

绘制顺序图

1. 在顺序图顶端绘制矩形框，定义参与交互的**类实例**（对象）名；
2. 在每个对象下面绘制竖直虚线，表示该对象的**生命线**；
3. 在对象间添加箭头表示各种类型的**消息**，跟踪对象间的控制流；
4. 生命线加竖直矩形定义对象**激活期**，表明对象正在执行某操作；
5. 根据需要添加**框**的组合与关联，表示复杂的控制结构。



顺序图建模过程:

• 参与者

顺序图中有关对象或者实体

• 消息

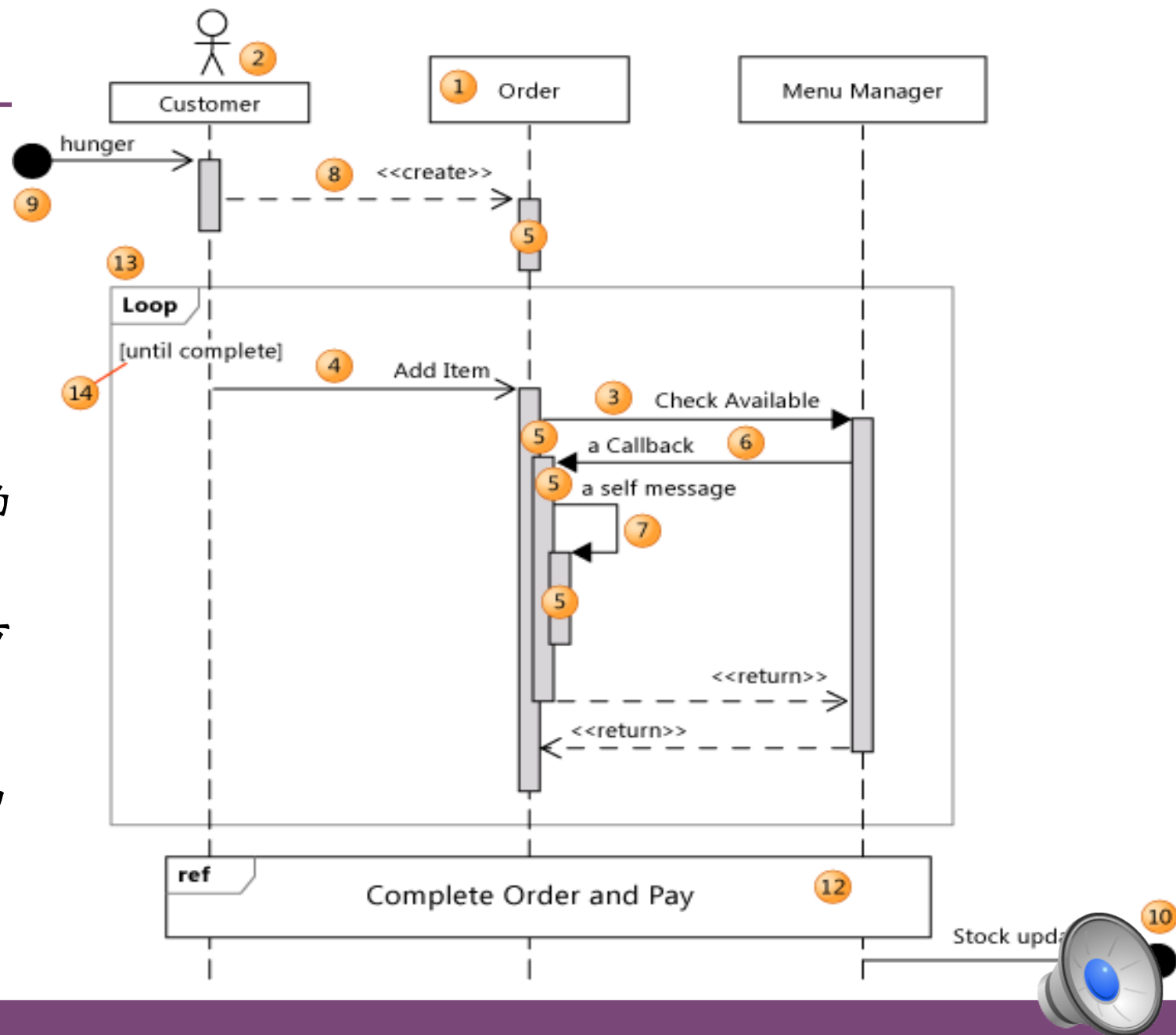
参与对象之间的通信, 通过箭头表示

- 顺序图的起始是一个没有发起对象的消息

- 每个消息代表的操作属于消息接受方

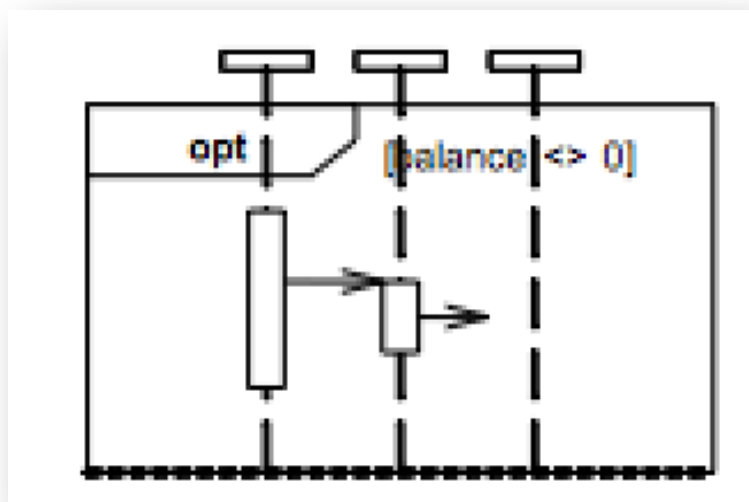
• 轴

- 横轴: 表明正在进行操作的对象/参与
- 纵轴: 时间 (向下表明时间的顺延)

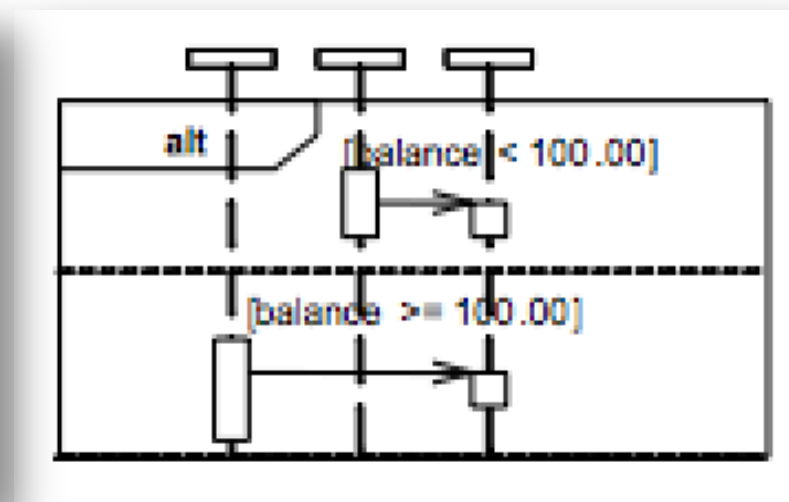


组合框：复杂控制结构表示

- **框 (Frame)**：框中包含顺序图的部分结构，表示选择(selection)或者循环(loop)结构，左上角注明结构类型，[]中注明条件。



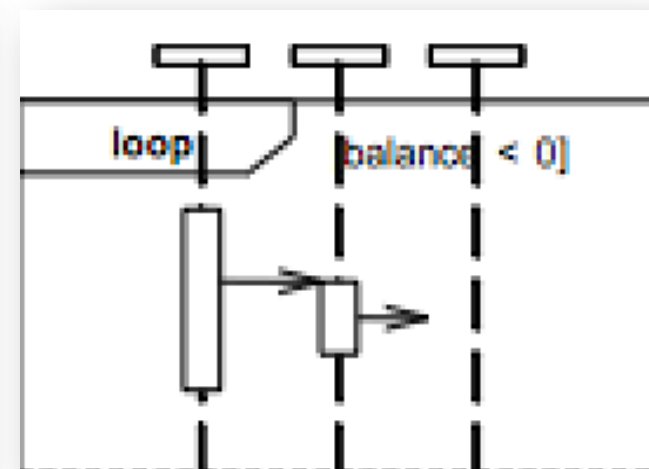
If -> (**opt**)[condition]



if/else -> (**alt**)[condition]

通过水平虚线分割不同情形

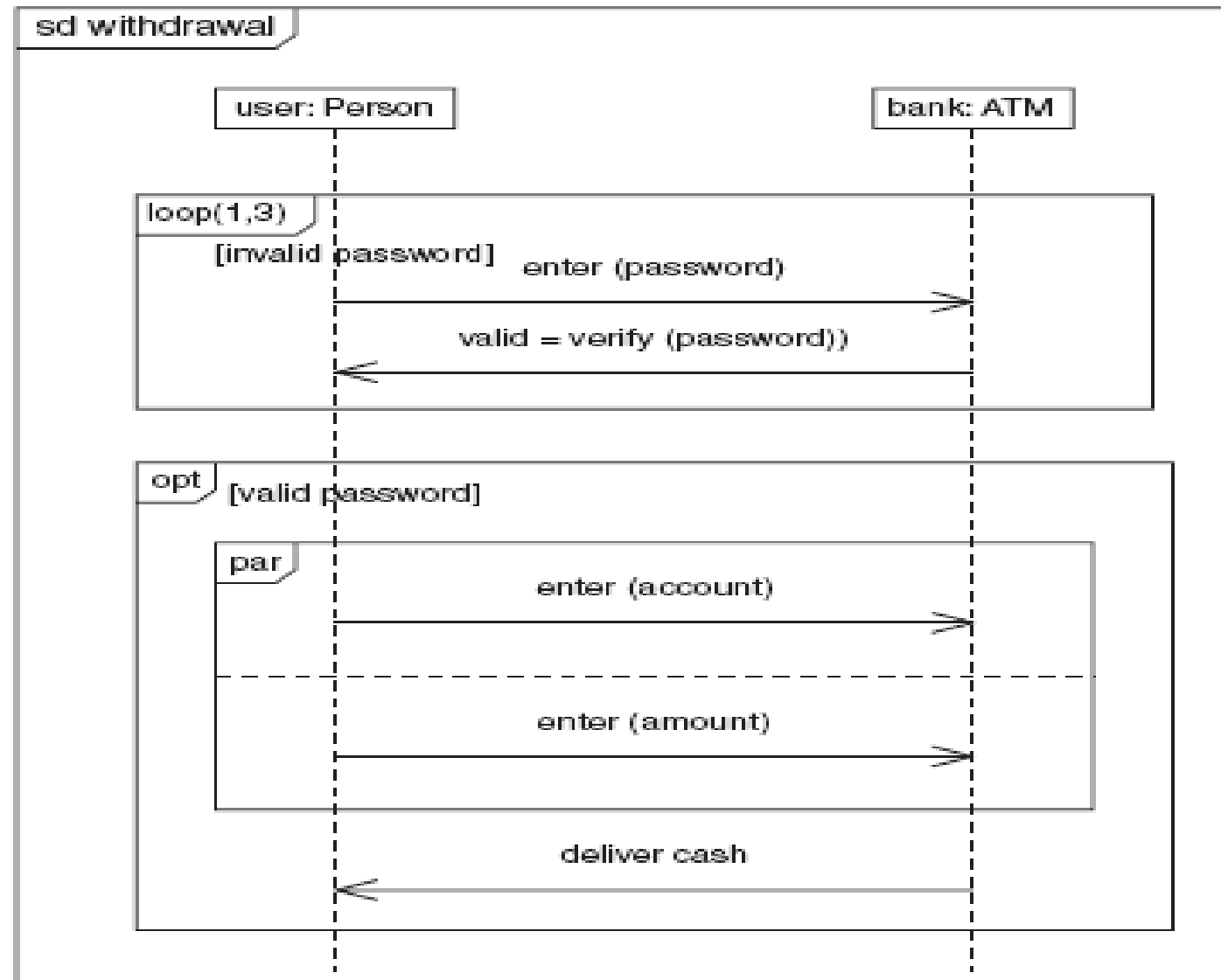
并发结构 -> (**par**)



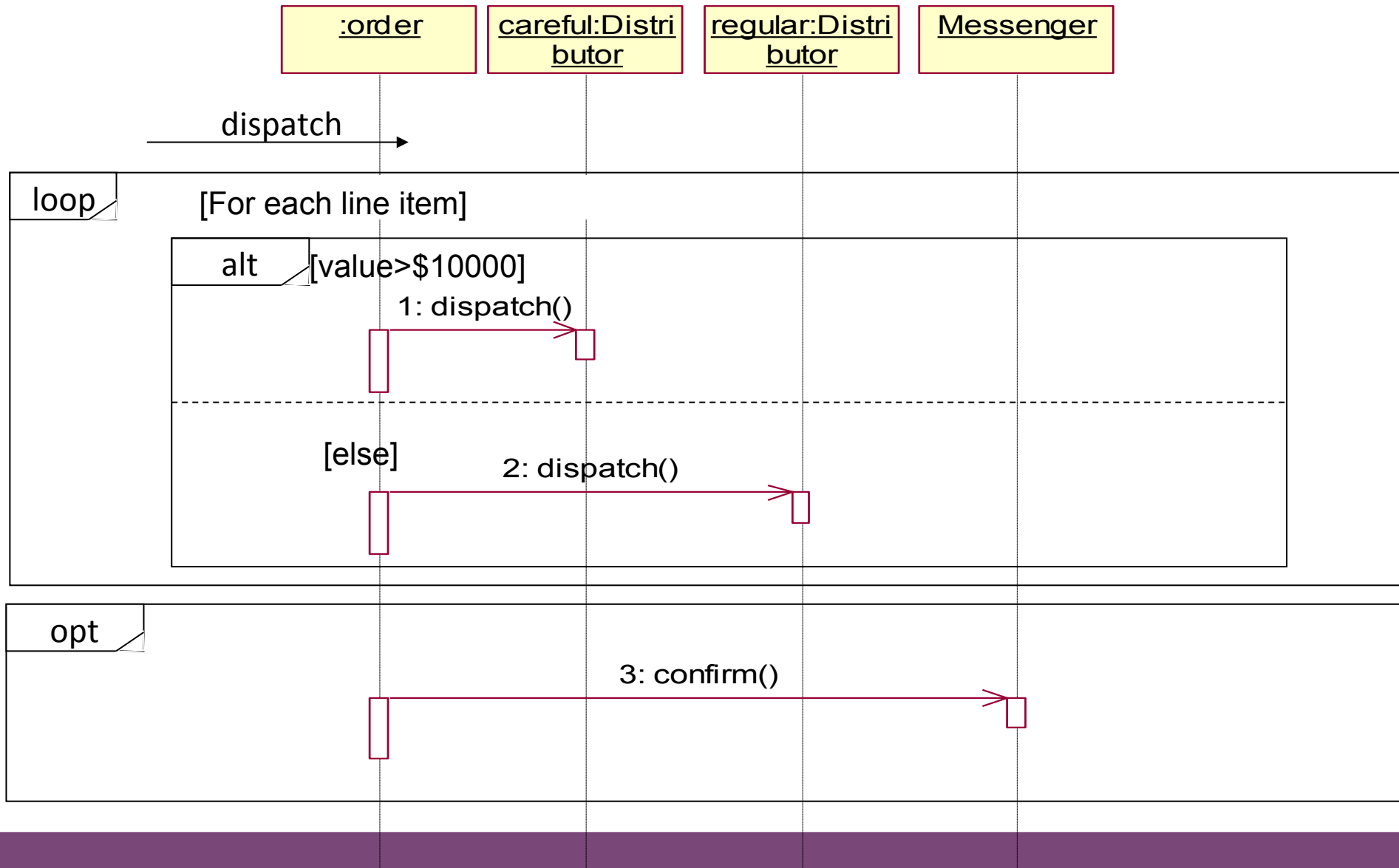
loop -> (**loop**)[condition or items to loop over]



控制框建模例子

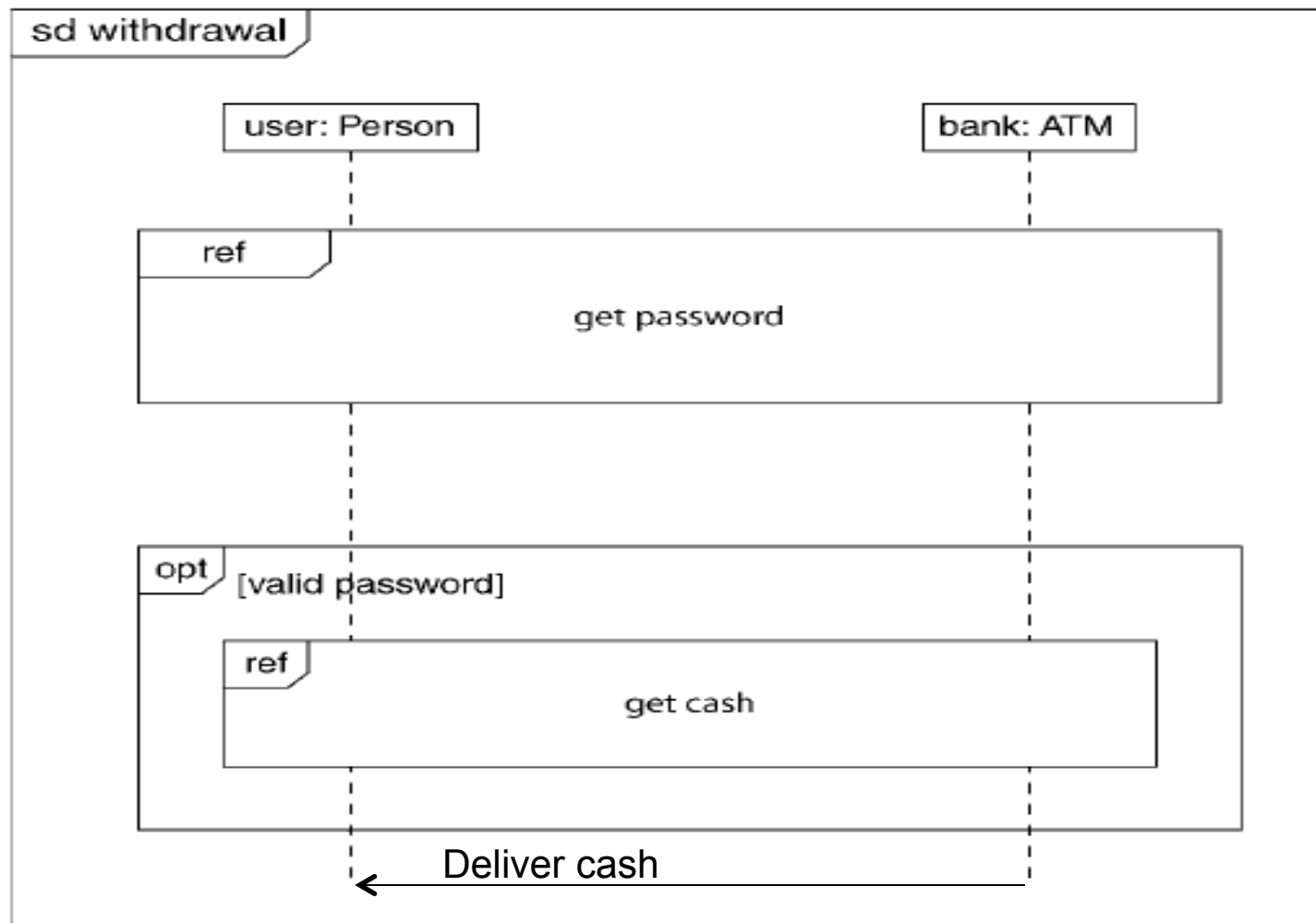


控制框建模例子



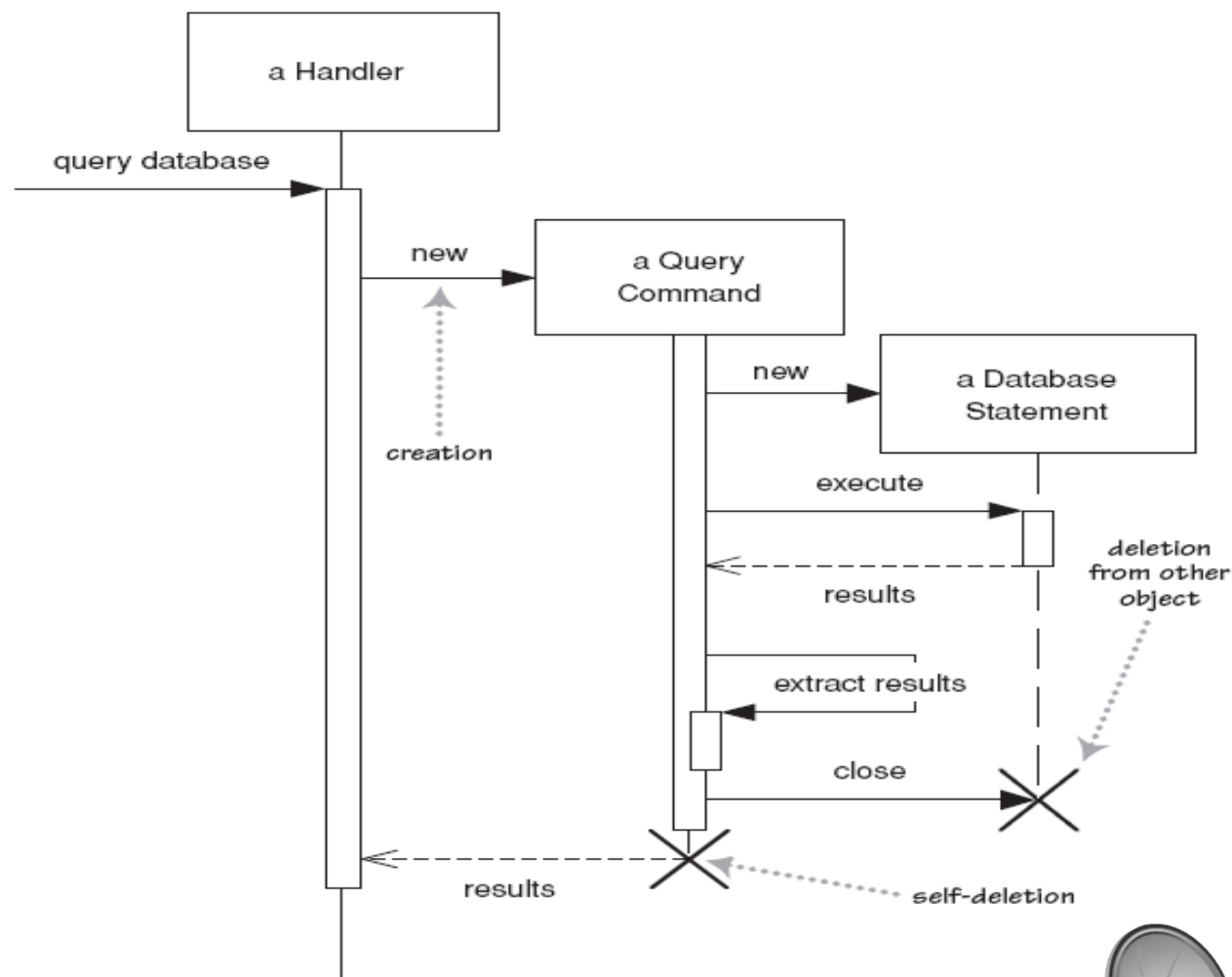
顺序图间的关联

- 当一个顺序图过大
- 需要引用其他图表时,
选择下述表示:
 - 不完整的箭头和注释
 - 通过名为“ref”的框图
引用相关图表

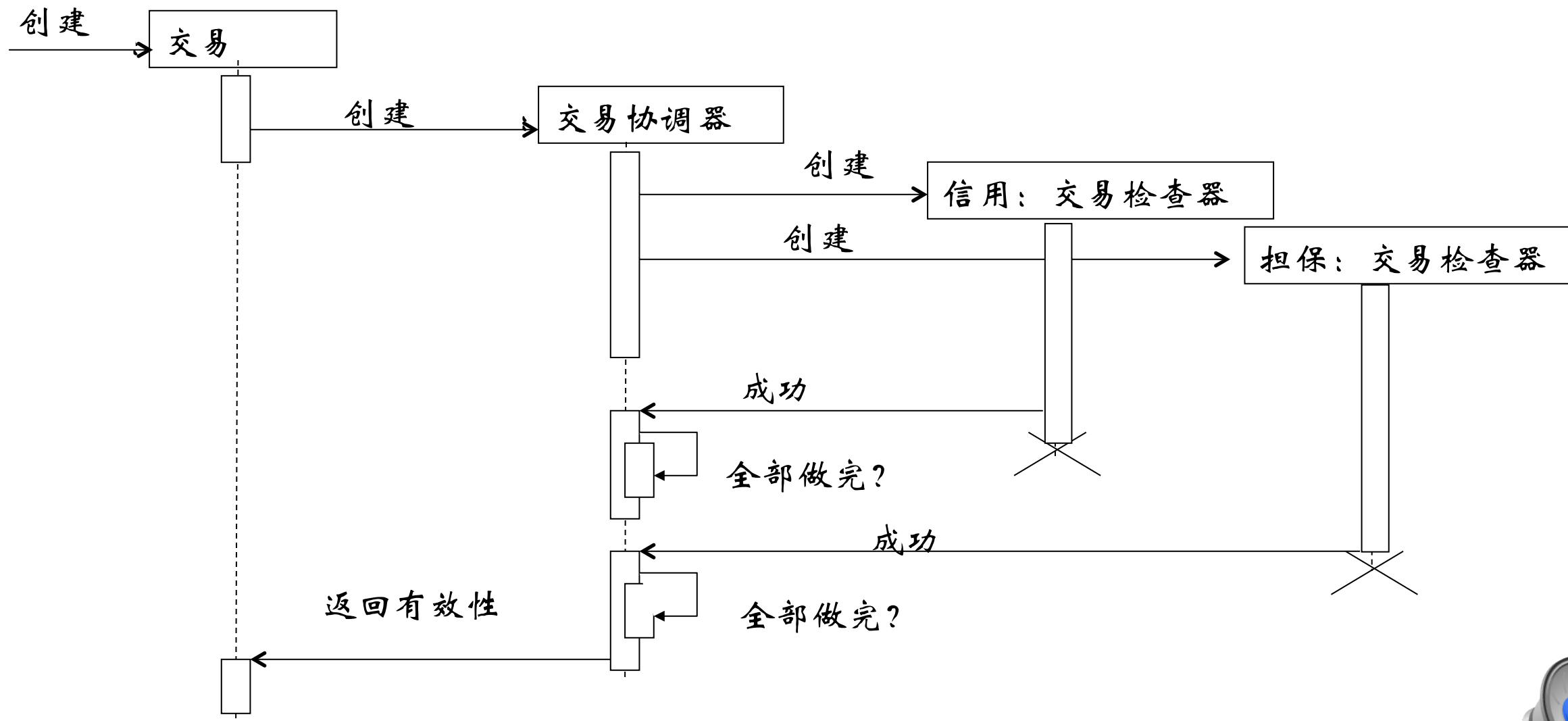


对象的创建与撤销

- 创建(creation): "new" 标明的箭头
 - 用例场景中新建的对象在图中的位置较低
- 撤销(deletion,destroy): 生命线底部的 "X"
 - 注: 在Java没有明确的销毁对象的操作, 通过垃圾回收机制处理

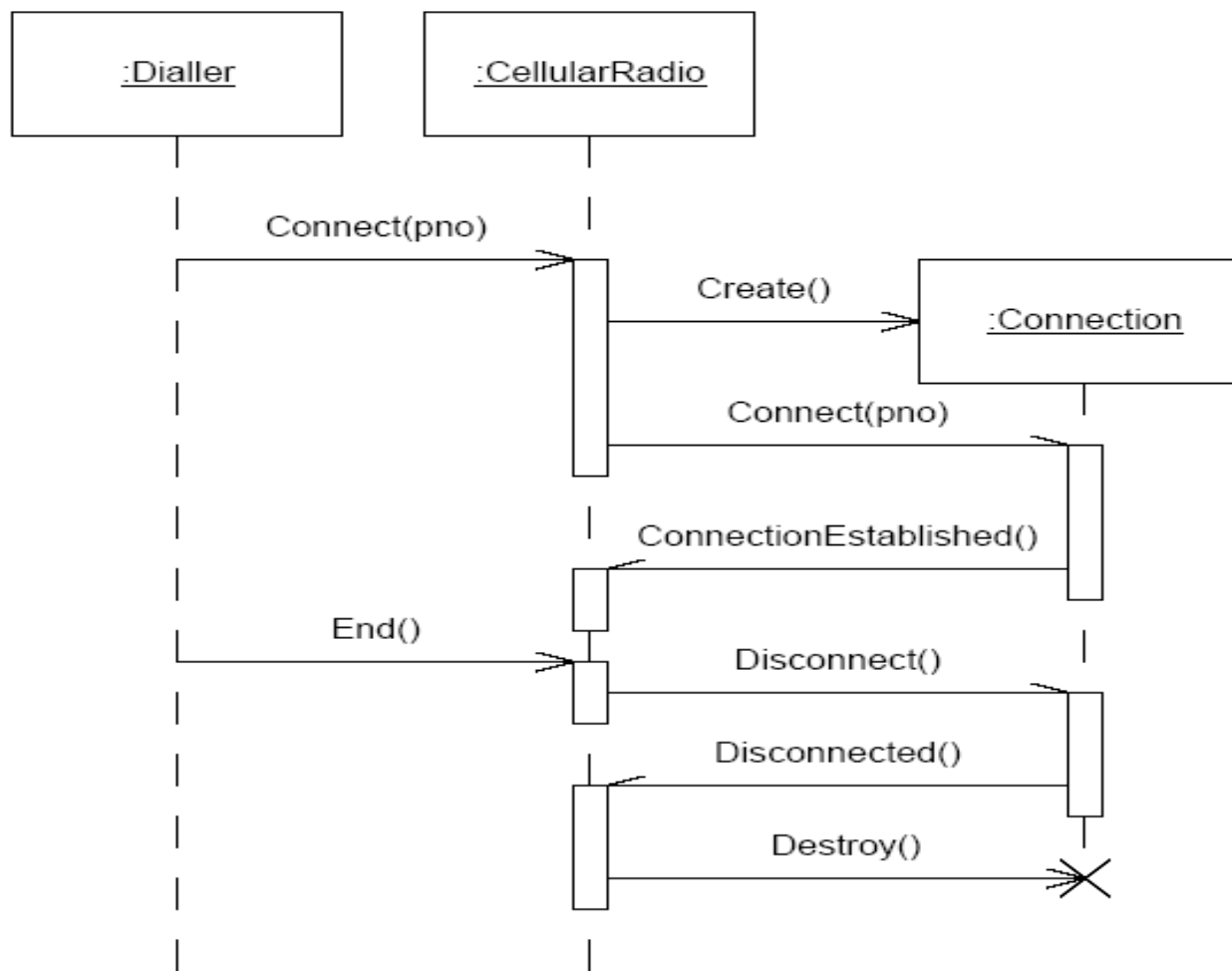


例:银行系统的交易验证

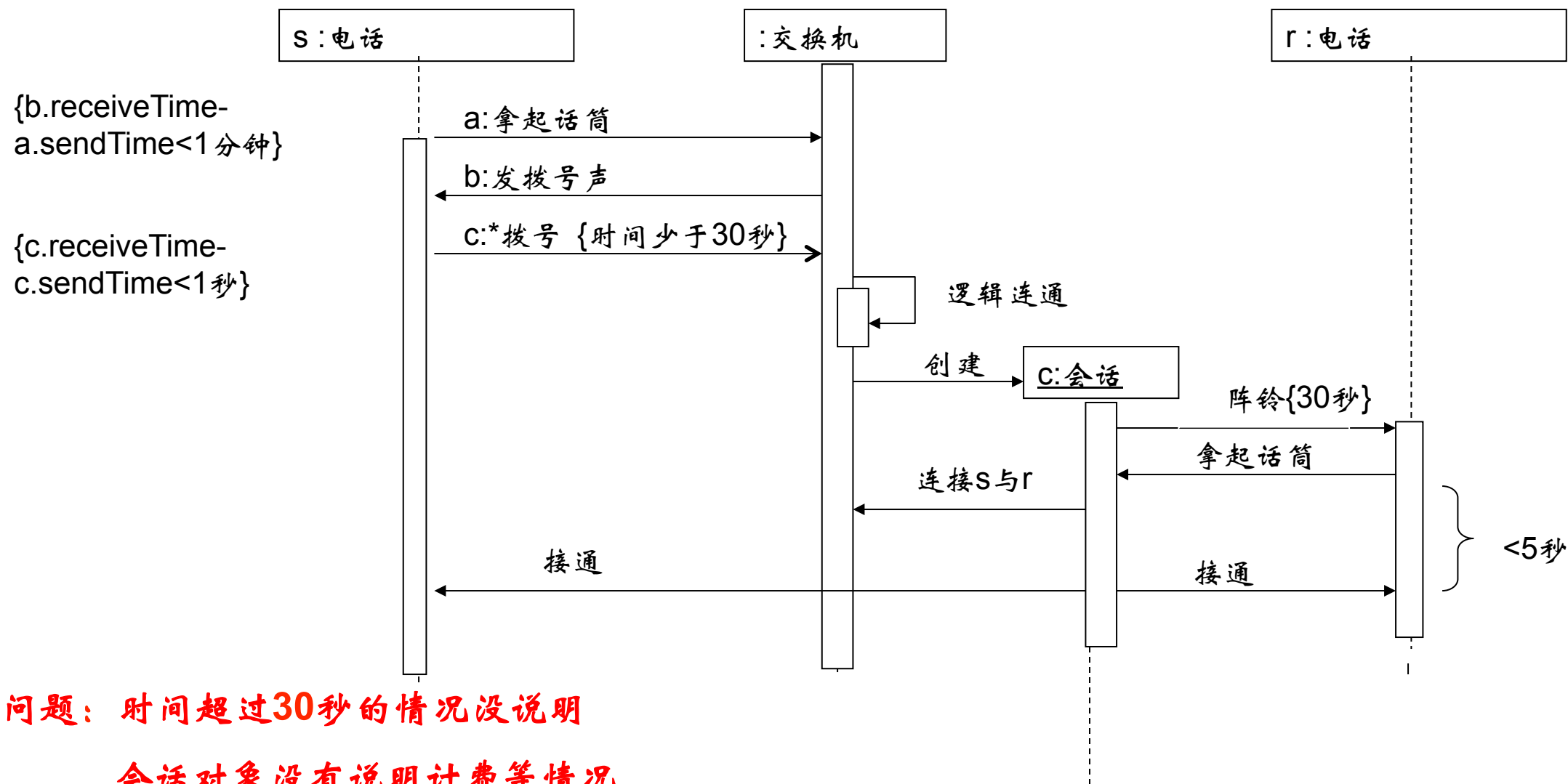


思考题：有缺陷的顺序图

- 这个顺序图有什么问题？
(留意UML的语法以及场景的可行性)

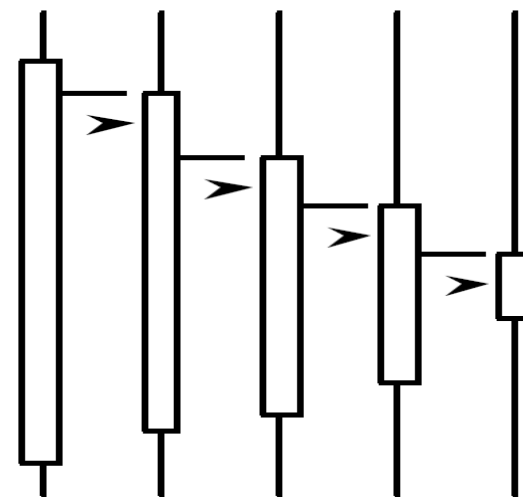
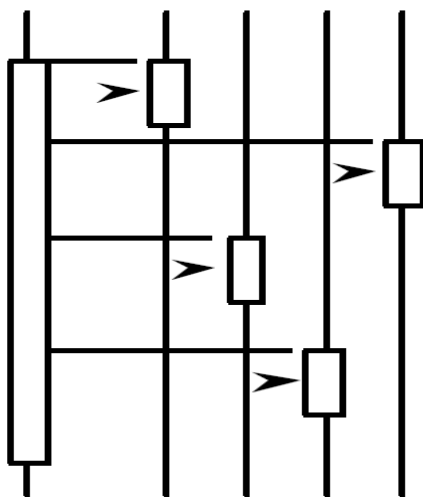


修改方案1

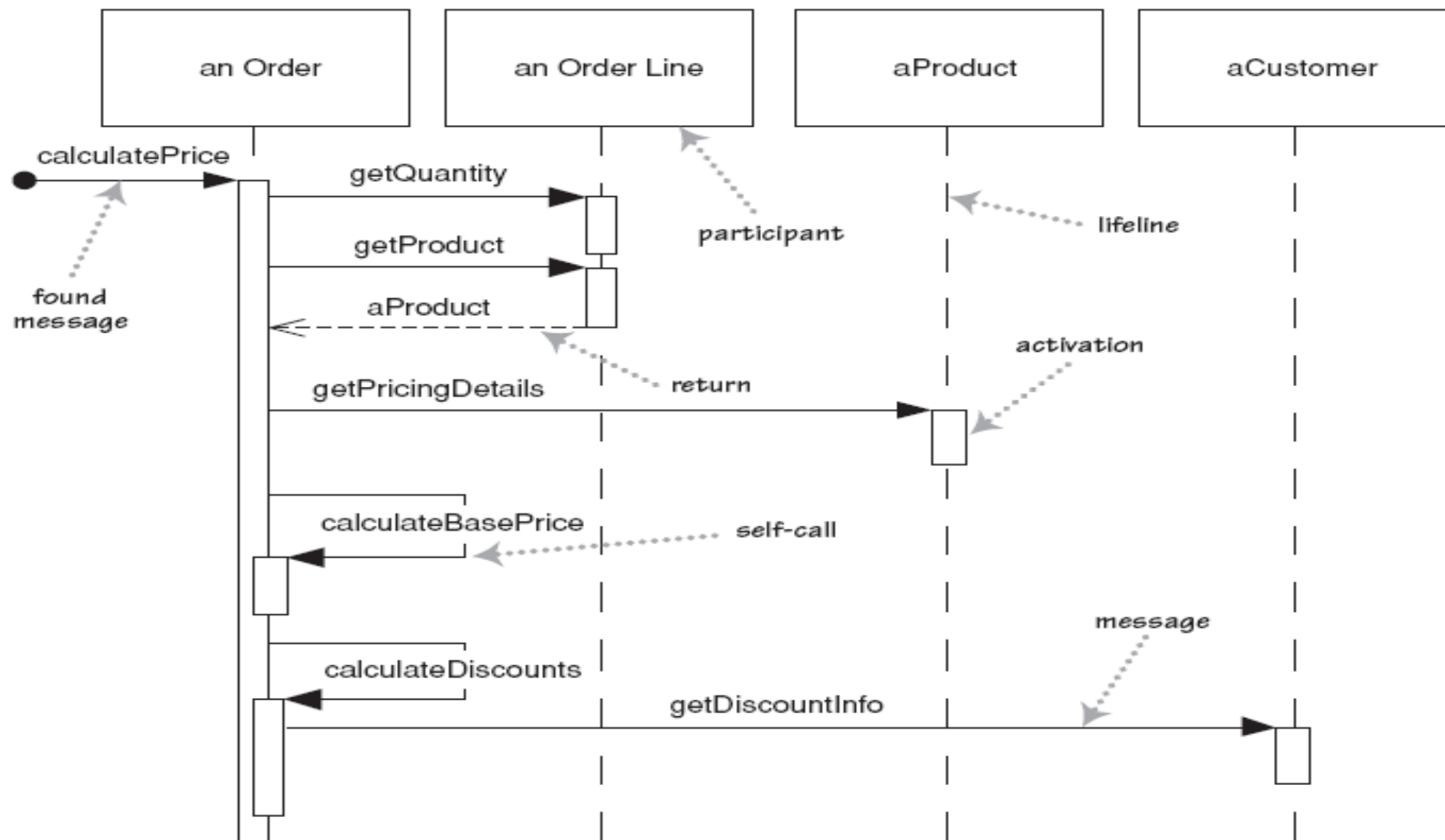


集中、分布控制

- 下述两种系统的控制流有什么特点？
 - 集中式的？
 - 分布式的？



例1：集中控制的计价系统顺序图



例2：分布控制的计价系统顺序图

