

面向对象分析与设计 Object-Oriented Analysis and Design

北京理工大学软件学院
马 锐
Email: mary@bit.edu.cn

第7章 交互图

- 7.1 概述
- 7.2 顺序图
- 7.3 通信图

2

7.1 概述

- 详细描述对象之间的交互，即捕获对象是怎样提供操作的，以及对象之间是如何协作的
 - ❖ 顺序图
 - ❖ 帮助分析员对照检查每个用例中描述的用户需求，是否已经落实到对象中去实现
 - ❖ 在OOD时用于描述参与者实例与界面对象的交互
 - ❖ 通信图
 - ❖ 围绕着角色与角色间关系组织的交互

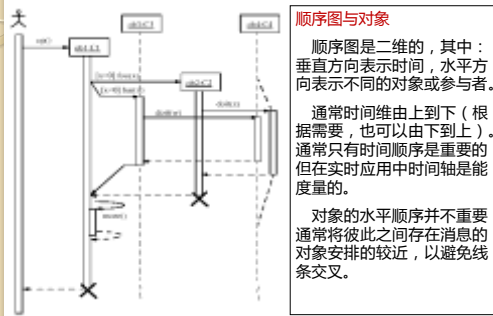
3

7.2 顺序图(1)

- 顺序图 (Sequence Diagram)
 - ❖ 是一种详细表示对象之间以及对象与参与者实例之间交互的图，它由一组协作的对象（或参与者实例）以及它们之间可发送的消息组成，它强调消息之间的顺序
- 构成
 - ❖ 对象（参与者）
 - ❖ 消息
 - ❖ 生命线
 - ❖ 执行规约

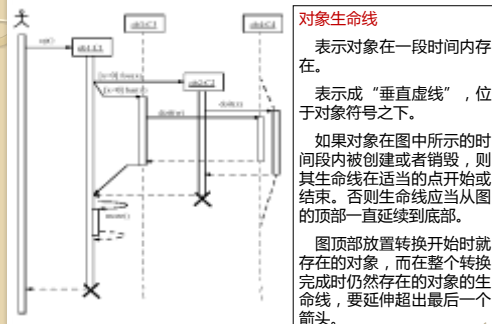
4

7.2 顺序图(2)



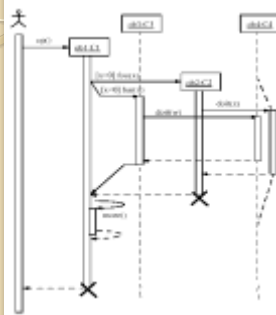
5

7.2 顺序图(3)



6

7.2 顺序图(4)



对象生命线

在生命线的顶部画对象符号

如果一个对象在图中规定的时间段被创建,把创建对象的箭头的头部画在对象符号上。

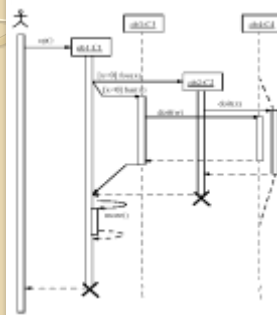
如果对象在图中被销毁,用“X”标记它的析构,该标记或者在引起析构的箭头处,或者在从被销毁对象最终返回的箭头处(在自析构的情况下)。

生命线可以分裂成两条或多条并发的生命线,表示条件性其中每一个生命线对应于通信中的一个条件分支。

生命线可在某个后续点处合并。

7

7.2 顺序图(5)



执行规约

表示一个对象执行一个操作的时期,在执行该操作时可能还调用了本操作或其他对象中的操作。

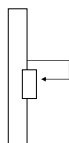
既表示了行为执行的持续时间,也表示了活动和它的调用者之间的控制关系。

用一个窄长的矩形表示执行规约,矩形顶端和它的开始时刻对齐,末端和它的结束时刻对齐。

当一个对象处于执行规约期时,该对象能够响应或发送消息,执行对象或活动。⁸

8

7.2 顺序图(6)

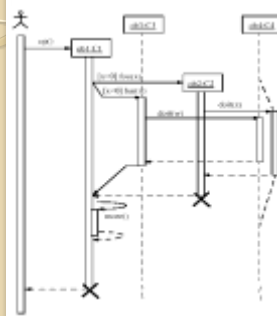


自身操作调用

在调用一个已激活的对象的一个操作的情况下,第二个执行规约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。

9

7.2 顺序图(7)



消息

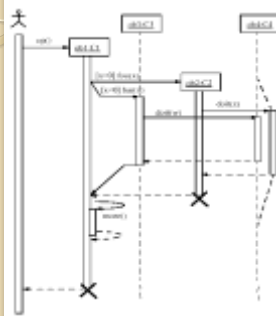
消息是对象之间的通讯的规格说明,用于传输将发生的活动所需要的信息。它既包含了控制信息(如调用)也包含了所使用的数据的规格说明。

把消息表示为从一个对象生命线到另一个对象生命线的一个水平实线箭头,即从源对象指向目标对象,以触发目标对象中的特定操作。对于对象到自身的消息,箭头就从同一个对象符号开始和结束。

用消息(操作或信号)的名字及其参数值或者参数表达式标示箭头。

10

7.2 顺序图(8)



同步消息

一般把它用于普通的过程调用。在外层控制恢复之前,要完成整个嵌套序列。

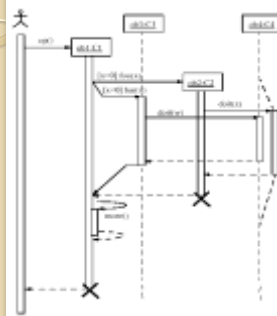
若在一个主动对象发送信号并等待完成一个嵌套的行为序列才继续时,也可以把它用于并发的主动对象。

同步消息返回

用它显式地表示从过程调用的返回。在控制的过程流中,可以省略返回箭头(暗示执行规约结束),假设每个调用在任何消息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。

11

7.2 顺序图(9)



异步消息

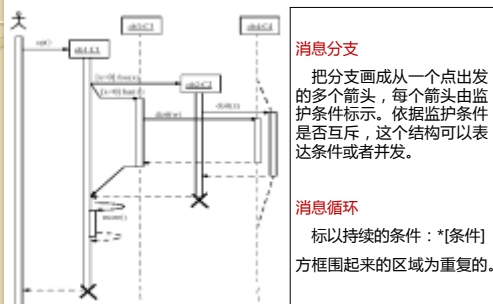
用它表示异步通讯,即发送者发出消息后,立即继续执行中的下一步,不进行等待。

异步消息返回

若请求方发了一个异步消息,且接收方响应它后要返回信息,则使用另一个异步消息。

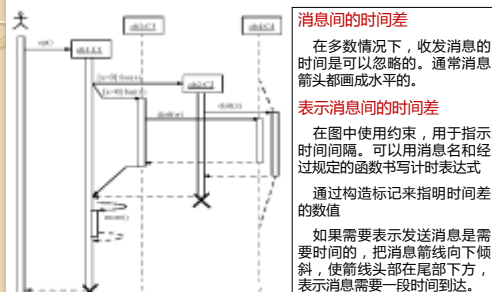
12

7.2 顺序图(10)



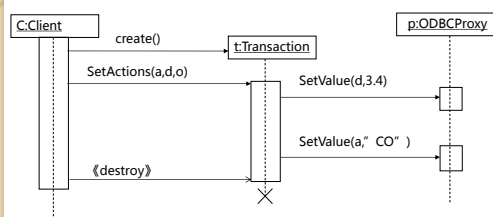
13

7.2 顺序图(11)



14

7.2 顺序图(12)



15

7.2 顺序图(13)



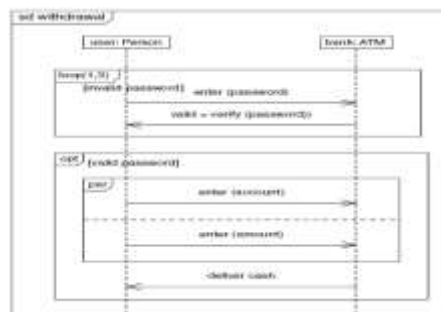
16

7.2 顺序图(14)

- 结构化控制
 - 条件消息分支、选择、循环和并行
- 顺序图边界的表示
 - 把顺序图用一个封闭的矩形包围起来，并在矩形的左上角放一个小五边形，在这个小五边形内先写上sd，再后面写出图的名字
 - 对每个子顺序图加上一个矩形区域作为外框，再在其左上角放一个小五边形，在这个小五边形内写上用来表明控制操作符的文字

17

7.2 顺序图(15)



18

7.2 顺序图(16)

➤ 可选执行

- ❖ 标签是opt
- ❖ 如果控制进入该操作符标识的交互区域时监护条件成立，那么执行该交互区域
- ❖ 监护条件是一个用方括号括起来的布尔表达式，它要出现在交互区域内部第一条生命线的顶端，在其中可以引用该对象的属性

➤ 条件执行

- ❖ 标签为 alt

19

7.2 顺序图(17)

- ❖ 用水平虚线把交互区域分割成几个分区，每个分区表示一个条件分支并有一个监护条件
- ❖ 如果一个分区的监护条件为真，执行该分区，但最多只能执行一个分区。如果有多于一个监护条件为真，那么选择哪个分区是不确定的
- ❖ 如果所有的监护条件都不为真，那么控制流将跨过这个交互区域继续执行。其中的一个分区可以使用特殊的监护条件[else]，即如果其他所有区域的监护条件都为假，就执行该分区

20

7.2 顺序图(18)

➤ 并行执行

- ❖ 标签是 par
- ❖ 用水平虚线把交互区域分割为几个分区，每个分区表示一个并发计算。当控制进入交互区域时并发执行所有分区；在并行分区都执行完后，那么该并行操作符标识的交互区域也就执行完毕。每个分区内的消息是顺序执行的
- ❖ 注意：并发并不总是意味着物理上的同时执行。并发其实是说两个动作没有协作关系，而且可按任意次序发生。如果它们确实是独立的动作，那么它们还可以交叠

21

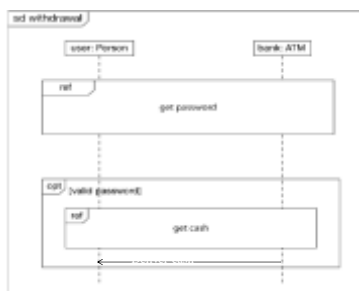
7.2 顺序图(19)

➤ 循环（迭代）执行

- ❖ 标签是 loop
- ❖ 在交互区域内的顶端给出一个监护条件。只要在每次迭代之前监护条件成立，那么循环主体就会重复执行。一旦在交互区域顶部的监护条件为假，控制就会跳出该交互区域

22

7.2 顺序图(20)



引用
 可在一张顺序图中引用另一张顺序图
 可应用于重复片段
 使用符号ref

23

7.3 通信图(1)

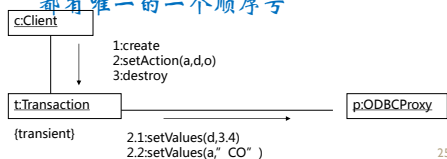
➤ 通信图

- ❖ 强调发送和接收消息的对象结构组织的图，用以展示围绕对象及其之间的连接器而组织的交互
- ❖ 明确地显示元素之间的协作关系，而不显示作为独立维的时间，消息的顺序和并发线程必须由顺序号确定
- ❖ 表示
 - ❖ 由对象（参与者）、连接器（两个对象之间的一条实线）以及连接器上的消息构成

24

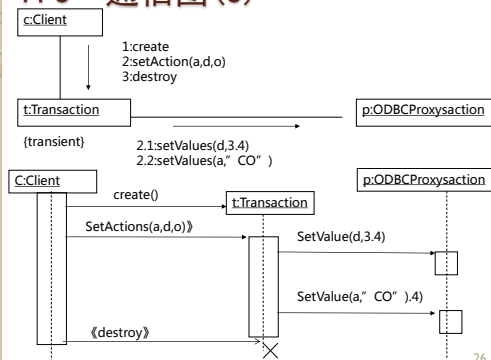
7.3 通信图(2)

- 可以给消息添加数字前缀表示消息的时间顺序，表示嵌套消息可使用带小数点的号码
- 沿同一个链，可以显示多个消息（可能发自不同的方向），并且每个消息都有唯一的一个顺序号



25

7.3 通信图(3)



26

7.3 通信图(4)

顺序图与通信图的区别

- 顺序图中不显式地展示对象之间的连接器，也不显式地展示消息的顺序号，而是将顺序号隐含在从顶部到底部消息的物理顺序中，并且对象间有消息存在隐含了其有连接器存在
- 顺序图中不能出现通信图中的 {transient} 标记
- 顺序图能描述对象生命线和执行规约

27