

# UML 顺序图建模（建模工具 EA）

李澎涛 俎涛（火龙果软件工程）

## 顺序图简介

顺序图是交互图的一种形式，它显示对象沿生命线发展，对象之间随时间的交互表示为从源生命线指向目标生命线的消息。

顺序图可用于：

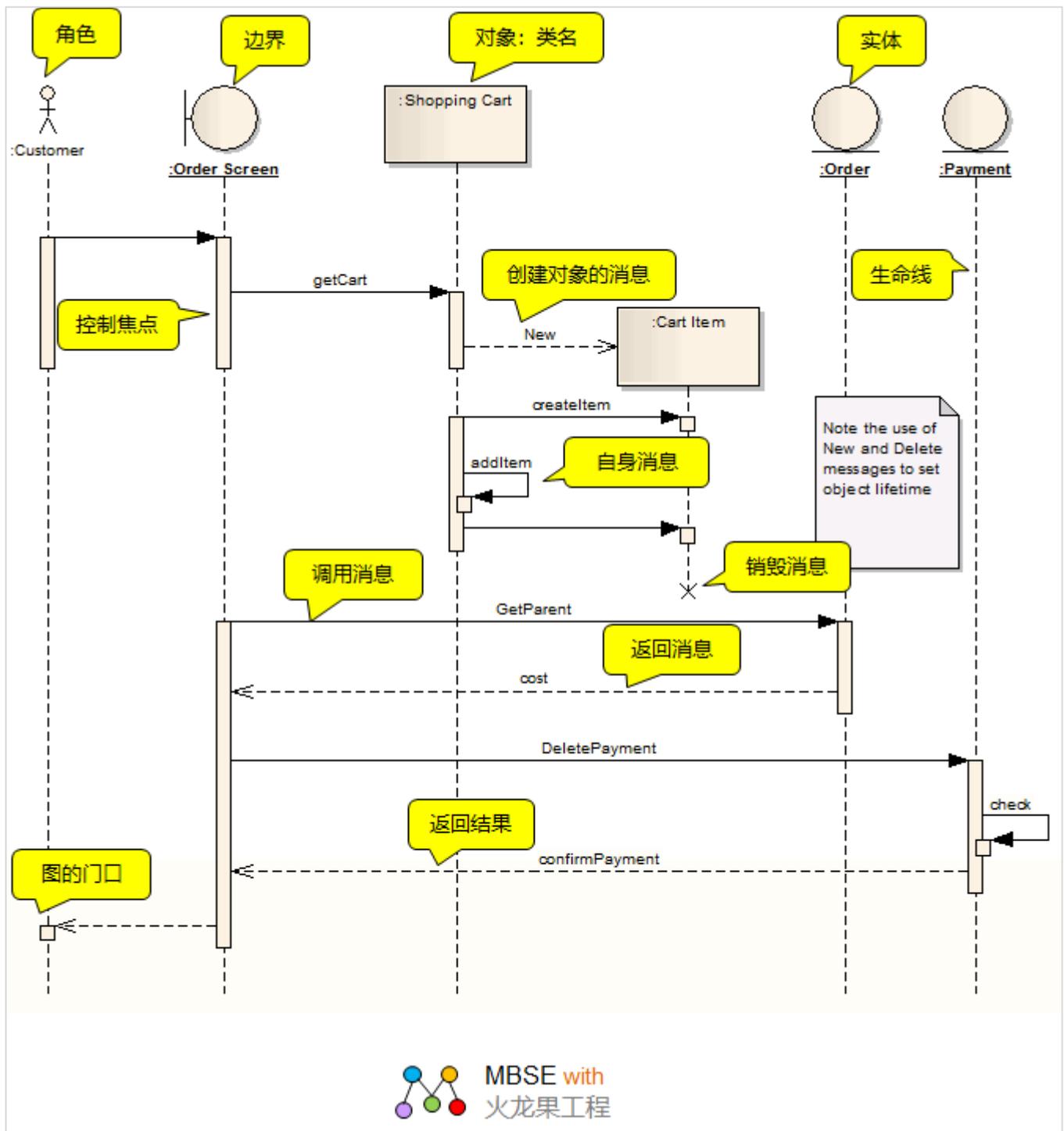
- 描述 workflow、消息传递以及各元素如何随着时间的推移相互协作。
- 在分析的早期，捕捉整个系统的信息流和职责；元素之间的消息最终成为 Class 模型中的方法调用。
- 为用例场景建立解释性模型；通过创建一个包含参与者和用例中涉及的元素的顺序图，可以对用户和系统完成所需任务的步骤顺序进行建模

## 顺序图的构成

- 序列元素按水平顺序排列，消息在元素之间来回传递
- 顺序图上的消息可以有多种类型；消息还可以配置为反映源和目标元素的操作和属性
- Actor 元素可用于表示启动事件流的用户
- 模式化的元素，如边界、控制和实体，可以分别用于说明屏幕、控制器和数据库项目
- 每个元素都有一个称为生命线的虚线词干，该元素存在于其中并可能参与交互

## 顺序图示例

如下是一个顺序图的示例，其中标识了元素和关系的类型。

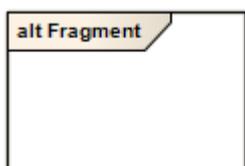


## 顺序图的元素

图标	元素	描述
 Actor	角色	Actor 是系统的用户；用户可以指模型中的人类用户、机器，甚至是另一个系统或子系统。

 Lifeline	生命线	生命线代表一个独特的可连接元素,是交互中的个体参与者。
 Boundary	边界类	Boundary(边界类)是一种类的构造型,它对某些系统边界进行建模,通常是用户界面。在分析中使用边界来描述用户和系统的交互。
 Control	控制类	Control(控制类) 是一个类的构造型,它为控制实体或管理器建模。控制类用于控制和调度其他的类,通常用于分析(包括稳健性)、序列和通信图中。
 Entity	实体类	Entity(实体) 是一个类的构造型,它建模系统中的信息或存储的数据。
 Fragment	组合片段	Fragment 元素可以表示顺序图中的迭代或有条件执行的交互过程。
 Endpoint	端点	端点用于交互图中以按顺序反映丢失或找到的消息。
 Diagram Gate	图的关口	Diagram Gate 是一种简单的图形方式,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	状态常量 / 延续,元素为顺序图提供了两个不同的目的,即状态不变量和延续部分。
 Interaction	交互	可以使用交互元素将交互图作为类元素的子元素插入。

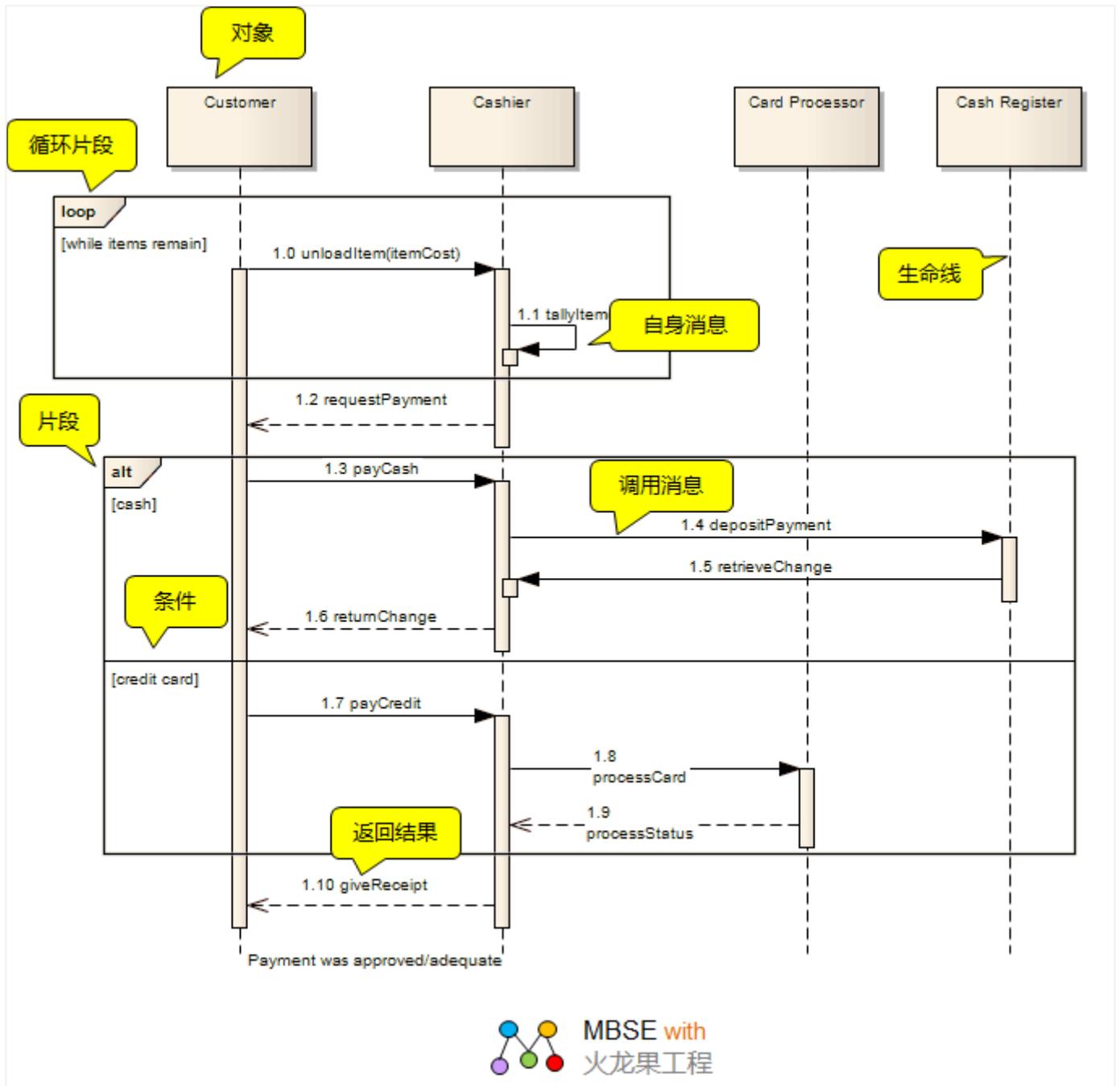
### 图例: Fragment 图例



Fragment(片段) 是顺序图中对一个交互片段的组合。一个复杂的顺序图可以划分为几个部分,每一个部分称为一个交互片段,每个交互片段由一个大方框包围,其名称显示在方框左上角的间隔区内,表示该顺序图的信息。

组合片段可以包含几个片段，每个片段都有布尔条件约束。组合片段 显示为一个透明窗口，其中的每个片段由水平线分隔，可以具有自己的触发条件。

下面是组合片段的示例：



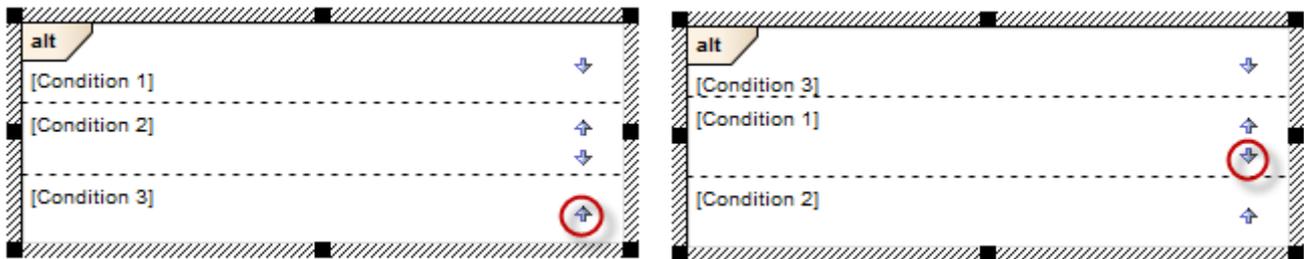
此顺序图说明了如何使用组合片段对简化的采购流程进行建模。

在如上的顺序图中有 2 个 fragment（片段）

- 其中一个循环片段，表示收银员（Cashier）对每个购买的商品进行计价，直到完成所有商品的计价，然后收银员（Cashier）请求顾客（Customer）付款。
- 另一个是条件判断组合片段，表示对支付方式进行判断，然后被分割以显示现金和信用卡这两个条件的处理片段。在支付片段执行完成后，收银员向客户提供收据。

交互片段条件的顺序可以直接在图上改变：

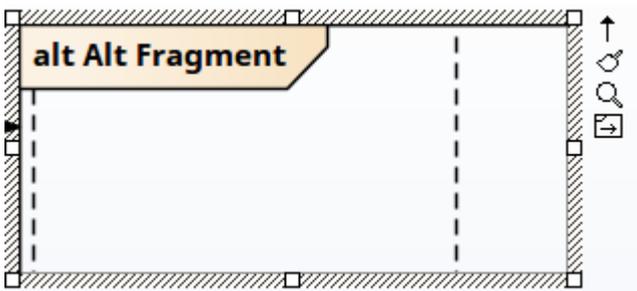
1. 选择一个定义了多个条件的交互片段；向上和向下箭头出现在每个条件的右侧。
2. 单击相应的箭头以更改顺序。



## 操作指南：选择和移动组合片段

为了选择一个组合片段，必须在靠近内边缘处单击或在片段周围拖动一个选择矩形。

一旦包含在片段或片段条件中，消息在图表中上下移动时将继续包含在其中。要将消息从片段中移出，或移到片段内序列中的不同位置，请在将消息拖动到位时按住 **Alt** 键。当其中的消息向上或向下移动时，顺序图上的片段将调整大小，以继续包含该消息。



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

除非片段处于“move freely”模式，否则组合片段条件内的交互片段不能移动到条件之外。移动条件行会根据条件行的移动量向下或向上移动该行下方的任何对象和消息。

当包含的片段被调整大小时，包含其他片段的片段会调整大小（除非该片段处于“move freely”模式）。

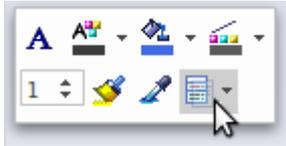
## 操作指南：填充不透明度

虽然交互片段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。可以通过更改元素的不透明度，将这些细微差别应用于交互片段后面和覆盖或重叠的元素的显示。

在设置不透明度之前，请检查元素是否具有填充颜色。

可以使用以下两个弹出元素工具栏中的图标设置不透明度：

- 单击交互片段元素和  图标：



- 右键单击 Interaction Fragment 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%，其中交互片段后面和重叠或覆盖的元素被隐藏（可以右键单击单个元素并选择“Z-Order | Bring to Top”选项以仅显示这些元素）
- 0% 表示没有不透明度，其中不应用填充颜色并且交互片段后面的任何内容都完全可见
- 75%、50% 或 25% 设置适当的不透明度并使被覆盖的元素可见但阴影过度

## 图例：端点

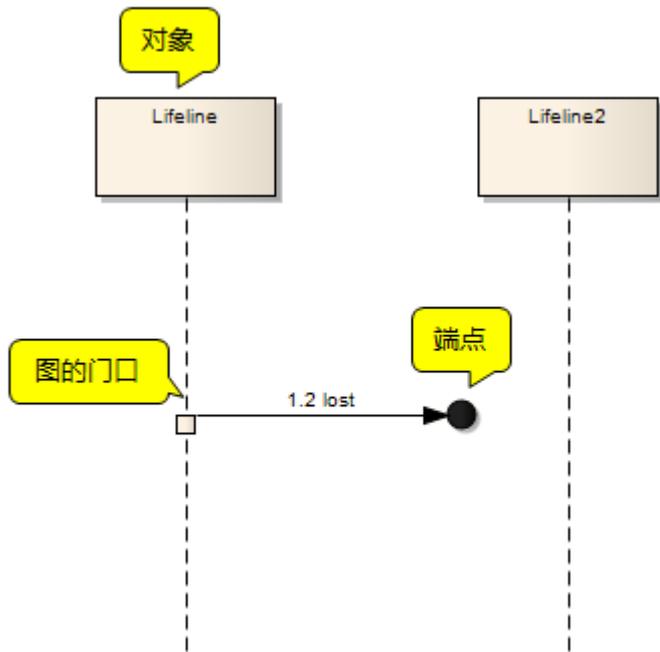


消息端点元素定义时间图中状态或数值生命线的终止。它表明消息：

- 起源于生命线外的未定义点，可以称之为消息找到。
- 发给在对象生命线外的未定义点，可以称之为消息丢失。

端点用于交互图（序列、时序、通信或交互概述）以按顺序反映丢失或找到的消息。

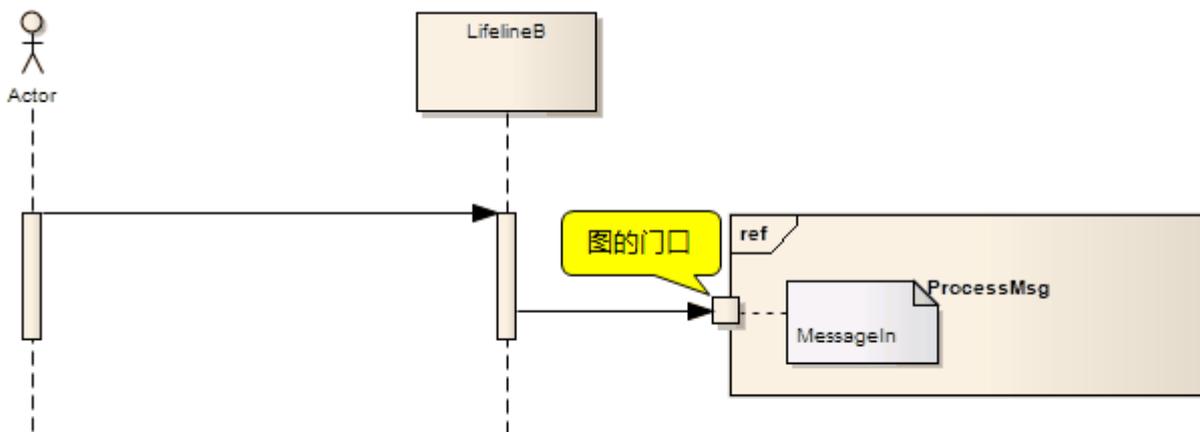
使用顺序图，将消息从适当的生命线拖到端点。对于顺序图，将生命线连接到端点的消息需要一些顺序规范来绘制连接。如下的此示例描述了顺序图中丢失的消息。



### 图例：Diagram gate(图的门口)



Diagram gate 是一种简单的图形化方法，用于指示消息可以传输到交互片段或传输到交互片段之外的位置。可能需要一个片段来接收或传递消息；在内部，有序消息反映了这一需求，在片段框架的边界上指示了入口。任何与此内部消息同步的外部消息必须适当对应。入口可以出现在交互图(序列、时间、通信或交互概述)、交互事件和组合片段(以指定表达式)中。



## 图例：状态/持续



State/Continuation

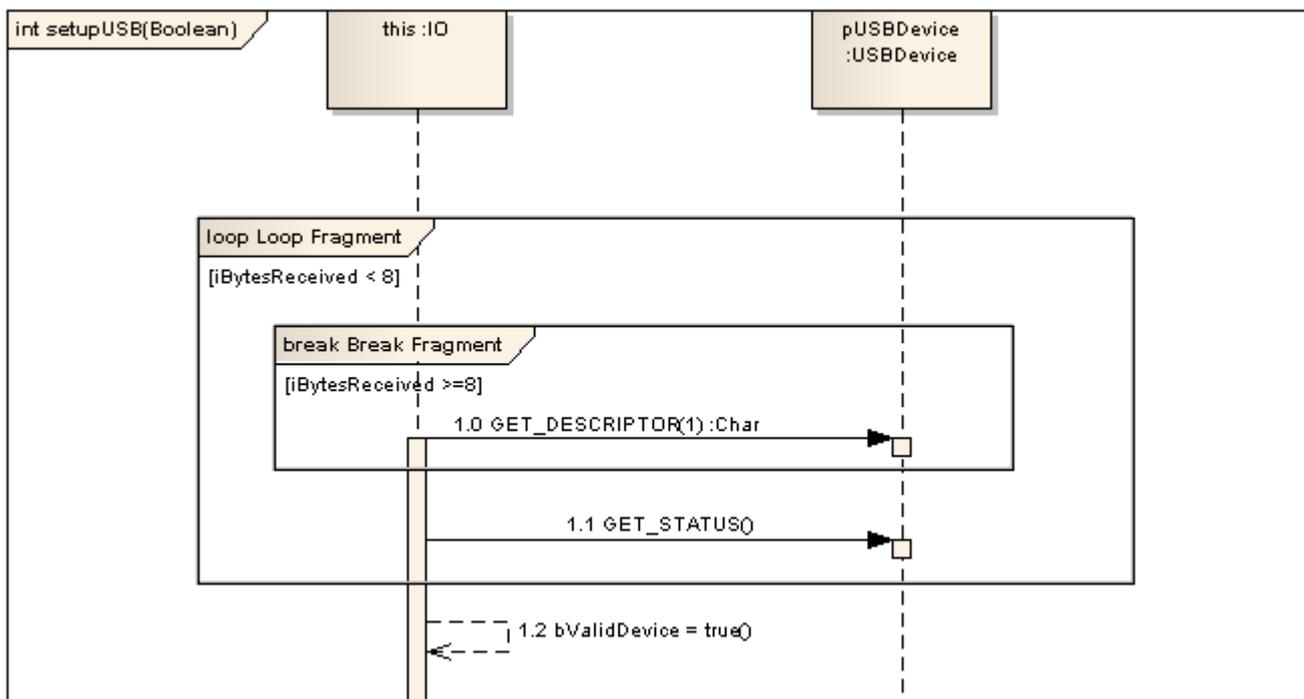
状态/持续-元素对于交互（序列）图有两个不同的用途，即状态不变量和持续。创建元素时，系统会提示确定用途。

## 图例：交互作用



可以使用交互元素将交互图作为类元素的子元素插入。交互元素可以包含以下任何类型的图表：

- 序列
- 通信
- 时间



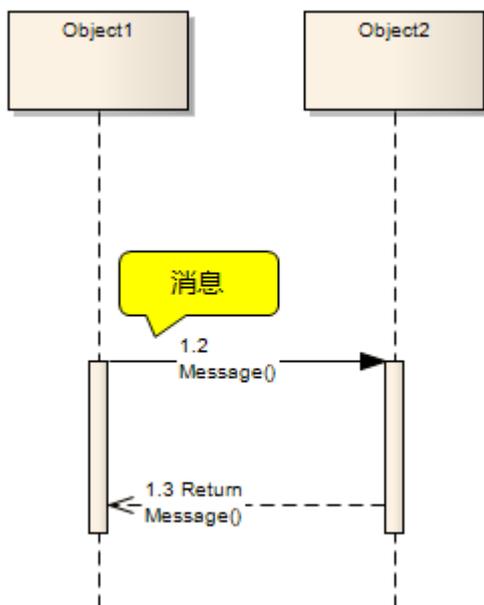
EA 中的交互元素被视为封装在其中的分类器的行为。它可以具有参数和返回类型，这些参数和返回类型使用交互元素的“Properties”对话框的“Behavior”选项卡进行建模。

## 顺序图的关系

图标	元素名	描述
 Message	消息	消息指示元素之间的信息流或控制转换。
 Self-Message	自身消息	自身消息反映了在调用生命线的操作中调用的新流程或方法
 Recursion	递归	递归是一种用于顺序图中的消息类型，用于指示递归函数。
 Call	呼叫	呼叫是一种消息连接器，它扩展了上一条消息的激活级别。

### 消息 (Message)

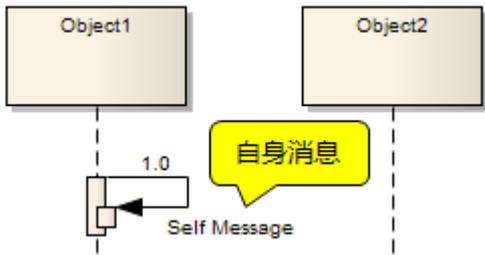
顺序图使用从元素到元素传递的消息来描述随时间变化的工作流程或活动。



### 自身消息

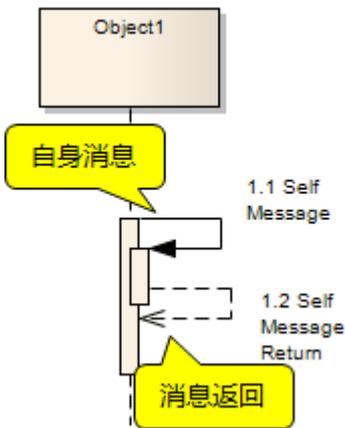
Self-Message 反映了在调用生命线的操作中调用的新流程或方法。它是消息的规范，通常在顺序图中。

Self-Message Calls 表示嵌套调用；每次调用都会添加新的激活级别。

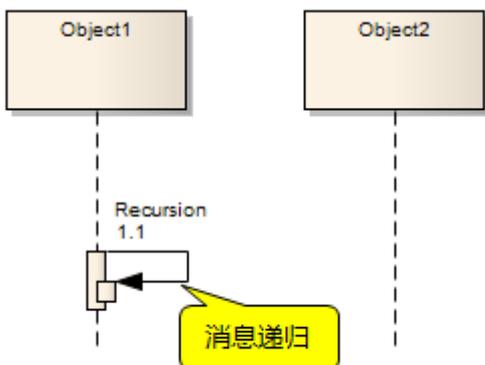


### 自身消息返回

可以描述自身消息调用的返回。

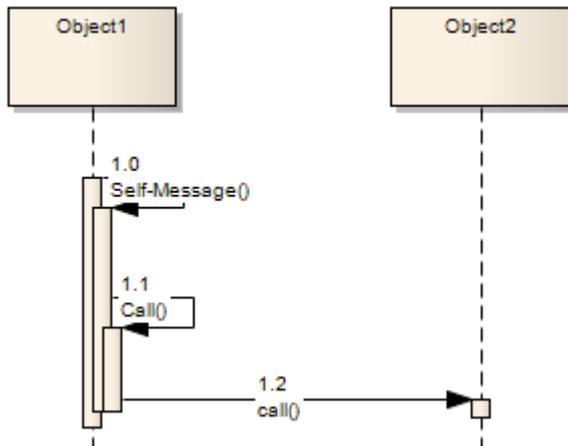


### 递归



内部通信表现为一个操作的递归调用，或一个方法调用属于同一个对象的其他方法。显示为生命线上执行事件的嵌套控制焦点。

## 呼叫

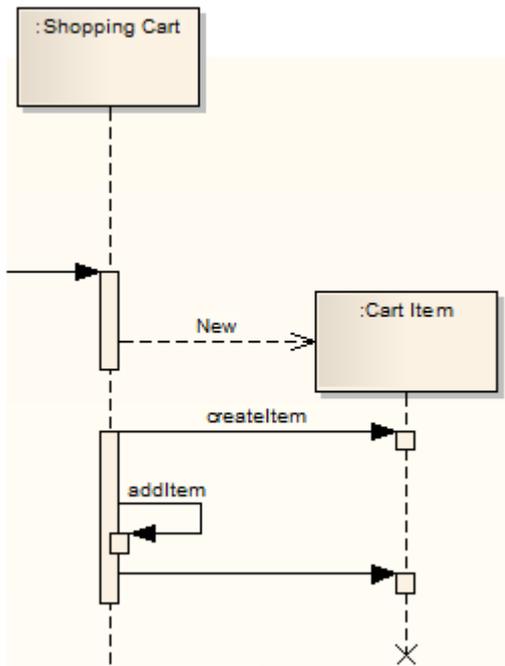


呼叫是一种消息连接器，它扩展了上一条消息的激活级别。所有的自我信息都会创建一个新的激活级别，但这种控制重点通常以下一个信息结束（除非手动调整激活级别）。自我消息调用，如第一个调用的图像所示，表示嵌套调用；每次调用都会添加新的激活级别。与元素之间的常规消息不同，元素之间的调用会继续源元素中的现有激活，这意味着调用是在前一个消息的激活范围内发起的。

## 顺序图建模指南

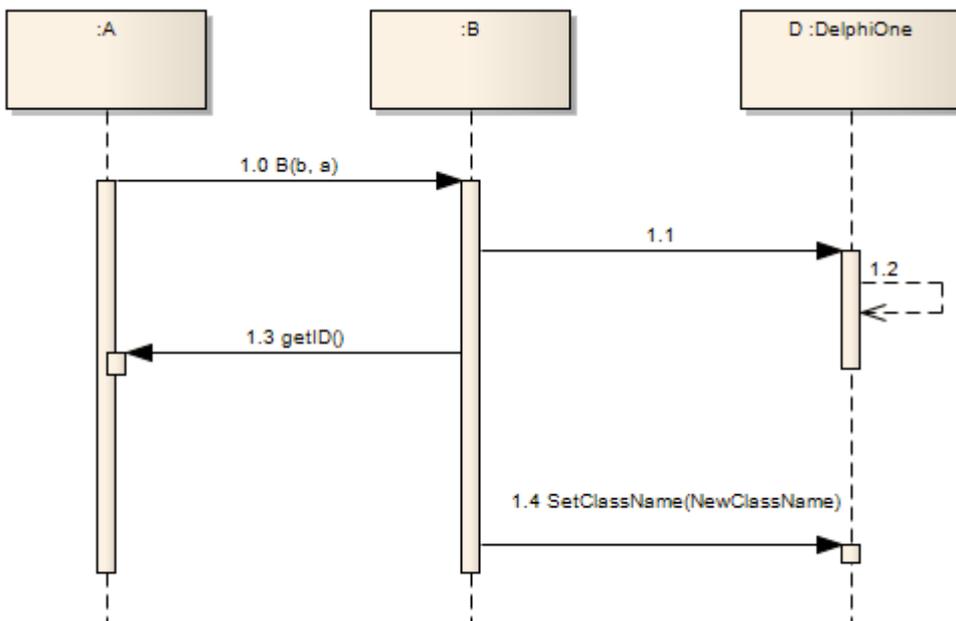
### 表示元素的生命周期 (Denote the Lifecycle of an Element)

使用表示为 New 或 Delete 消息类型的消息捕获元素生命周期



此示例显示了具有特定创建和删除时间的两个元素。

### 顺序图的布局 (Layout of Sequence Diagrams)

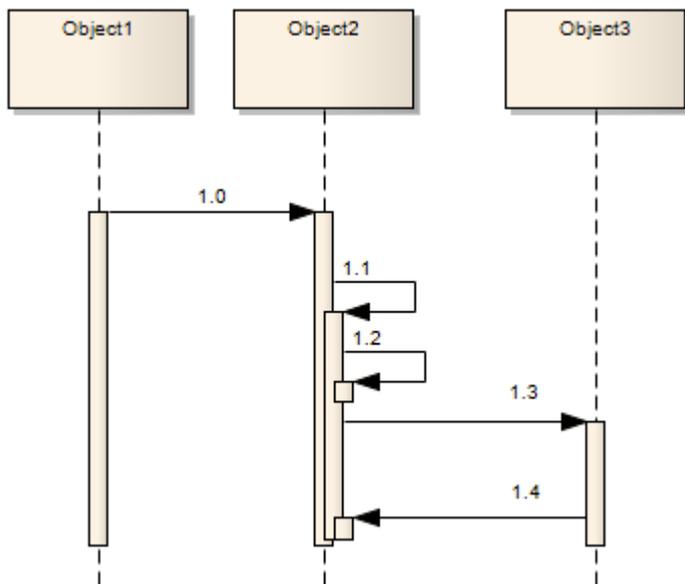


这个例子展示了顺序图的布局使用。

## 激活序列元素 (Sequence Element Activation)

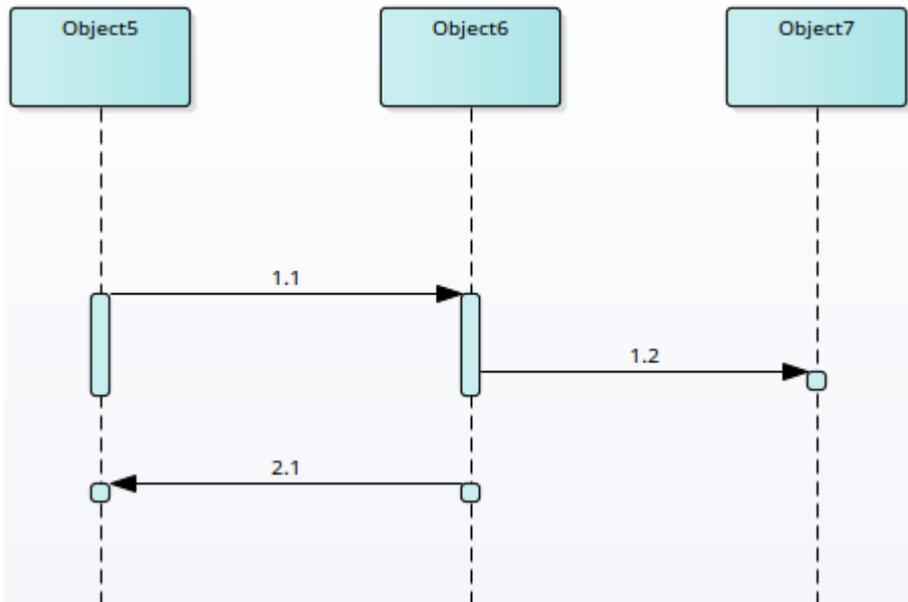
顺序图中的序列元素具有沿其生命线绘制的矩形。这些矩形描述了元素在整个处理期间处于活动状态的时间。这种可视化的表示可以通过右键单击顺序图，并选择“Suppress Activations”来抑制。

通常，EA 会为计算激活的周期。在 Sequence Message 上有几个上下文菜单选项，可以使用它们来完成这个任务。要访问上下文菜单，右键单击消息并选择“Activations”。



## 带有先前消息的分支

使用适当的消息分组和激活级别设置生命线和序列消息后，可能希望指示从生命线发出的不同消息组中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：

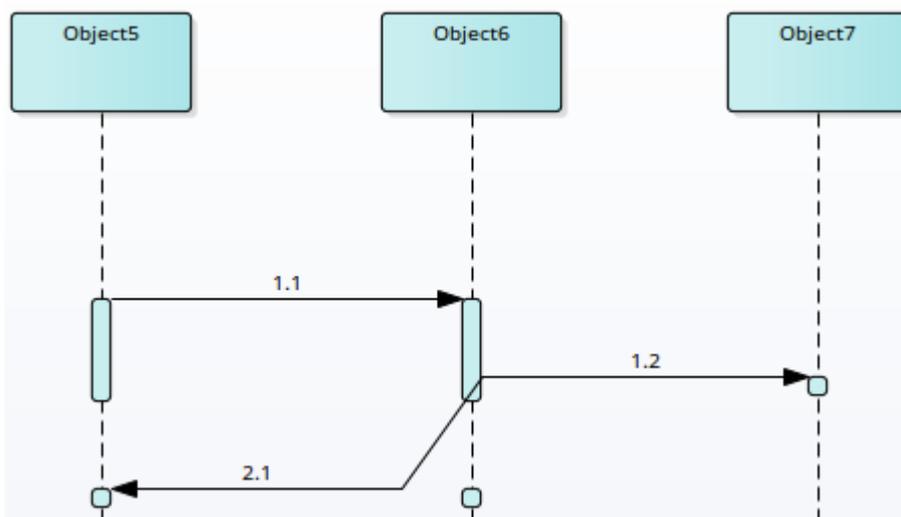


说明：消息 1.1 从对象 5 传递到对象 6，然后消息 1.2 传递到对象 7，消息 2.1 传递回对象 5。消息似乎按照 1.1、1.2 和 2.1 的顺序进行。但是，想指出消息 2.1 虽然是分开的，但与消息 1.2 是并发的。

在这种情况下：

- 右键单击后面的消息 (2.1) 并选择选项“带有上一条消息的分支”

消息 2.1 的源锚点随后变为与消息 1.2 的源相同，即前一条消息。它们是来自同一生命线的独立但并发的消息。

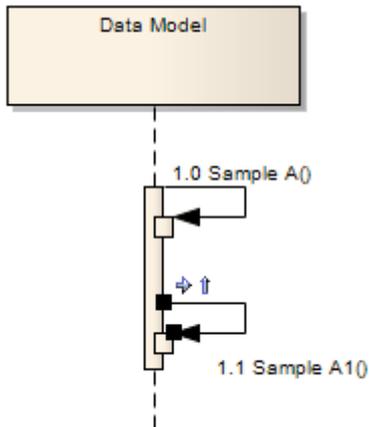


如果以后不需要显示消息是分支，请右键单击后面的消息 (2.1) 并取消选择“带有先前消息的分支”选项。

## 生命线激活级别 (Lifeline Activation Levels)

通过在单个生命线上添加激活层，复杂的处理系统可以很容易地反映在顺序图中。

示例 1:

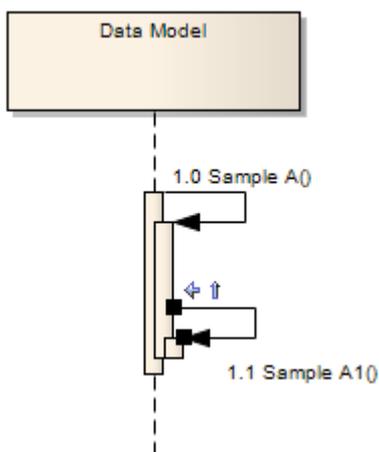


类调用方法 `Sample A`，该方法又调用 `Sample A1`。

操作如下：

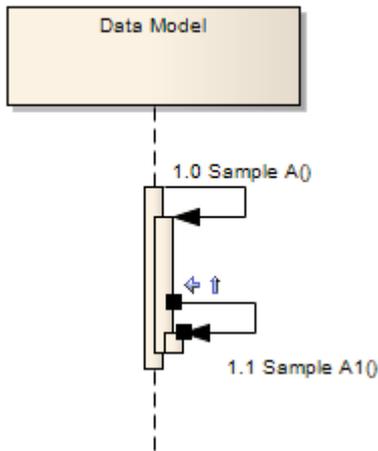
1. 在图表工具箱中单击  以显示“Find Toolbox Item”对话框并指定“Interaction”。
2. 单击“Self-message”面板中的“Interaction Relationships”图标。
3. 点击生命线。

示例 2:



生命线现在可视化地描述了在处理 `Sample A` 期间调用的方法 `Sample A1`。

示例 3:



在本例中，添加了更多的自身消息。

消息 Sample A2a 从 Sample A2 调用，而 Sample A2 又从 Sample A(而不是 Sample A1)调用。

Sample A1 是从 Sample A 中调用的。

## 序列消息标签可见性 (Message Label Visibility)

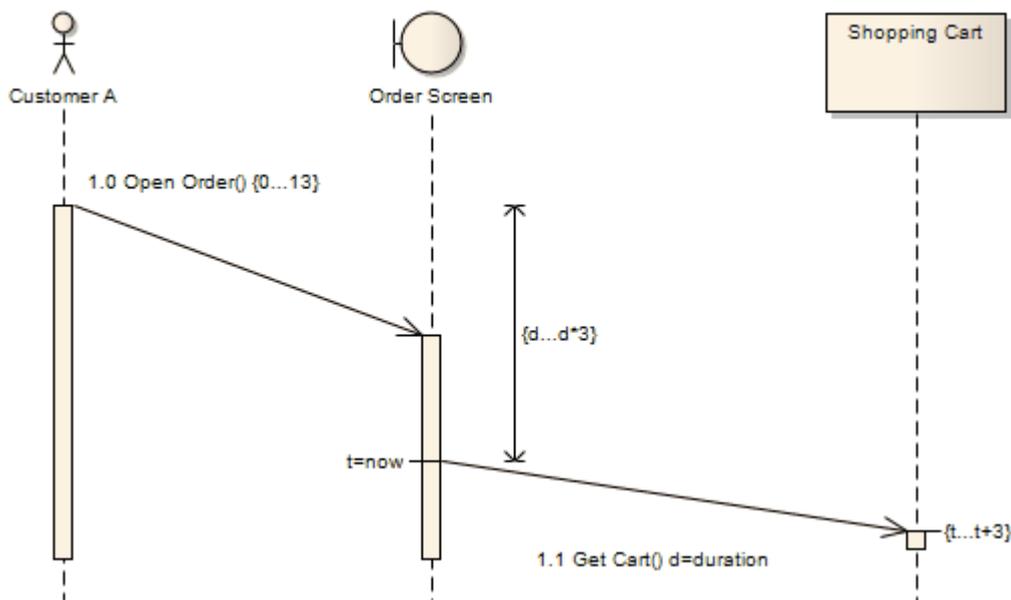
隐藏和显示序列消息中使用的标签，操作如下：

1. 右键单击序列图中的消息，并选择“Set Label Visibility”。
2. 显示“Label Visibility”对话框。
3. 分别选择或清除针对每个要显示或隐藏的消息标签的复选框。
4. 单击 OK 按钮保存设置。

## 更改上边距 (在图表顶部创建可用空间) (Change the Top Margin (create usable space at the top of the diagram))

为了从默认的 50 个单位更改序列图的顶部边缘，右键单击图并选择“Set Top Margin”选项。可以将顶部空白设置为 30 到 250 个单位之间的任何值。然后，可以使用这个空间，例如，添加 Note 或 Text 元素来提供图表上的文档。

## 更改时间细节 (Change the Timing Details)



在此图中，在未结订单消息上：

- “持续时间约束”已设置为 0...13

在获取购物车消息上：

- “消息之间的持续时间约束”已设置为  $d \dots d * 3$
- “持续时间观察”已设置为  $d = \text{duration}$
- “时间约束”已设置为  $t \dots t + 3$
- “时间观察”已设置为  $t = \text{now}$

通过在“持续时间约束”字段中键入一个值，可以调整消息角度。

还可以通过将“General Ordering”箭头向上拖动到前一条消息加入当前消息的源生命线的点来创建“消息之间的持续时间约束”行。将显示一个对话框，可以在其中输入约束值。创建线后，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。可以通过“计时详细信息”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。

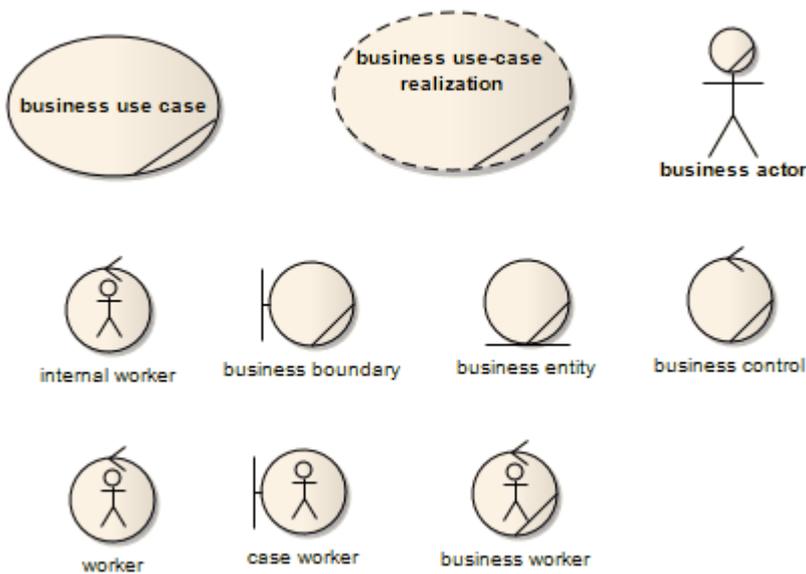
场地	行动
持续时间约束	指明消息可以持续多长时间的最小和最大限制。
消息之间的持续时间约束	指示在当前消息的源生命线发送或接收上一条消息与发送当前消息之间的最小和最大间隔。

持续时间观察	捕获消息的 <b>持续时间</b>
时序约束	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。

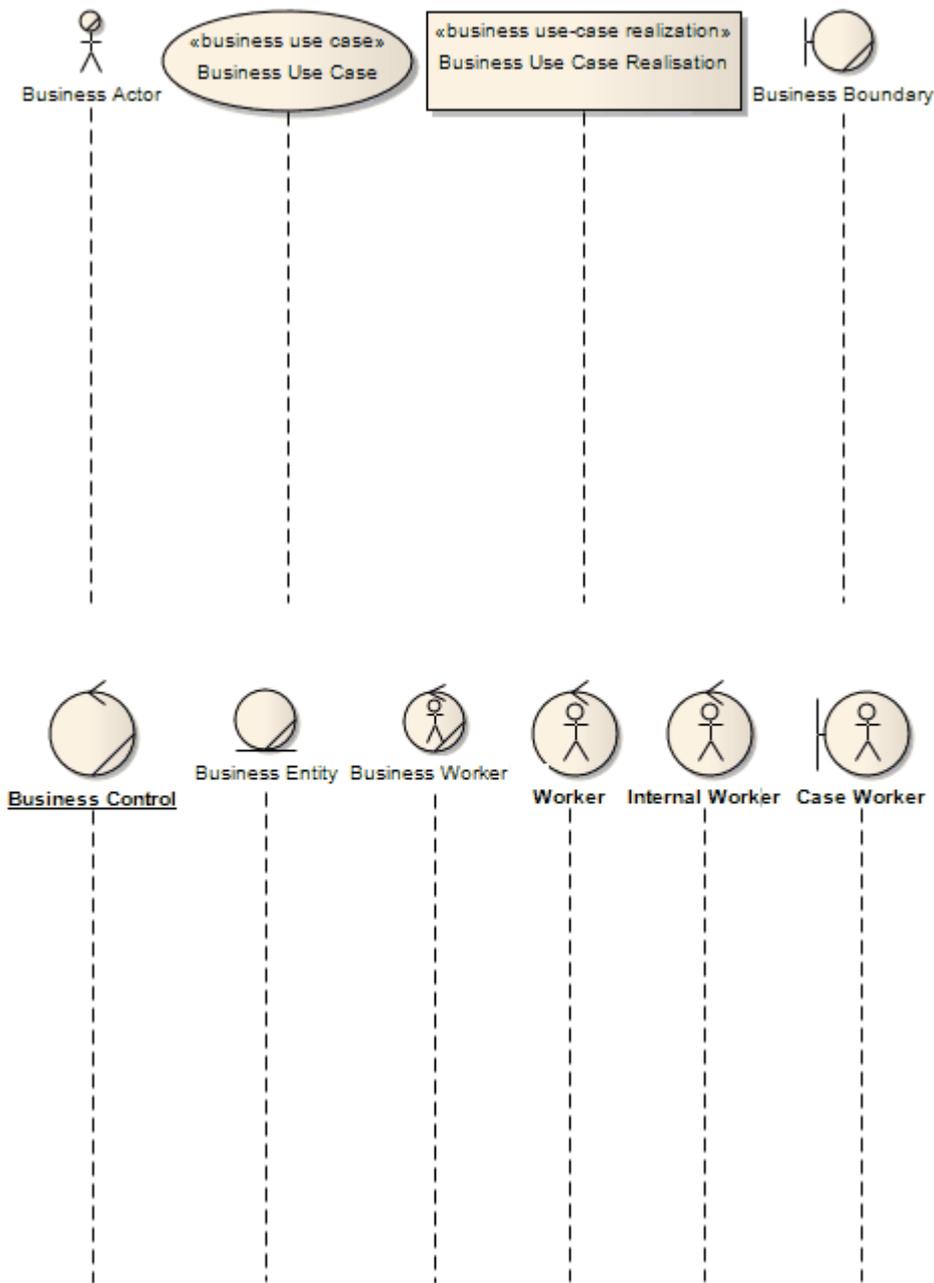
## 业务建模/交互 (Business Modeling/Interaction)

业务建模图和业务交互图能够对业务系统的结构和行为建模。业务建模图基于类(UML 结构)图，而业务交互图基于序列(UML 行为)图。这两种图表类型都具有相同的默认 Toolbox，它由 Business Modeling 元素页组成。可用的元素包括原型对象，原型参与者(业务参与者)，用例(业务用例)和协作(业务用例实现)。

该图显示了从图表工具箱的“业务建模”页面拖放到业务建模图上的元素的外观。



此图显示了从图表工具箱的“业务建模”页面拖放到业务交互图上的元素的外观。



## 顺序图和版本控制 (Sequence Diagrams and Version Control)

可以创建使用其他包中的元素作为图中生命线的序列图。在这种情况下，当元素包在版本控制下签入和签出时，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，尽管它们在图表中恢复，但连接它们的任何消息都没有。

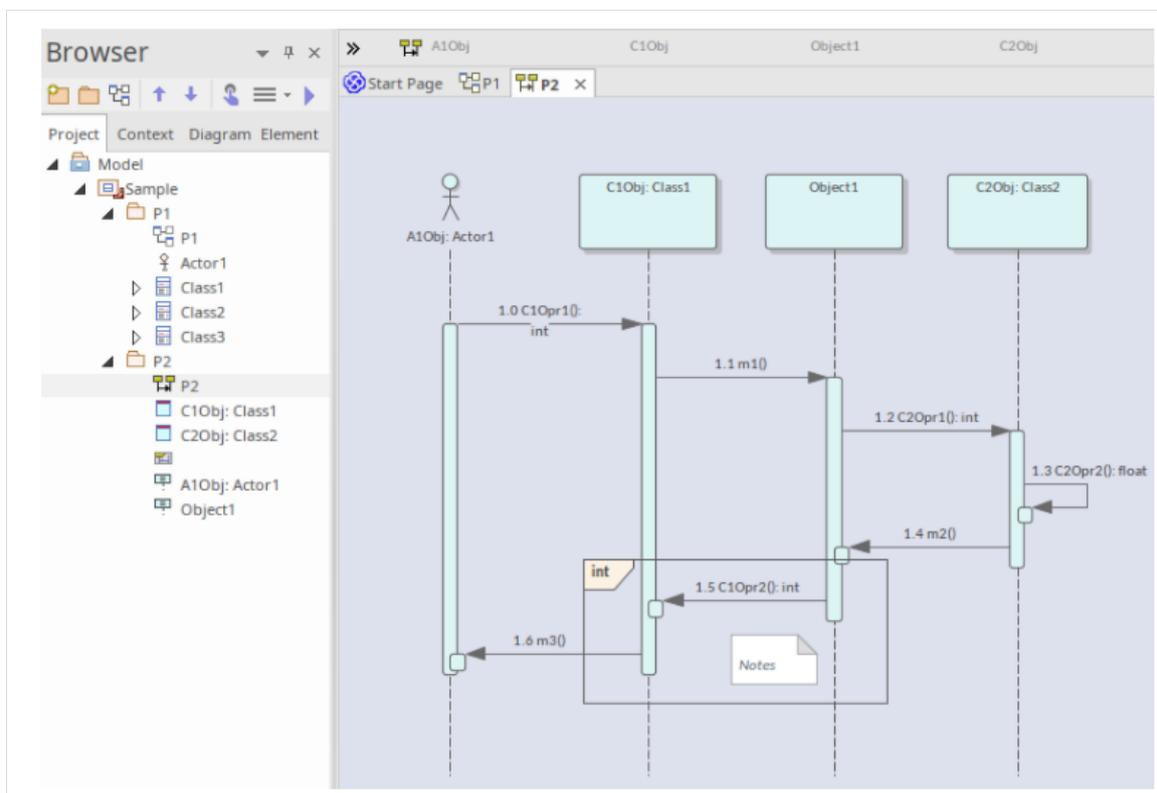
因此，如果图表及其元素位于不同的包中，则元素包通过版本控制的往返可能会损坏序列图。

解决方案是将每个类作为对象拖放到序列图上 - 当将类拖放到序列图上时，在“粘贴元素”对话框中选择“作为元素（对象）的实例”选项。这会根据选定的 Class 元素在图表的父包中创建一个新对象。然后，在对象之间创建消息。

因此，为了确保序列图不被其他包通过版本控制的往返损坏，请记住：

- 生命线必须是对象（即使可以将元素作为生命线拖放到序列图上，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。



此图在通过版本控制往返时不会损坏，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。

## 序列图选项 (Sequence Diagram Options)

当使用序列图建模时，有许多特定于该图类型的呈现功能，例如控制指示器的焦点。除了在“首选项”对话框的其他图表页面上定义的更一般的图表设置之外，还可以为这些功能定义默认设置。

---

## 后记

希望您读了此文后有所受益。

如果您有经验乐于分享，欢迎投稿给我们。

如果您对我们的培训、咨询和工具感兴趣：

- 课程：
  - [基于 UML 和 EA 进行分析设计](#)
  - [MBSE（基于模型的系统工程）](#)
  - [基于模型的需求管理\)方法与实践](#)
  - [基于 SysML 和 EA 进行系统设计与建模](#)
  - [企业架构建模](#)
  - [系统架构建模方法与案例](#)
  - [领域驱动的建模与设计](#)
  - [基于模型的设计](#)
  - [业务建模与业务分析](#)
- 工具：
  - 建模工具：[EA](#)
  - [CodeEngineer](#)
  - 模型 web 浏览工具：[WebEA](#)
  - 基于模型的需求管理工具：[iSpace](#)
- 咨询方案：
  - [MBSE\(基于模型的系统工程\)](#)
  - [基于 UML 的模型驱动的开发](#)

如果您希望进一步了解 EA：

- 欢迎访问工具频道 <http://tool.uml.com.cn>
- 也欢迎直接联系我们 [zhgx@uml.net.cn](mailto:zhgx@uml.net.cn) ， 010-62670969

本文使用的建模工具为 EA，可以下载试用版 <http://tool.uml.com.cn/ToolsEA/download.asp>。



MBSE with 火龙果



火龙果软件 (EA 全球合作伙伴)  
提供最专业建模方法与工具