过程"	分类器
«精炼»	抽象
要求"	注解
责任"	注解
自己"	关联结束
发送"	用途
存根"	包
桌子"	抽象
线"	分类器
“痕迹”	抽象
类型"	类
效用"	分类器

# 带有选择的构造型图片

如果您想使用图像来表示元素（例如，使用 3-D 框来描绘硬件组件，甚至使用单元本身的图像），您可以使用与元文件相关联的构造型来实现。当构造型应用于支持替代图形格式的类或其他元素时，使用图像而不是标准 UML 形状来绘制元素。例如，在这个部署图中，部件元素都有替代图像。



## 笔记

- 您无法更改包含生命线的元素的表示，例如序列图中的元素；标准表示在这些元素的使用和函数中很重要

# 自定义构造型

A定义构造型将不同的外观或特性应用于基本UML模型组件或特征。您可以通过两种不同的方式应用自定义构造型：

- 更改特定类型的模型组件实例的外观或属性；这些构造型在“UML类型”对话框的“构造型”选项卡上定义，并通过object的“属性”对话框在创建它们的模型中应用，尽管您可以在模型之间传输自定义构造型定义，如参考数据
- 作为元素上的一个简单标签，用于识别元素所代表的object的角色或性质；这些构造型只是键入到object“属性”对话框的“构造型”字段中的名称，并且不会影响元素显示，除非随后对其进行编辑以产生效果

您可以对object的形状、尺寸和外观进行更明显的更改，您可以通过将元文件（图像文件）和自定义颜色与原型相关联，或将形状脚本附加到原型来应用这些更改。当您定义并保存了构造型后，您可以将它应用到与之关联的基类的任何新的或现有的object上。

## 访问

功能区	设置 > 参考 > UML类型 > 构造型
-----	-----------------------

## 保持自定义的刻板印象

选项	行动
构造型	类型或选择构造型的名称。
团队名字	(可选) 类型复数名称，用于对属性和操作的特征型进行分组；该名称将显示在属性和操作隔间中的图表上。
基础类	单击下拉箭头并选择预先存在的object类型的名称，以便原型元素将继承该类型的基本特征。
注记	(可选，但推荐) 类型任何关于构造型的注记(不是要应用构造型的元素)。
新的	单击此按钮可清除字段以创建新的构造型定义。
节省	单击此按钮以保存新的或编辑的构造型定义。
删除	单击此按钮可从模型中删除构造型定义。
覆盖外观	
没有任何	选择保留此构造型的默认元素外观。
元文件	选择将构造型与图像元文件(.cmf或.wmf)相关联，以在使用构造型时应用该图像。
形状脚本	选择以将构造型与使用形状脚本语言创建的自定义形状相关联。

分配	<p>单击此按钮可以：</p> <ul style="list-style-type: none"> <li>• 显示浏览器以找到与原型关联的 .emf 或 .wmf 元文件，或</li> <li>• 打开形状编辑器或创建与原型关联的形状脚本</li> </ul>
编辑	<p>如果一个形状脚本已经与原型关联，点击这个按钮打开形状编辑器来更新形状脚本。</p>
消除	<p>从构造型中移除关联的元文件或形状脚本。</p>
默认颜色	
充满	<p>单击下拉箭头并选择或定义要由构造型细化的元素的默认背景颜色。</p> <p>此颜色将应用于所有已应用该构造型的任何元素的出现；如果颜色随后更改，则更改立即应用于应用了构造型的任何元素的所有出现（对于构造型的任何其他属性的更改）。</p> <p>但是，在使用构造型创建的元素上，默认颜色可能会被已应用于元素的更高优先级的其他颜色定义覆盖。</p>
边界	<p>单击下拉箭头并选择或定义要由构造型细化的元素边框的默认颜色。</p>
字体	<p>单击下拉箭头并选择或定义要由构造型细化的元素的文本的默认颜色。</p>
重置	<p>将默认颜色重置为与构造型关联的基本元素的颜色。</p>

## 注记

- 您可以使用 设置>模型>传输>导出参考”和 导入参考”功能区选项在模型之间传输自定义构造型定义
- 您还可以创建扩展基本UML模型元素类型的构造型元素，以创建新的模型元素类型；您可以在其他项目中重新使用这些扩展模型元素，方法是将它们合并到配置文件中（通常在MDG技术中）并将其导入到各种目标项目中

# 扩展UML

有时，使用基本UML模型元素无法充分表达建模问题，或者类似地，工作领域属于需要定制建模方法或程序语言支持的专业领域。为了满足这些要求，您可以扩展UML的功能以开发新的建模构造，使用MDG 技术来组合和部署广泛的或扩展机制，例如：

- UML Profiles
- 构造型
- 形状脚本
- 标记值
- 约束
- 模式
- 自定义代码和变换模板，以及
- 语法

使用MDG 技术创建向导，您可以快速轻松地将扩展集成到技术中，并快速定制UML和Enterprise Architect以解决原始UML规范中未明确定义的特定建模域，但使用仍然是一部分的扩展机制规格。

## 功能

功能	描述
<p>扩展UML</p> 	使用MDG 技术向导快速轻松地将UML扩展到配置文件和技术。
<p>使用MDG 技术</p> 	包装您的UML Profiles、代码模块、脚本、模式、图像、标记值类型、报告模板、链接文档模板和工具箱页面。
<p>MDG 技术SDK</p> 	构建自己的技术所需的一切，例如形状脚本、标记值类型、代码模板框架、语法框架等等。

## 使用UML Profiles

UML概要文件是A轻量级的扩展机制，它是UML标准的一部分。使用配置文件，您可以创建一组适用于对特定领域、平台或方法进行建模的模型构造。Enterprise Architect为创建和部署配置文件提供了一种灵活且直观的机制。标准UML构造增加了原型和标记值，以创建适合建模目的的新定制元素。配置文件只是这些结构及其原型和相关标记值A集合。构造型可以应用于元素、特征、连接器和连接器末端。A模型驱动生成(MDG)技术分发和实施配置文件。

部署的技术会自动为该技术中的每个UML配置文件在图表工具箱中生成一个元素和关系页面。当您将元素和连接器从工具箱拖到当前图表上时，构造型、标记值和默认值、注记和元文件(如果已指定)将自动应用于新元素。您还可以将配置文件属性和操作拖放到现有的类上，以便立即添加指定的原型和标记值。

## 将配置文件对象添加到一个图表

将技术导入您的项目后，可以从工具箱图表技术页面中获得分析对象（元素和连接器）和特征（属性和操作）。将配置文件对象添加到图表的方式与在系统上使用标准UML对象的方式没有什么不同。

### 访问

功能区	设计 >  工具箱图表查找工具箱项”对话框并指定<技术名称>
键盘快捷键	Ctrl+Shift+3 :  显示 查找工具箱项目”对话框并指定<技术名称>

### 使用配置文件对象

行动	描述
将基于配置文件的元素添加到图表	单击工具箱页面中的元素并将其拖到图表上。
将基于配置文件的连接器添加到图表	点击工具箱页面中的连接器，然后点击图中的源元素，将其拖到目标上。
将基于配置文件的属性或操作添加到图表	点击工具箱页面中的属性或操作，将其拖到图上的宿主元素上。 系统提示您输入特征的名称。

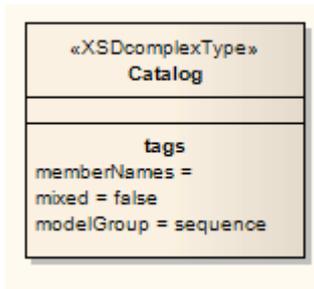
## Profiles中的标记值

构造型元素或连接器内的构造型可以定义一个或多个相关的标记值。当您从图表工具箱一个已配置的元素或连接器图表图表上时，任何相关的标记值都会自动添加到新的元素或连接器中。配置对象中的标记值是进一步扩展UML建模的多功能性的极好方法。

例如，用于 XSD ( XML Schema ) 的UML配置文件提供了 XSDComplexType 构造型来扩展类；这种刻板印象有标记值：

- 成员姓名
- 混合和
- 模型组

创建复杂类型元素时，标记值会被添加并在元素的标签栏中可见（包括那些没有设置值的元素）。



当您选择元素时，属性窗口会显示所有关联的标签。

- 配置文件中导入的标记值覆盖“UML类型”对话框中等效标记的值；如果未设置配置文件中标记的初始值，则即使“UML类型”对话框中有标记的默认值，元素中显示的标记的值也会为空
- 自动设置具有默认配置文件值的标签
- 如果标记的元素中的标记值有一个值部分（例如，values="元素| attribute | both" default="both"），您可以从下拉列表中选择非默认值
- 在不存在值的情况下，您可以将值添加为自由文本；您可以对没有初始值的配置文件标签执行此操作，以使用“UML类型”对话框中的默认值

## 同步标记值和约束

当您从配置object创建元素、属性、操作或连接器时，标记值和约束会从配置文件原型中添加。随后，您可能会更新配置文件中特定定型的约束或标记值，在这种情况下，模型中已经创建的项目将没有那些额外的约束或标记值标签和注记。

类似地，您可能已经手动将构造型添加到一组对象，这会添加标记值但不添加与该构造型关联的约束，并且现在希望对象接收约束。

您可以使用 Synchronize构造型函数应用更新或缺失的标记值类型和约束。这适用于您的模型中的任何配置元素，来自与Enterprise Architect集成或导入的任何技术。

### 访问

功能区	设计>  工具箱图表查找工具箱项”对话框并指定<技术名称> 已配置元素/连接器/特征的右键单击图标 同步构造型
键盘快捷键	Ctrl+Shift+3：在 查找工具栏项”对话框中指定<技术名称>  已配置元素/连接器/特征的右键单击图标 同步构造型

### 使用技术工具箱页面同步对象

节	行动
1	在 同步配置元素”对话框中，单击确定按钮。 使用选定的轮廓object图标创建的所有元素、特征或连接器都会在整个模型中更新。 已修改的项目，以及所做的更改，都列在 行动”字段中。
2	更新完成后，单击 取消”按钮。

### 选择-单物件更新

您可以快速同步图表中单个元素的标签和约束。去做这个：

节	行动
1	将更新后的分析元素从图表工具箱页面拖到图表中的元素上。 显示A简短的上下文菜单。
2	选择 应用«stereotype name»”菜单选项。 图表元素使用来自自己分析元素的任何标签和约束进行更新，但它还没有。

## 注记

- 当图表工具箱表示配置的元素或连接器时，将显示“构造型”上下文菜单选项；它不会显示基本UML object图标
- 您可以通过显示元素“属性”对话框、打开“标签”选项卡并单击适当的配置文件来审阅任何元素
- 从object中删除原型会自动删除由该原型分配的任何标记值

# 扩展构造型

Enterprise Architect支持大量定义了元素和连接器集的建模语言和平台。然而，Enterprise Architect为建模者提供了创建大量其他元素的能力，通常是通过向现有元素添加构造型。作为建模者，您可以使用原型和形状脚本等功能自由地创建自己的新元素。用户社区为特定领域创建和共享一组共同的刻板印象是很常见的。

- [Analysis Stereotypes](#)
- [Boundary](#)
- [Composite Elements](#)
- [Control](#)
- [Entity](#)
- [Event](#)
- [Feature](#)
- [Hyperlink](#)
- [Image](#)
- [N-Ary Association](#)
- [Packaging Component](#)
- [Process](#)
- [Requirements](#)
- [Risk](#)
- [Screen](#)
- [Task](#)
- [Test Case](#)
- [Database Tables](#)
- [UI Control Elements](#)
- [Web Stereotypes](#)

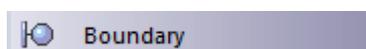
# 边界



A 是用于对某些系统边界（通常边界用户界面屏幕）进行建模的构造型物件。您还可以将边界创建为构造型类。边界元素用于分析以捕获用户交互、屏幕流和元素交互（或“协作”）。

A 在概念相中用于捕获用户在屏幕级别（或其他边界界面类型）与系统的交互。它通常用于序列和稳健性边界分析）图中。它是模型视图控制器模式中的视图。

## 工具箱icon



## 创建边界

有两种方法可以在图表上创建边界。

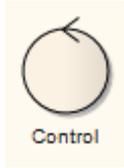
### 创建一个元素作为一个边界印象的类

节	行动
1	插入一个新类。
2	右键单击元素并选择“属性”选项；显示“属性”对话框。
3	在“构造型”字段中，键入值“边界”。
4	单击应用按钮和确定按钮。
5	按 Ctrl+S 保存图表。

### 创建一个边界作为物件元素

节	行动
1	在工具箱中，单击图表以显示“查找  工具箱”对话框并指定“分析”。
2	从“分析元素”页面中，将“边界”图标拖到图表上。

# 控件



A是物件控件A组织和分析控件包括鲁棒性 )、序列和通讯图中。它是模型-视图-控制器模式的控制器。您还可以创建一个控件作为构造型类。

## 工具箱icon



## 创建一个元素控件

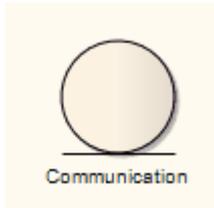
### 在元素上创建一个控件作为物件

节	行动
1	在工具箱中，单击图表以显示 查找  工具箱“对话框并指定 分析”。
2	从分析元素页面中，将控件拖到图表上。

### 创建一个控件作为一个元素类

节	行动
1	插入一个新类。
2	右键单击元素并选择 属性”选项；显示 属性”对话框。
3	在 构造型”字段中，键入值 控制”。
4	单击应用和确定按钮。
5	按 Ctrl+S 保存图表。

# 实体



实体是一种刻板的物件，它模拟了系统中捕获信息或知识的存储或持久性机制。它是模型-视图-控制器模式中的模型。

您还可以创建实体作为构造型类。请参阅创建实体主题。

## 工具箱icon



## 创建一个实体

### 在图上创建实体元素作为物件

节	行动
1	在工具箱中，单击图表以显示 查找  工具箱“对话框并指定 分析”。
2	从分析元素页面中，将实体图标拖到图表上。

### 创建实体元素作为原型类

节	行动
1	插入一个新类。
2	右键单击元素并选择 属性”选项；显示 属性”对话框。
3	在 构造型”字段中，键入值 实体”。
4	单击应用和确定按钮。
5	按 Ctrl+S 保存图表。

# 超链接

您可以超链接元素放置到图表上。该元素是一种文本元素，但可以包含指向一系列对象的指针，例如相关文档文件、网页、帮助、模型特征甚至其他Enterprise Architect模型文件。当您双击元素时，Enterprise Architect会执行该链接。

要添加超链接元素，可以：

- 将“超链接”图标从图表工具箱的“公共”页面拖到图表上，或者
- 单击UML元素工具栏中的“超链接”图标，然后单击图表



## 配置超链接

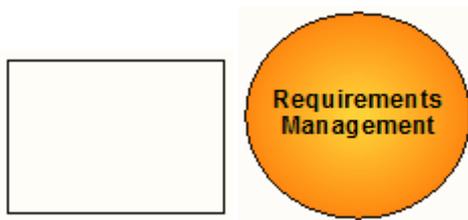
超链接添加到图表时，立即键入一些链接文本，单击关闭元素，然后双击该元素。将显示“超链接细节”对话框。如果您想以更易读的布局显示信息，您可以调整对话框的大小。

字段/按钮	行动
类型	单击下拉箭头并选择要链接到的object类型。 在许多情况下，当您选择类型时，会为该类型的object显示一个浏览器对话框，您可以从中选择要链接到的实际object。
行动	当对话框首次显示时，类型“默认为“文件”，或者如果您从“类型”下拉列表中选择“文件”，则启用此字段。 该字段默认值为“打开”，以只读模式显示文件内容。如果您希望用户能够更改文件内容，请单击下拉箭头并选择值“编辑”。 系统会自动选择合适的编辑器。例如，如果您超链接到.rtf文件，您可以在任何合适的内部查看器中查看该文件；但是，您不能在Enterprise Architect中编辑.rtf文件，因此该文件总是在窗口默认的.rtf编辑器中打开。
别名	此字段显示您在图表上创建元素时输入的文本作为链接文本。如果要更改此文本，请用新文本改写它。 如果您不提供别名，则文本默认为链接本身，或者（对于某些链接目标，例如矩阵配置文件）系统会生成简单的文本指令。
隐藏图标	如果您希望仅显示链接文本而不显示  图标，请选中此复选框。
注记	类型在您可能需要解释超链接的任何注记中。这些注记没有显示在图上的元素中。您可以使用注记工具栏格式化注记。
地址	如果浏览器在输入到“类型”字段时显示，则当您选择要链接到object名称或位置的object时，此字段中会显示。（如果object不是通过路径或“地址”访问的，则通常不标记该字段。） 如果没有显示浏览器，或者您想将链接object更改为另一个相同类型的对象，请输入object位置或单击  按钮以显示相应的浏览器，然后选择目标object。

## 注记

- 如果需要，您可以创建多个空超链接以完成；如果您然后双击一个空的超链接，将显示“超链接细节”对话框，您可以输入详细信息
- 将光标悬停在超链接上时，标准超链接元素器按钮也将显示在弹出工具栏中，允许您显示“属性”对话框，在图表中查找元素，在浏览器窗口中查找元素，或打开附加到元素的链接文档

## 图像



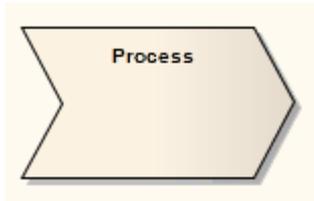
图像是一个边界系统元素，它首先自动显示“边界 属性”对话框，然后显示“选择备用图像”对话框以将其表示更改为导入的图像。您可以将其用作元素或元素组的图标，或用作图表背景。

图像元素可从工具箱的“公共”页面获得。

## 工具箱icon



# 进程



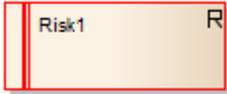
A是具有构造型流程的活动元素，它表达了业务流程的概念。通常，它涉及输入、输出、工作进程、目标以及与其他流程的连接。进程元素通常用于分析图表中。

业务流程通常涉及组织的许多部分并跨越一个或多个系统。

## 工具箱icon



# 风险



风险元素不同于您通过“风险”窗口分配给元素的风险。此类风险是所选元素的内部风险，而风险元素可与多个元素相关联，这些元素既可以位于逻辑组中，也可以完全独立。

风险要素可从工具箱的“需求”页面获取。

## 使用风险元素

A 定义为不确定性对目标的影响。在项目管理中，需要尝试识别风险风险评估：

- 它们对项目产生负面影响的可能性以及
- 这种影响可能有多大

那些发生概率高和/或对项目影响大的风险可以得到减轻。

风险管理A可能包括以下五个步骤：

1. 识别风险并用风险元素来表示每个风险。
2. 确定哪些元素（如组件、使用案例或特征）容易受到每个风险的影响；您可能决定从这些元素到风险元素创建“跟踪”依赖关系。
3. 评估风险的可能性和大小。
4. 找到减轻风险的方法。
5. 根据风险降低措施的可能性、严重程度以及缓解的难易程度，确定其优先次序。

## 注记

- 风险元素可以在元素的右上角显示或不显示标识“R”；要切换此字母的显示，请在“首选项”对话框的“对象”页面上选择或取消选择“显示要求的原型图标”复选框

## 工具箱icon



# 任务

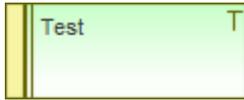


A元素表示必须执行与任务相关元素任务。通过任务元素您可以将资源分配给任务本身，而不仅仅是父元素。您可以创建任务元素的层次结构或树结构，将大任务分解为单独的部分，并为每个部分分配不同的资源。

## 工具箱icon



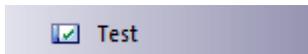
## 测试元素



测试A表示在用例或元素元素中创建的场景的基本路径、备用路径和异常路径中的一个步骤。测试元素是在测试用例元素中生成的。

每个测试元素左端都有一个状态带，该带采用颜色编码，以直观地表示元素属性中“状态”字段的值。元素右上角有一个标识“T”，您可以隐藏它如果您不想显示它。

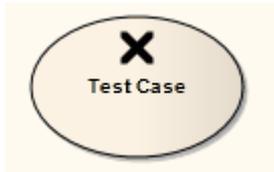
### 工具箱icon



### 注记

- 要切换元素右上角字母“T”的显示，请在“首选项”对话框的“对象”页面上选择或取消选择“显示需求的原型图标”复选框

## 测试用例



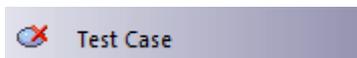
A测试用例是一个刻板的用例元素。您可以使用它来扩展测试窗口的功能，通过将元素属性和功能应用于由另一个元素或更合适的元素集表示的特征的测试。也就是说，您可以在测试用例元素的测试案例窗口中一次性定义适用于每个元素的测试的详细信息，而不是在每个元素中单独记录详细信息。

在测试用例元素属性中，您可以定义测试需求和约束，并将测试与测试文件相关联。您还可以将元素链接到文档工件或（在企业、统一和终极版本中）直接链接到链接文档，例如测试计划。

测试用例元素使您能够在浏览器窗口、图表列表、包浏览器、模型搜索、关系矩阵、可追溯性窗口和报告中更好地了解测试。

测试用例元素可通过图表工具箱的“用例”和“维护”页面获得。

### 工具箱icon



# 设计模式

设计模式是用于解决经常出现的设计问题的模板。设计模式由元素和连接器组成，可以在新的时间上下文。使用这些模式的优点是它们已经在许多上下文中经过测试和改进，因此通常是常见问题的稳健解决方案。

Enterprise Architect为创建和使用设计模式提供了广泛的支持。模式通常由经验丰富的建模者创建，他们可以了解如何从具体模型中提取抽象问题和解决方案。模式用户必须能够识别要使用的正确模式，并且必须为时间中的模式元素选择适当的上下文。

可以从任何图表中保存模式，创建一个描述该模式的 XML 文件；这些文件可以作为资源导入存储库，然后可以在任何上下文使用。

## Sparx-Created GoF模式

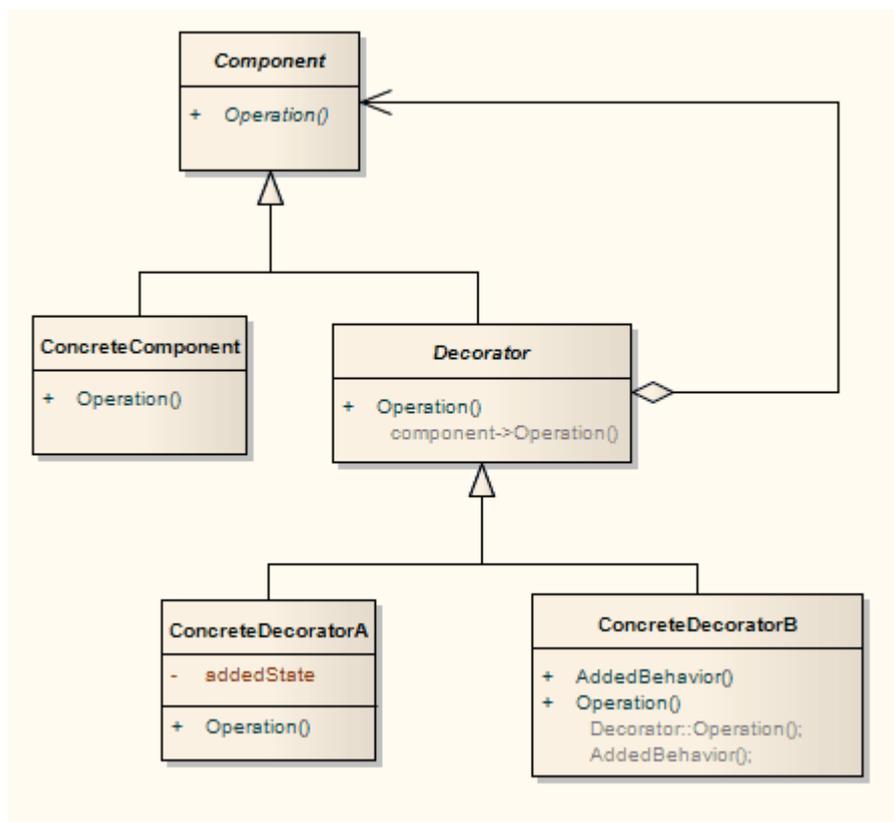
为了帮助您开始在Enterprise Architect中使用设计模式，Sparx Systems为您提供了最初发表在Gamma等人设计模式-元素*Reusable Object-Oriented Software*一书中的模式，称为“Gang of Four”或GoF模式。启用GoF技术后，您可以通过一组工具箱页面访问这些模式。

## 注记

- 您可以使用 设置>模型>传输>导出参考”和 导入参考”功能区选项在项目之间传输浏览器窗口的 资源”选项卡中列出的所有模式

## 发布模式

要发布设计模式，您首先必须将模型模式作为Enterprise Architect中的图表。此示例图是根据Gamma等人的 GoF 书籍设计模式元素*Reusable Object-Oriented Software*中的一个示例创建的。



### 访问

功能区	打开图表： 特定>技术>发布技术>发布图表模式
-----	----------------------------

### 定义模式文件

字段/按钮	行动
模式名称	类型模式名称。
文件名	类型包含已发布模式的目录路径和 .XML 文件名。
类别	类型模式应在“模式”中列出的类别（必需）。

版本	类型模式版本号。
注记	类型上的任何注记模式
行动	<p>选择适当的复选框以选择模式中包含的元素的<code>操作</code>；这些动作在使用该模式时执行。</p> <p>可用的操作是：</p> <ul style="list-style-type: none"> <li>• <code>Create</code>：直接创建元素模式，无需修改</li> <li>• <code>合并</code>：将元素模式与现有元素合并，使现有元素承担选定元素模式的角色</li> <li>• <code>Instance</code>：创建模式元素作为现有元素的实例</li> <li>• <code>类型</code>：将元素模式创建为现有元素类型</li> </ul> <p>如果您包含一个物件元素模式，您将使用 <code>实例</code> 将物件的分类器设置为图表中要删除该模式的类之一。</p> <p>如果您包含一个属性模式（<code>端口模式</code>或<code>部件</code>），您将使用 <code>类型</code> 将属性的类型设置为图表中要删除的类之一。</p>
确定	<p>单击此按钮两次以发布模式。</p> <p>发布后，您可以将模式加载到Enterprise Architect中，进入浏览器窗口的 <code>资源</code> 选项卡。</p>

## 注记

- 在Enterprise Architect的企业版、统一版和终极版中，如果启用了安全性，您必须具有 `管理图表` 权限才能将图表模式为
- 如果您的源图包含信息流，则不会将 `Information`项和 `Information Flows Realized`数据复制到模 中
- 要更改其中一个元素的名称，请双击元素以显示 `编辑`对话框；在此对话框中，您还可以添加详细说明元素用途的注释
- 不能为序列图发布模式

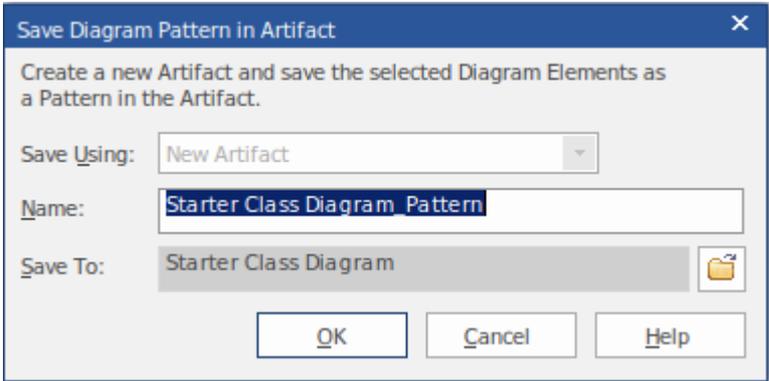
## 工件一个模式

Enterprise Architect使您能够在图表中创建选定的元素，并将模式模式存储为模型中的工件。然后，您可以将存储模式工件到模型中的任何图表中以重新创建模型。这类类似于将图表作为模式发布，除了：

- 仅将图表中的选定元素另存为模式
- 该模式工件在模型中，而不是文件系统中的 XML 文件

### 将元素模式为一种工件的图表

按照本表中的步骤操作。

节	行动
1	打开相应的图表，按住 Ctrl 键并单击要包含在模式中的每个元素。您还可以 拖动“图表中的一组元素以另存为模式”。
2	<p>右键单击选定元素之一，然后单击 将选定元素另存为模式”选项。将显示 保存图表模式”工件。</p> 
3	如果 保存使用”字段不显示值 工件”，请单击下拉箭头并选择此值。如果这是工件中的第模式模型，则该字段默认为此值作为它可以具有的唯一值。
4	名称”字段默认为父图名称加上 “_Pattern”。要么留下这个名字，要么用你喜欢的模式名改写它。
5	保存到”字段默认为图表父包的名称。要么点击包名称，要么点击  图标并浏览以创建工件的包。
6	点击确定按钮，在选定的包下生成图模式工件。选定的工件被保存为一个模式。

### 从工件应用图表模式

按照本表中的步骤操作。

节	行动
---	----

1	<p>打开图表，您将把模式粘贴到工件图表中。                  图表必须是图形视图，而不是内部规范视图、甘特视图或列表视图。</p>
2	<p>找到所需的图模式 在浏览器中将原型工件打开的图表中。新元素和连接器从模式中生成图表。</p>

## 工件一个模式

更新模式有工件类似的方法。

方法	描述
1	<p>按照步骤1和2 在新工件中的模式1 和 2 中图表元素 保存“使用 现有工件表设置”。</p> <p>单击 名称”字段和要更新的模式名称。 保存到”字段显示为灰色，因为它使用现有工件的包地址。</p> <p>点击确定按钮；新模式的工件已更新。</p>
2	<p>打开包含构成模式的元素的图表。</p> <p>在浏览器中，点击图模式可以用新模式工件。</p> <p>在图表中，选择所需的元素，然后右键单击并选择 将所选元素保存为模式”选项。</p> <p>'Save图表模式' 对话框显示 'Save Using' 字段工件工件' 和其他两个灰色字段。 名称”字段显示工件名称的名称。</p> <p>点击确定按钮；所选工件会随着新模式更新。</p>

## 注记

- 在Enterprise Architect的企业版、统一版和终极模式中，如果启用了安全性，您必须拥有 更新图表”权限才能从图模式模式 在图表中工件印象

# 导入模型模式

在能够在您的模型中使用自定义模式之前，您必须先将模式XML 文件导入Enterprise Architect安装路径中的模型模式目录；然后可以从浏览器窗口的 资源”选项卡中获得它，也可以从工具箱中获得它。

## 访问

使用此处概述的方法之一来显示浏览器窗口的 资源”选项卡。

在浏览器窗口的 资源”选项卡中，右键单击 模型模式|导入模型模式”。

功能区	>开始> 所有窗口>设计> 通用 .浏览 >资源 探索 >门户>窗口> 探索 >资源
键盘快捷键	Alt+6

## 导入模型模式

节	行动
1	在 导入模型模式”对话框中，输入或浏览要导入的 XML 文件的名称。
2	选择将文件导入模型或用户 APPDATA 位置。
3	点击确定按钮导入模式。 导入的模式被放置在 XML 文件中定义的适当类别中；如果 模型模式”下不存在该类别，则创建一个新类别。

## MDG 技术中的模式

许多技术A了自己的模式，有些技术主要设计为使特定模式可用于模型的工具，例如四人帮模式的技术。这些是通过浏览器窗口的 资源模式”选项卡和该技术的图表工具箱页面提供的。如果要使用这种模式，请检查模型中是否已经加载并启用了相应的技术。

# 使用模式

使用设计模式，您可以快速创建在其他情况下执行相同类型任务的代码结构的模板解决方案，并使用模型中定义的模式与模型。

## 访问

使用此处概述的方法之一来显示浏览器窗口的“资源”选项卡。

功能区	>开始> 所有窗口>设计> 通用 > 浏览 >资源 探索 >门户>窗口> 探索 >资源
键盘快捷键	Alt+6

## 之前导入模型的使用a模式

节	行动
1	打开要添加模式的图表。
2	选择浏览器窗口的“资源”选项卡。
3	展开文件夹“模式”并根据需要展开子文件夹，直到找到您需要的模式。 您可以通过右键单击名称并选择“视图模式详细信息”选项以只读模式查看模式详细信息。
4	任何一个： <ul style="list-style-type: none"> <li>选择“模式到模型图表”上下文菜单选项或</li> <li>将浏览器窗口的“资源”选项卡中的模式拖放到图表上</li> </ul> '模式<pattern group> <pattern name> to图表'对话框显示。
5	完成对话框，根据需要进行选择。 做出适当的选择后，单击确定按钮将模式导入模型，使用新的 GUID 重新创建原始图表。

## 更改元素模式的默认值

节	行动
1	在“添加模式<pattern group> <pattern name> to图表”对话框中，确保未选中“导入as包Fragment”选项。
2	在“模式元素”面板中选择单个元素。

3	<p>单击项目行末尾的  按钮以显示“编辑”对话框。</p> <p>更改元素名称的具体方法取决于“元素”面板的“行动”列中的模式。</p>
4	<p>如果元素的“行动”条目是“创建”，则在“编辑”对话框的“默认”字段中删除现有值并键入您自己的用户定义值。</p> <p>点击确定按钮。</p> <p>元素默认值在“添加模式...”对话框中更新。</p>
5	<p>如果元素的“行动”条目是“合并”，则在“编辑”对话框中单击  按钮以浏览到现有的元素分类器。</p> <p>将显示“选择 &lt;Item&gt;”对话框。</p>
6	<p>找到并选择现有的元素分类器。</p> <p>您可以通过从特定命名空间中选择元素来限制选择的数量；为此，请单击“在命名空间中”下拉箭头并选择一个命名空间。</p>

## 添加模式对话框

当您使用或编辑设计模式时，会显示“添加元素模式<pattern group> <pattern name> to图表”对话框。

### 参考

选项	行动
模式元素	<p>访问模式中包含的单个元素。</p> <p>从这里您可以：</p> <ul style="list-style-type: none"> <li>通过单击下拉箭头选择单个元素的操作（创建、合并、实例或类型，适用于每个元素），或者</li> <li>修改元素模式的默认值或 - 对于合并的元素- 通过单击默认条目右侧的  按钮来选择命名空间</li> </ul>
预习	显示模式的预览。
使用自动名称	如果要应用为项目定义的元素自动命名约定，请选中此复选框。
元素注记	<p>显示描述模式中元素的注释。</p> <p>突出显示“模式元素”面板中的元素以查看注记。</p>
导入为包Fragment	<p>选择此选项可在新图表而不是当前图表上导入模式。</p> <p>点击确定按钮。Enterprise Architect将：</p> <ul style="list-style-type: none"> <li>在浏览器窗口中当前选中的包下新建一个与模式同名的包</li> <li>在这个包下新建一个与模式同名的图</li> <li>导入模式导入新图表</li> </ul>

### 注记

- When the option '导入as包Fragment' is selected, the '模式元素' section and '使用自动Names' option will become disabled

