

8 顺序图

8.1 交互图的种类

实例之间交互的模式由交互图来表达。基于同样的由交互指定的基础信息，交互图分成两种形式，但每种形式强调了同一个交互的不同方面。这两种形式是顺序图和协作图。顺序图说明了激励的精确顺序，更适用于实时的详细说明和复杂的场景。协作图说明了实例之间的关系，更适用于理解对给定实例的所有影响和过程设计。第八部分详细地描述了协作图。应当与本部分一同阅读那一部分，因为它们之间有很多共同之处。

顺序图表示按时间顺序排列的交互。特别是，它利用实例的生命线表示参与交互的实例，以及实例交换的按时间顺序排列的激励，它不表示对象之间的关联。

顺序图表示带有叠加交互的协作。协作定义了一组对于某些特定的目的有意义的参加者和关系。对参加者及其关系的识别不具有全局性。这些参加者定义了实例在相互交互时扮演的角色。因此，协作说明了一组类目角色和关联角色。遵循（或绑定）某个类目的实例扮演由类目定义的角色，同时两个实例之间的链符合协作中的关联角色。类目角色（关联角色）定义了一个实例（链）的用法，类目（关联）定义了实例（链）的所有性质。（看 63 节“协作”）

在协作的上下文中定义交互。交互说明了角色之间通讯的模式。更精确的说，它包含了一组部分有序的消息，每个消息说明了一次通讯（也即，发送什么信号和调用什么操作）以及发送者和接收者分别扮演的角色。

在形式上顺序图可以有些轻微差别，分别用于不同的目的，例如关注执行控制或并发。顺序图可以表现为一般形式（描述了所有可能序列）和实例形式（描述一个符合一般形式的实际的序列）。在没有循环和分支的情况下，这两种形式是同构的。

在下文中使用术语对象，但可用任意种类的实例代替它。

8.2 顺序图

8.2.1 语义

一个顺序图表达一个交互，一个交互是在一个协作中的类目角色之间的一组消息，以实现所期望的操作或达到某种结果。

8.2.2 表示法

顺序图是二维的，其中：1) 垂直方向表示时间，2) 水平方向表示不同的对象。

正常地，时间维由上到下（根据需要，也可以由下到上）。通常只有时间顺序是重要的，但在实时应用中时间轴是能度量的。对象的水平顺序并不重要。在图中可以把对象分组，形成“泳道”。（下面描述顺序图的详细内容）

8.5 中“消息和激励”描述了用在顺序图中的不同种类的箭头。顺序图中的箭头与协作图中的是相同的，这些在 9.2 “协作图”中叙述。

注意，该表示法的很多的内容直接来自 Buschmann、Meunier、Rohnert、Sommerlad 和 Stal 的对象消息顺序图表示法，它本身是由对象消息顺序图表示法派生的，但经过了修改。

8.2.3 表示选项

生命线的水平顺序是任意的。通常调用箭头在一页中被安排在同一方向，但这也经常是不可能的，并且这种顺序并不表达信息。

可以交换轴，所以时间维可以水平向右进行而不同的对象用水平的线表示。

可以把各种标签（例如，计时约束，对活动中的行为描述等）显示在页的边缘或在它们标记的转换或激励的旁边。

可以在消息名上用时间表达式表达计时约束。可以把函数 `sendTime`（对象发送消息的时间）和函数 `receiveTime`（对象接收消息的时间）应用到消息名上，以计算时间。时间函数集是可扩充的，因此用户可根据需要发明新的函数，以用于特定的情境或实现特性（如 `elapsedTime`、`executionStartTime`、`queuedTime` 和 `handledTime`）。

蓝图图中的一些构造标记能用于指示时间间隔，且约束可以附在其上（如下图的右下部）。这样的表示法在视觉上是吸引人的，但如果消息线是水平的，就会有模糊性，因为发送时间和接收时间不能得到区分。在很多情况下，转换时间是可以忽略的，因此这种模糊性是无害的，然而在把信息放在语义模型中之前，工具必须把这样的表示法无歧义地映射到消息名上的表达式。（工具可以接纳这种映射的缺陷。）类似地，工具可以准许省略时间函数，用消息名指示在计时表达式中的发送或接收消息的时间（如下图中的“`b.receiveTime - a.sendTime < 1 sec.`”），但再次强调，这仅是必须映射到语义模型中的正确时间表达式的表面表达式。

8.2.4 例子

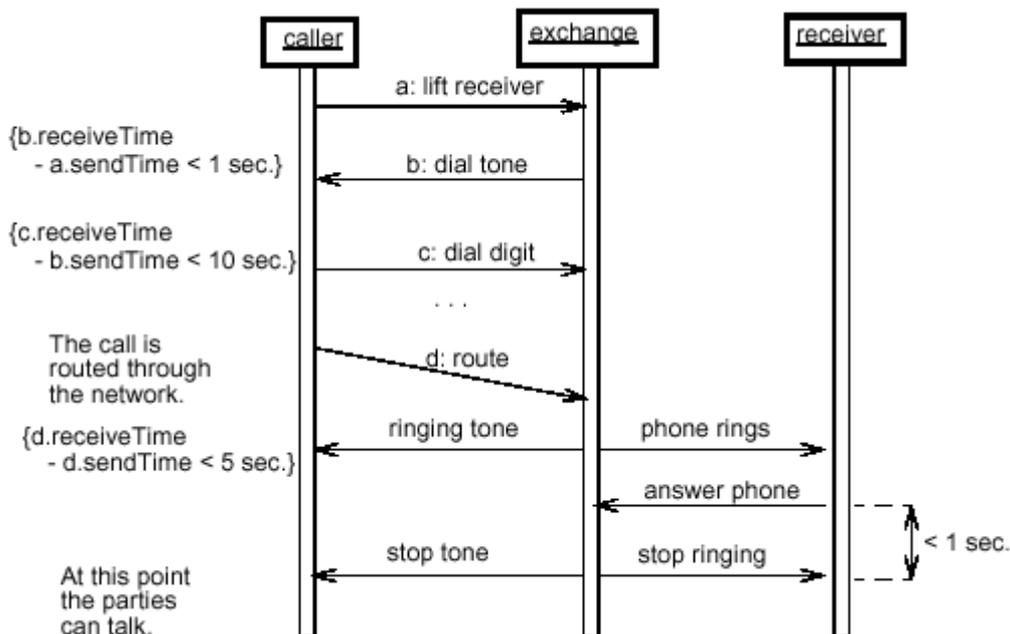


图 8-1 带有并发对象（对象表示符带有黑框）的简单顺序图

8.2.5 映射

这部分对顺序图以及在顺序图中的元素进行了概括，有些在后面进行讨论。

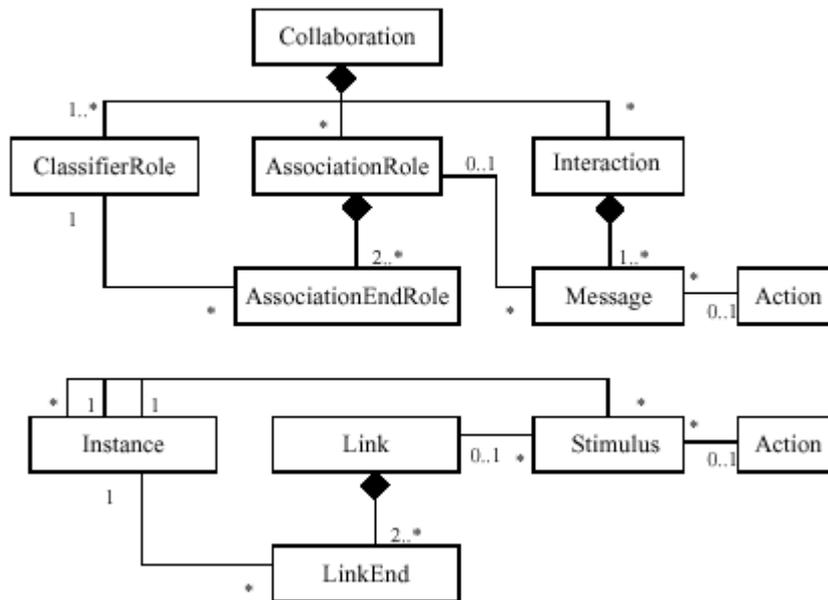


图 8-2 对用在如下部分的本部分的结构总结

8.2.5.1 顺序图

交互图映射为交互和基础协作。交互描述了一系列通讯；它含有一组部分有序的消息，每一个消息指定了发送角色和接收角色间的一个通讯。遵循拥有交互的协作中的类目的对象集通过发送遵循交互中的消息的激励进行通讯。交互图表示一组对象符号，以及映射为遵循在交互和其协作中的类目角色和消息的对象和激励的箭头。

在顺序图中，每个带有生命线的对象框映射为协作中的遵循类目角色的对象。名称域映射为对象的名称，角色名映射为类目角色名，类域映射为是类目角色的基类目的类目名。在顺序图中不显示角色间的关联。必须从补充的协作图或通过其它方式在模型中获得它们。消息箭头映射为与发送对象和接收对象连接的激励。激励遵循对应着箭头连接的两个对象生命线的类目角色之间的消息。用于激励的通讯的链扮演由将消息连接起来的关联角色指定的角色。除非能从补充的协作图或通过其它方式决定正确的链，否则由于缺乏完整信息，激励是不依附链的（模型不完整），或它依附任意的链或实例间虚设链，这样的实例遵循由两个类目角色暗示的关联。被调用的操作名或被发送的信号名映射为由连接消息的动作联系的操作或信号的名称。对显示激励的参数有不同的选择。如果显示被作为参数传递的对实际实例的引用，就把它映射为激励的参数。如果显示参数表达式，就把它映射为与分发动作相关的动作参数。最后，如果参数的类型和操作或信号的名称一起显示，就把它分别映射为操作的参数类型或信号的属性类型。在箭头尾端的定时标记映射为相应消息的名称。在图中的约束映射为整个交互上的约束。

在图的框架内指向对象符号的箭头映射为由创建动作分发的激励（即激励遵循与创建动作相关的交互内的消息）。解释是由分发激励创建对象，对象遵循由消息指定的接收角色。在创建对象后，它可以立即和其它对象交互。这暗示着对象的创建方法（构造器和初始程序）分法这些激励。如果对象终结符号（“X”）是箭头的目标，那么箭头映射为引起接收对象被转移的激励。激励遵循在具有附加到消息的销毁动作的交互图中的消息。如果对象终结符号出现在没有进入箭头的图中，它映射为终结动作。

在图中的箭头次序映射为消息间的一对关联，它对应着箭头映射在其上的激励。以垂直顺序，在对应连续箭头的消息间建立前趋关联。在当前箭头领先一个箭头的情况下，相应的消息有前趋集。此外，每个消息都有与激活进入箭头相应的消息关联的激励器。

过程性顺序图

在过程性顺序图中（带有控制中心和调用），在相同生命线上的顺序箭头映射为服从相应消息间的前驱关联的激励。到控制中心的区域头部的箭头建立嵌套激活。箭头映射为遵从带有分发动作（调用动作）的消息的激励。激励拥有发送者对象和接收者对象，以及为调用和对要被调用的目标操作的引用提供的参数对象。对操作的参数求值的表达式是在连接到消息的调用动作上的参数表达式，而发送者和接收者对象角色由消息的发送者和接收者类目角色描述。发送者和接收者对象遵从这些类目角色。依附到箭头的全部监护条件和迭代条件成为依附到消息的动作的再现值。所有离开嵌套激活的箭头映射为带有激活关联的消息，该激活关联连接到相应激活箭头头部的消息。从激活尾部离开的返回箭头映射到遵从具有如下性质的激活：

- a) 到对应激活箭头头部的消息的激活关联；
- b) 到同一个激活内的前一个消息的前趋关联（也即，在激活中被发送的最后一条消息），在前趋链中返回的必须是最后的消息。它不是任何消息的前趋。

8.3 对象生命线

8.3.1 语义

在顺序图中，对象生命线表示扮演特定角色的对象。在生命线之间的箭头表示扮演这些角色的对象之间的通讯。在顺序图中表示处于一个角色状态的对象的存在和持续，但没有表示对象之间的关系。类目角色规定这样的角色；它描述扮演该角色的对象的性质，并描述处于该角色的对象和其他对象之间的关系。

8.3.2 表示法

把对象表示成称之为“生命线”的垂直虚线，生命线代表一个对象在特定时间的存在。如果对象在图中所示的时间段内被创建或者销毁，那么它的生命线就在适当的点开始或结束。否则，生命线应当从图的顶部一直延续到底部。在生命线的头部画对象符号。如果一个对象在图中被创建，那么把映射为创建对象的激励的箭头的头部画在对象符号上。如果对象在图中被销毁，那么用一个大的“X”标记它的析构，该标记或者放在映射为引起析构的激励的箭头处，或者放在从被销毁的对象最终返回的箭头处（在自析构的情况下）。在图的顶部（第一个箭头之上）显示在转换开始时就存在的对象，而在整个转换完成时仍然存在的对象的生命线，要延伸超出最后一个箭头。

生命线可以分裂成两条或更多条并发的生命线，以表示条件性。每一个单独的轨迹对应于通讯中的一个条件分支。生命线可以在某个后续点处合并。

8.3.3 例子

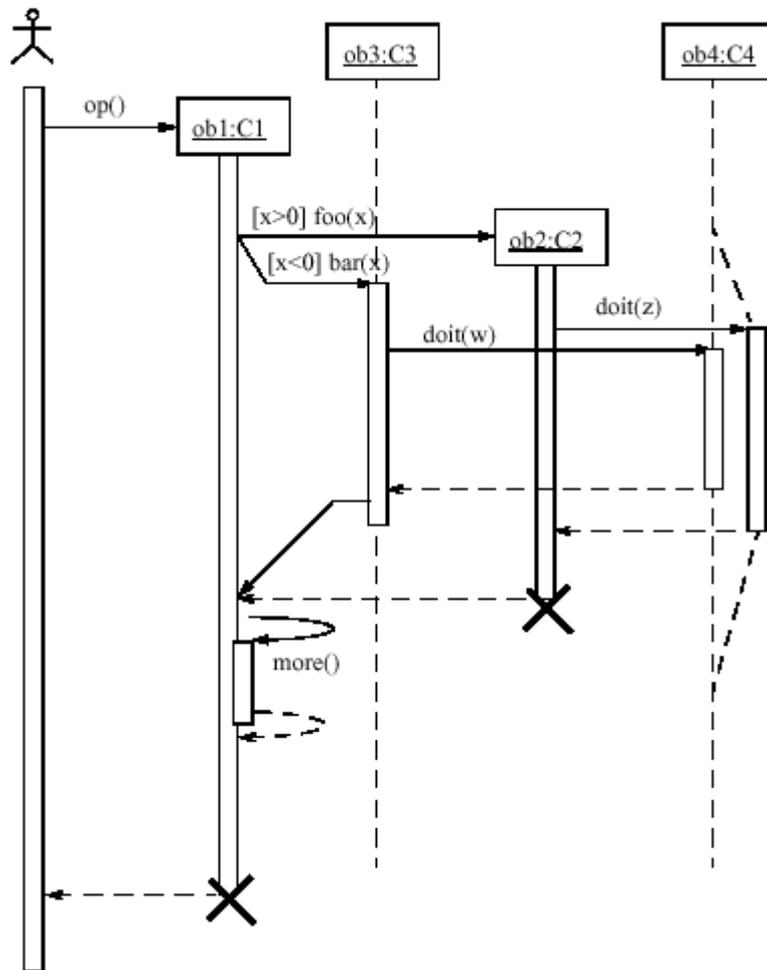


图 8-3 带有控制焦点、条件、递归、创建和析构的顺序图

8.3.4 映射

见 8.2 的“映射”。

8.4 激活

8.4.1 语义

激活（控制焦点）表示一个对象直接或者通过从属例程执行一个行为的时期。它既表示了行为执行的持续时间，也表示了活动和它的调用者之间的控制关系（栈框架）。

8.4.2 表示法

用一个窄长的矩形表示激活，矩形顶端和它的开始时刻对齐，末端和它的结束时刻对齐。可以用文本标注被执行的动作，依赖于整体风格，或者把标注放在激活符号的旁边，或者放在图左边的空白处。作为选择，进入的箭头可以指示动作，在这样的情况下，在激活本身上可以忽略动作。在程序的控制流中，激活符号的顶端画在进入的箭头的尖端（开始该动作的那个箭头），底端画在返回的箭头的尾部。

在每个拥有自己的控制线程的对象并发的情况下，一个激活说明了一个对象执行一个操作的持续时间。其他对象的操作并不相关。如果直接计算和间接计算（由嵌套过程执行）的区别并不重要，那么可以用整条生命线表示一个激活。

在过程性代码的情况下，一个激活表示在一个对象中一个过程是活动的或者它的从属过程（可能在其它的对象中）是活动的持续时间。换句话说，可以在一个特定的时间看到所有活动着的嵌套过程激活。在递归调用一个已激活的对象的情况下，第二个激活符号画在第一个符号稍微靠右的位置，在视觉上它们看起来像是叠起来一样（可以以任意的深度嵌套递归调用）。

8.4.3 例子

参看上两个图。

8.4.4 映射

见 8.2 的“映射”。

8.5 消息和激励

8.5.1 语义

激励是两个对象间的通讯，这样的通讯传输将产生的动作所期望的信息。一个激励会引起一个被调用的操作，产生一个信号，或者引起一个对象被创建或者被消除。

消息是激励的规约。（也即，它说明发送对象和接收对象应当遵从的角色，以及当执行时发送遵从消息的激励的动作。）

8.5.2 表示法

在顺序图中，把激励表示为从一个对象生命线到另一个对象生命线的的一个水平实线箭头。对于对象到自身的激励，箭头就从同一个对象符号开始和结束。箭头用激励（操作或信号）的名字及其参数值或者参数表达式标示。

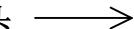
箭头也可以用一个序列数标示，以表示激励在整个交互中的顺序。由于箭头的物理位置标示了相关的顺序，所以在顺序图中一般省略序列数，但它们在协作图中是必需的。在两种图中，序列数对于标示并发控制线程很有用处。另外，还可以用监护条件标示激励。

8.5.3 表示选项

可以用如下的箭头变体表示不同种类的通讯：

实箭头 

过程调用或其它的嵌套控制流。在外层控制恢复之前，要完成整个嵌套序列。可以把它用于普通的过程调用。在某个主动对象发送信号并等待完成嵌套的行为序列时，也可以把它用于并发的主动对象。

枝状箭头 

单调的控制流。每个箭头表示按顺序进入到下一步。通常所有的消息都是异步的。

半枝状箭头 

异步激励。替代枝状箭头，用于显式地表示在过程顺序中的两个对象之间的异步通讯。

虚的枝状箭头 

从过程调用的返回。

变体：

在控制的过程流中，可以省略返回箭头（暗示激活结束），假设每个调用在任何激励后都有一个配对的返回，并可以把返回值标示在初始的箭头上。对于非过程控制流（包括并发处理和异步消息），都应当显式地标出返回。

变体：

在并发系统中，一个满箭头表示产生一个控制线程（有等待的语义），一个半箭头表示发送消息而不产生控制（有非等待的语义）。

变体：

通常消息箭头都画成水平的。这表示发送激励所需要的持续时间是“原子的”（也即，它与交互的粒度相比是短暂的，并且在传送激励的中间不能发生任何事情）。这在很多计算机中都被假设是正确的。如果激励需要一段时间到达，且这中间可能发生某些事情（例如反向的激励），那么箭头可以向下倾斜，使箭头头部在尾部下方。

变体：分支

把分支画成从一个点出发的多个箭头，每个箭头由监护条件标示。依赖监护条件是否互斥，这个结构可以表达条件或者并发。

变体：迭代

可以把一组相连的箭头闭合，表示为迭代。在一般的顺序图中，迭代表示一组激励的分发可以多次发生。在一个过程中，可以在迭代的底部标出迭代持续的条件。如果存在并发，那么图中的一些箭头可以是迭代的一部分，而另一些是独自执行的。希望对图进行布局，以使迭代箭头能容易地闭合在一起。

变体：

在表示高层视图的图中,生命线可以包含一整组对象。

变体：

对一个对象的具有有效的激活的时期和一个激活实际被计算的时期可以做一个区分。前者（在它在栈中具有控制信息而控制驻留在它调用的事物的时间段内）用普通的双线显示。后者（在它是栈中的最顶部的元素的时间段内）用阴影区显示。

8.5.3 例子

见 8.2 .5。

8.5.4 映射

见 8.2 .5。

8.6 转换时间

8.6.1 语义

消息可以指定几个不同的时间（例如，发出时间和接受时间）。这些时间可以是用在约束表达式中的正式名称。不同种类的时间集都是可以扩展的，这样用户就可以按需要为特定的目的而发明新的时间，如 `elapsedTime` 和 `startExecutionTime`。可以在约束中使用这些表达式，以为消息指派具体的有效时间约束。

8.6.2 表示法

可以赋予转换实例一个名字（例如顺序图和协作图中的激励，或状态图中的转换）。把计时约束构成为一个基于转换名称的表达式。例如，如果激励的名称是 `stim`，用 `stim.sendTime()`表示发送时间，用 `stim.receiveTime()`表达接收时间。可以把计时约束表示在与箭头对齐的图的左边上（在顺序图中），或者接近箭头的尾部（在协作图中）。在顺序图中，可以通过把布尔表达式（可能包括时间表达式）放在括号中指定约束。

8.6.3 表示选项

若在上下文很清楚，消息的名称或激励的名称本身可以用于指示转换开始的时间。在转换的执行不是瞬间的情况下，可以用带有附加撇号的名称指定转换结束的时间。

8.6.4 例子

见 8.2 的图。

8.6.5 映射

见 8.2 .5。