

第8章.面向对象分析与 UML建模

韩锐

北京理工大学 计算机学院

Email: 379068433@qq.com
hanrui@bit.edu.cn

面向对象分析与UML建模(4)

- 1. 动态模型
- 2. 顺序图
 - 2.1. 顺序图概念与表示
 - 2.2. 顺序图的建立
- 3. 协作图
 - 3.1. 协作图概念与表示
 - 3.2. 协作图的用途
- 4. 状态图
 - 4.1. 状态图概念与表示
 - 4.2. 状态图的建立
- 5. 活动图
 - 5.1. 活动图概念与表示
 - 5.2. 活动图应用
- 6. 建立动态模型示例

1. 动态模型

■交互图Interaction Diagram

描述对象间以及对象与参与者间动态交互过程的次序，通常对应一个用例。

包括：{ 顺序图---强调消息交互的时序；
协作图---强调对象收发消息的组织结构
(UML2.0称为通信图)

■状态图Statechart Diagram

描述一个对象在其生存期中的状态序列，表现引起状态变换的事件和状态变化的动作。

■活动图 Activity Diagram

描述系统的工作流程和并发行为。



面向对象分析与UML建模(4)

- 1. 动态模型
- 2. 顺序图
 - 2.1. 顺序图概念与表示
 - 2.2. 顺序图的建立
- 3. 协作图
 - 3.1. 协作图概念与表示
 - 3.2. 协作图的用途
- 4. 状态图
 - 4.1. 状态图概念与表示
 - 4.2. 状态图的建立
- 5. 活动图
 - 5.1. 活动图概念与表示
 - 5.2. 活动图应用
- 6. 建立动态模型示例

2.1. 顺序图概念与表示

顺序图Sequence Diagram 作用:

- 在给定的语境中，通过对象之间的消息通信，展现对象的行为；用以发现对象的操作。
- 顺序图可协助发现主动对象

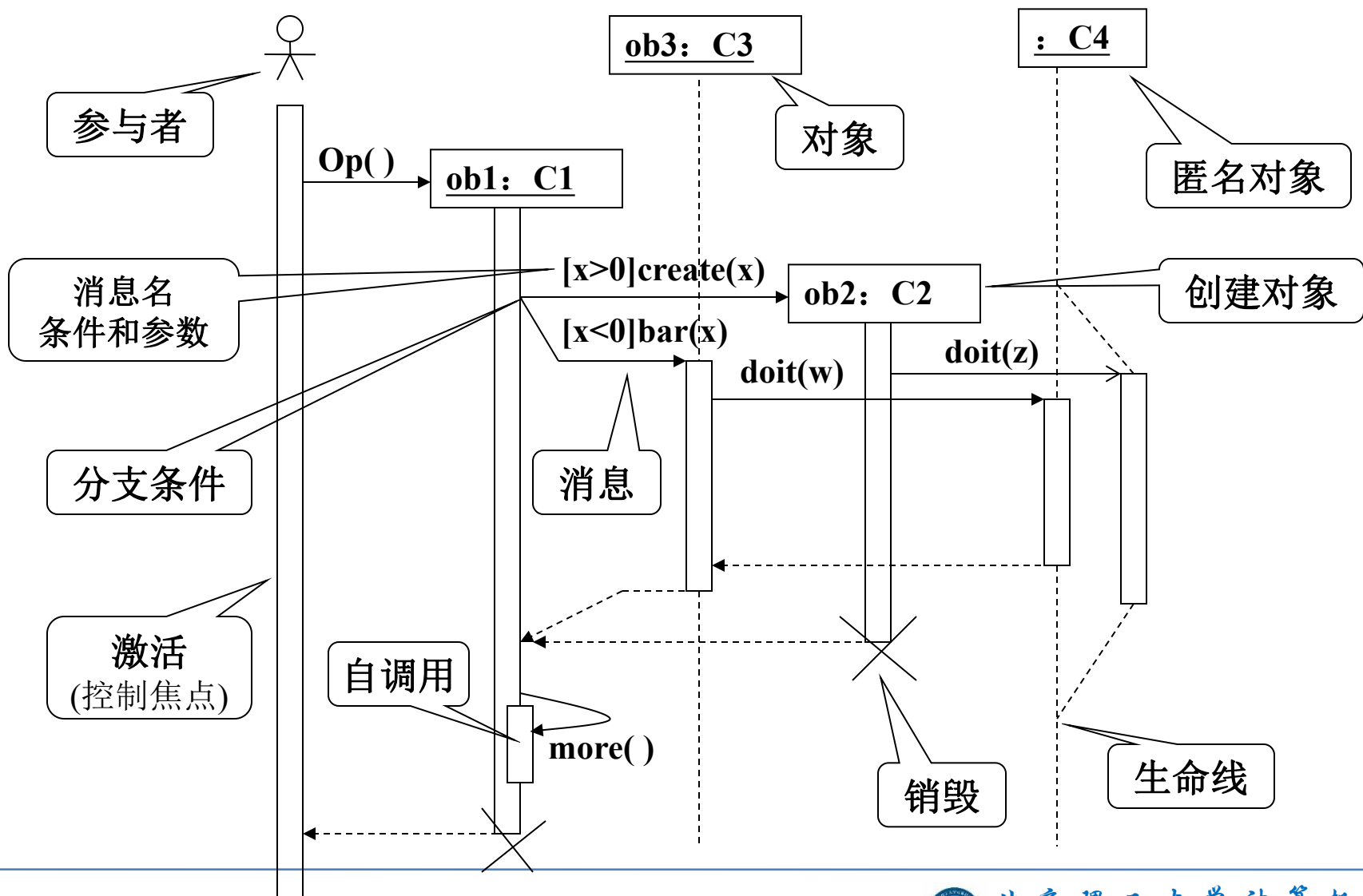
表示： 对象(与参与者)之间的交互

元素： 一组协作的对象(与参与者)和之间发送的消息

强调： 消息之间的顺序



顺序图示例



顺序图的主要元素

- **对象与角色**：最顶上一排矩形框。
- 参与交互的对象既可以是具体的事物，又可以是原型化的事物。
- 例如，aOrder作为类Order的一个实例，可以代表一个特定的订单；而如果作为一个原型化的事件，则aOrder可以代表类Order的任何一个实例。



顺序图的主要元素

- **生命线与控制焦点：**每个对象都有自己的生命线，对象生命线是一条垂直的虚线，用来表示一个对象在一段时间内存在。
- **消息：**用来描述对象之间所进行的通信的，该信息带有对将要发生的活动的期望。当传送一个消息时，它所引起的动用是一个通过对计算过程的抽象而得到的可执行语句。
- 消息分为五种：调用、返回、发送、创建和销毁
- 调用：表示调用某个对象一个操作



顺序图的主要元素: 调用

UML 2表示法:



UML 1表示法:

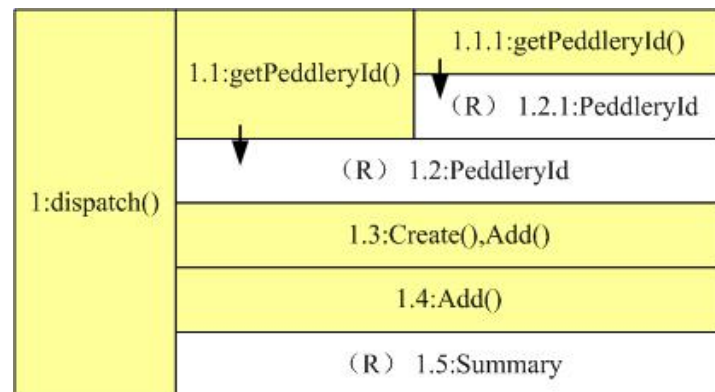
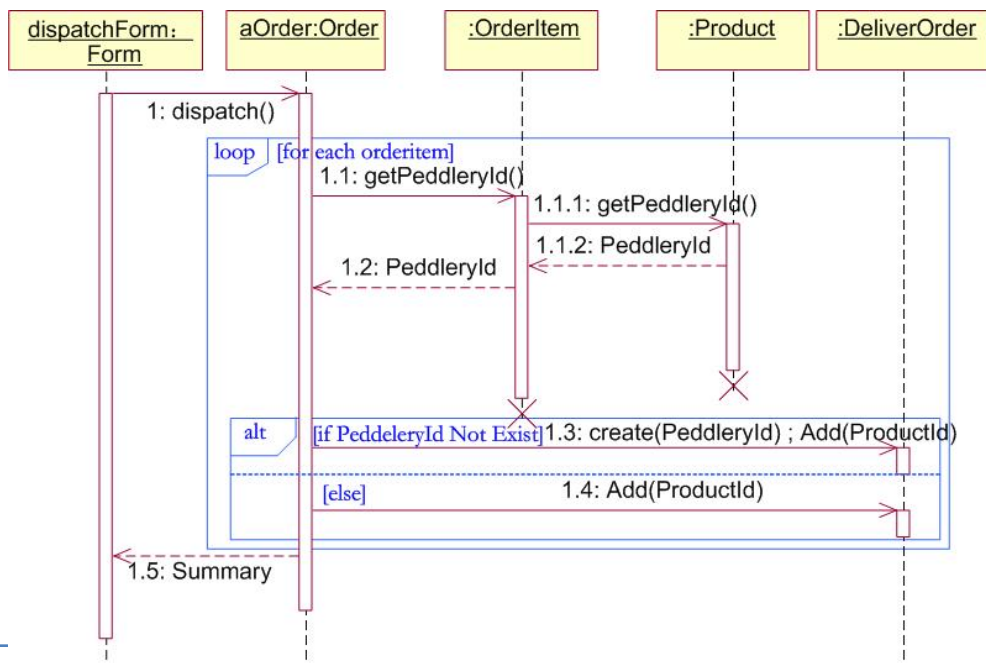


Rose表示法:



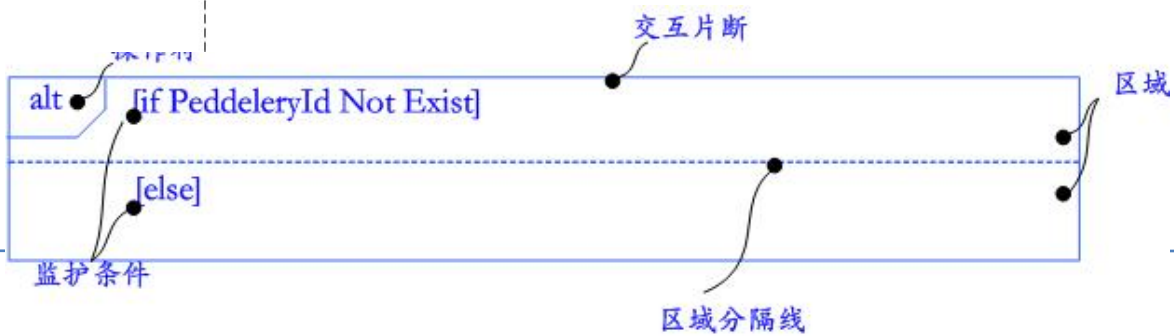
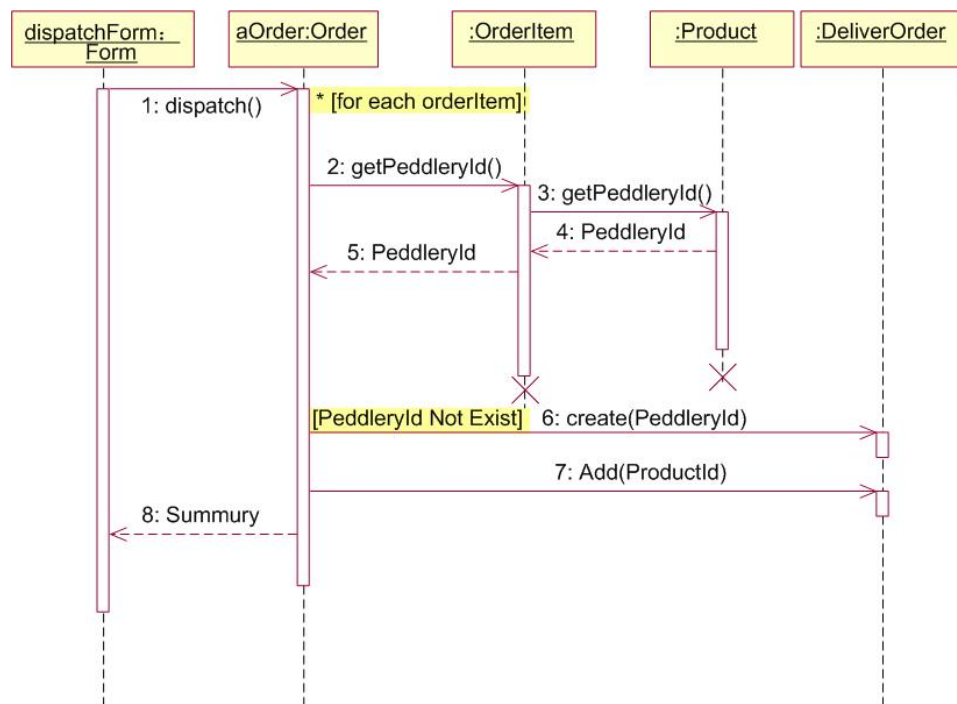
顺序图的主要元素

- **顺序编号：** 整个消息的传递过程就形成了一个完整的序列，因此通过在每个消息的前面加上一个用冒号隔开的顺序号来表示其顺序。除了顺序编号之外，还可以采用嵌套方案：



顺序图的主要元素

循环与分支



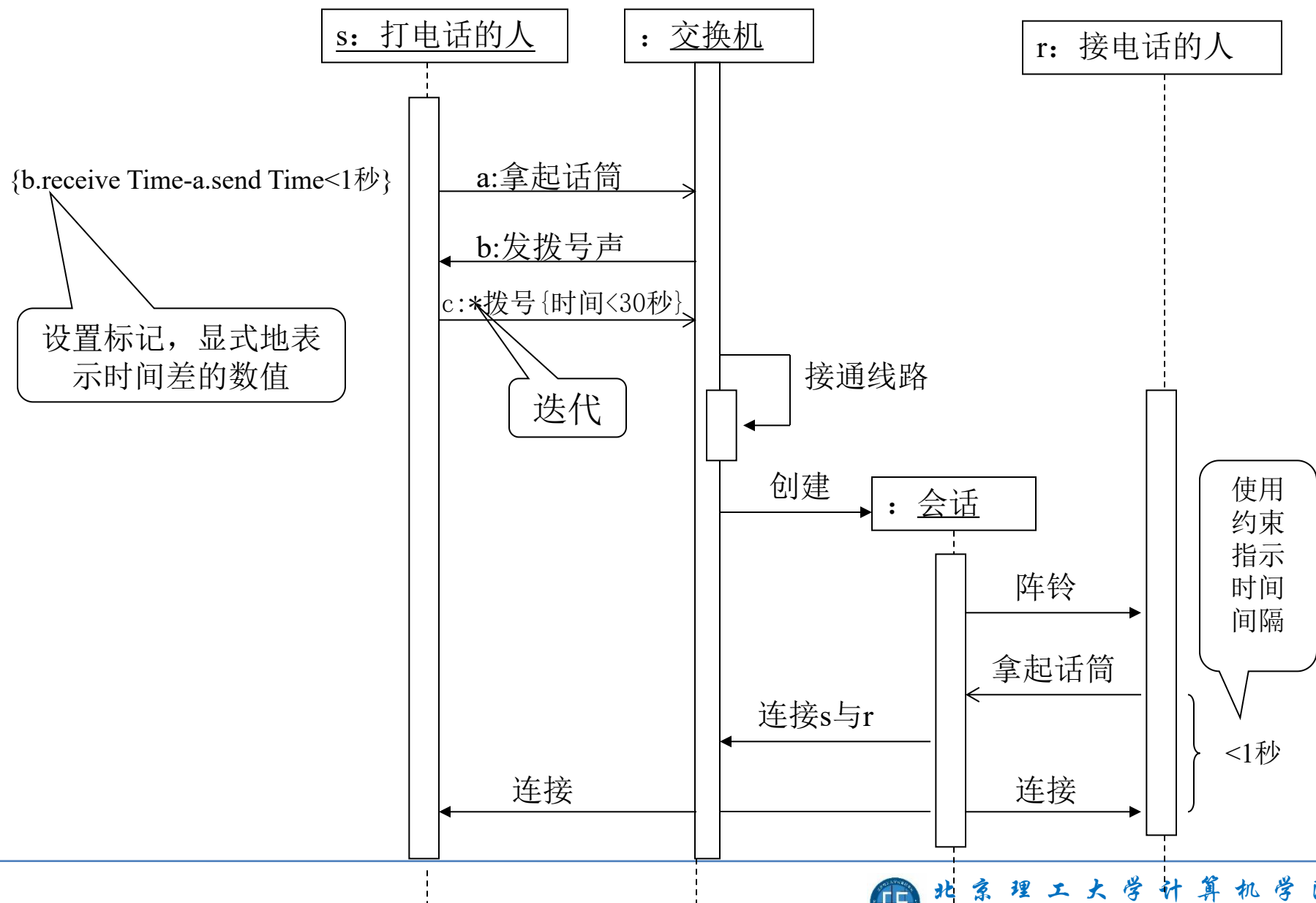
2.2. 顺序图的建立

过程和步骤:

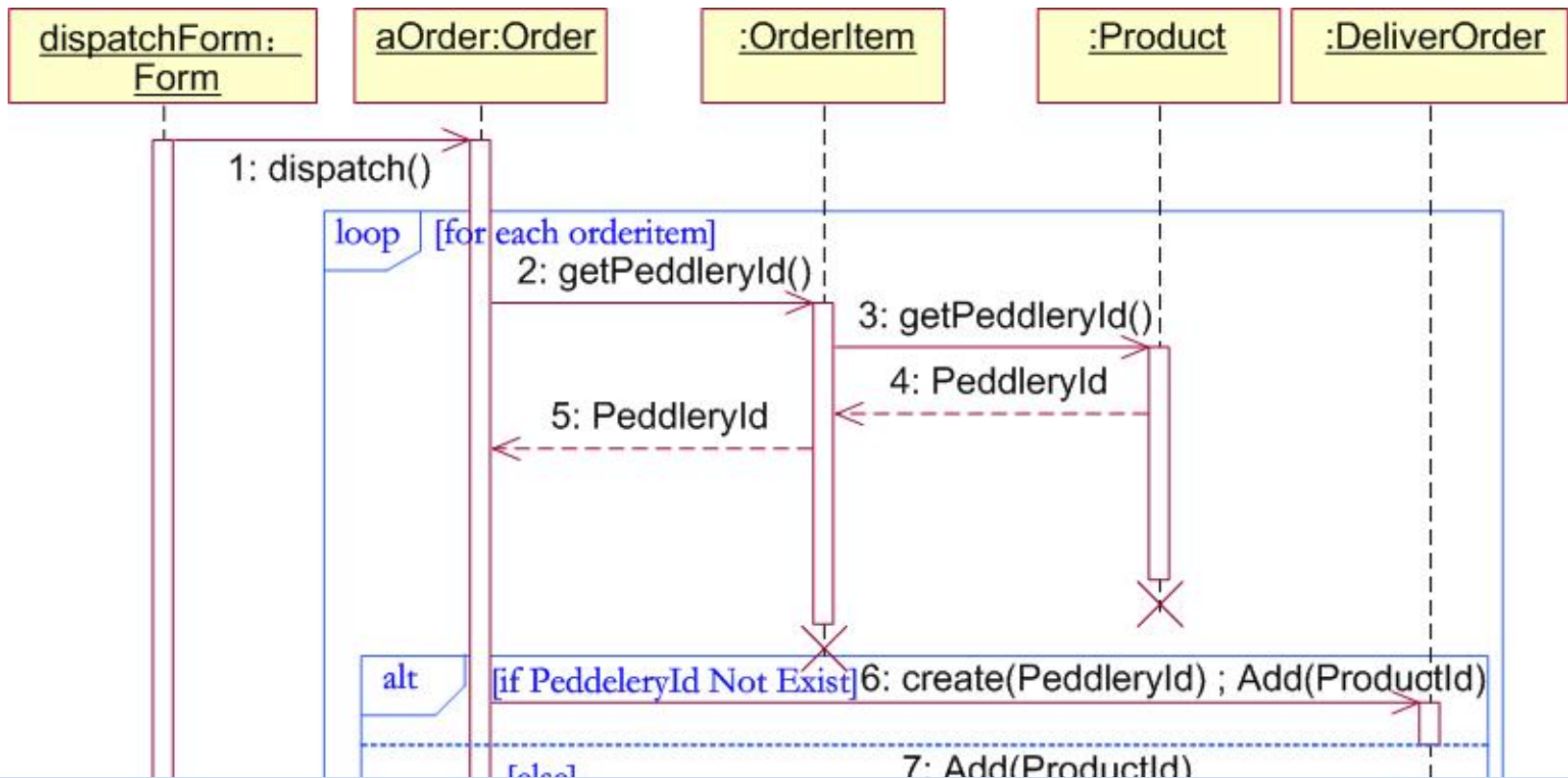
- 1) 根据具体用况中的对象或参与者的交互语境, 设置交互;
- 2) 在顺序图上部列出所选的一组对象或参与者;
- 3) 为每个对象和参与者设置生命线。
- 4) 用消息箭线显式地标出交互中将被创建和撤消的对象;
- 5) 在对象之间, 标出消息进行传递的序列;
- 6) 在对象生命线上, 按对象操作的次序, 排列各操作的激活区间, 若两个对象的操作执行属于同一控制线程, 则接收者操作的执行应在发送者发出消息之后开始, 并在发送者结束之前结束;
- 7) 描述对象执行的操作功能、时间、范围约束;
- 8) 可显式地表示消息的迭代或分支。



打电话的顺序图



顺序图示例



- 在dispatchForm（分发窗体）中，对于某个已支付的Order进行分发时，就会调用该订单
 - 当Order的实例对象aOrder得到返回的PeddlerId后，根据该值判断是否已经有相对应的DeliverOrder对象，如果没有就创建它（调用create(PeddlerId)），然后再将对应的Product添加到这个DeliverOrder对象中。否则就直接添加到相应的DeliverOrder对象中
- getPeddlerId()方法来获取供应商ID

面向对象分析与UML建模(4)

- 1. 动态模型
- 2. 顺序图
 - 2.1. 顺序图概念与表示
 - 2.2. 顺序图的建立
- 3. 协作图
 - 3.1. 协作图概念与表示
 - 3.2. 协作图的用途
- 4. 状态图
 - 4.1. 状态图概念与表示
 - 4.2. 状态图的建立
- 5. 活动图
 - 5.1. 活动图概念与表示
 - 5.2. 活动图应用
- 6. 建立动态模型示例

3.1. 协作图概念与表示

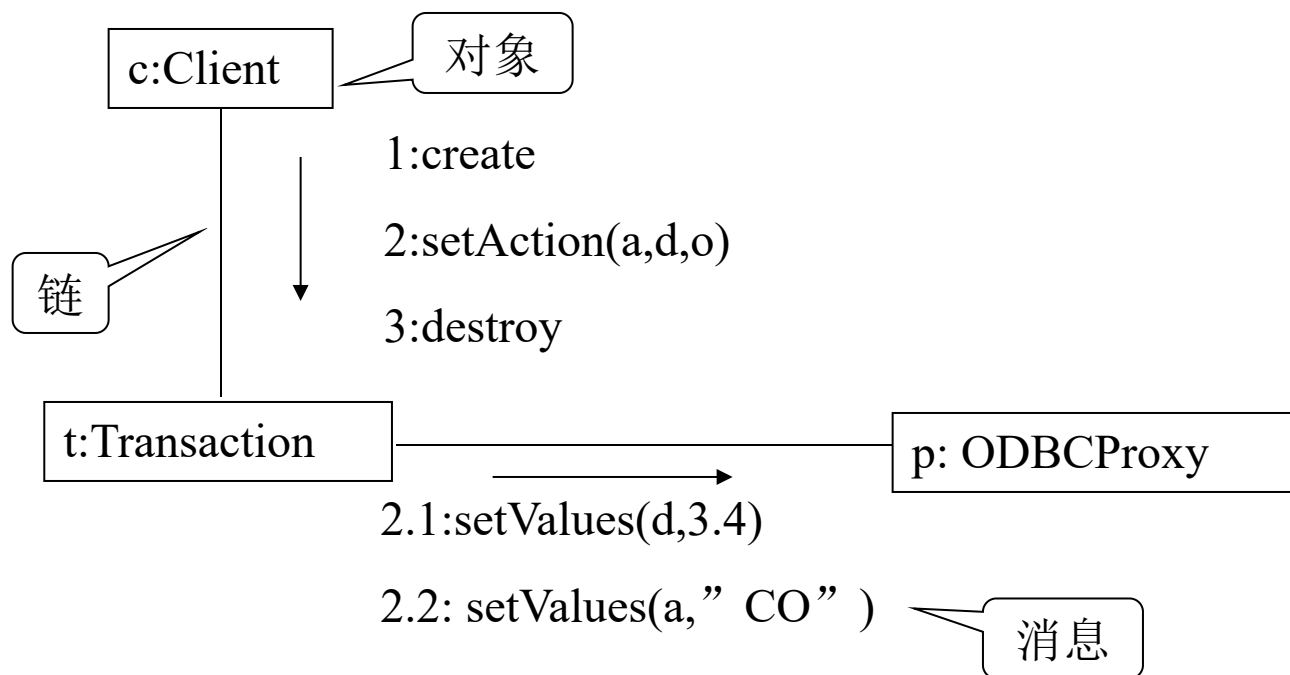
协作图 Collaboration Diagram 表示协同完成某行为对象之间的交互，强调对象的消息结构，而忽略时间顺序。

协作图由对象，消息，链等构成。

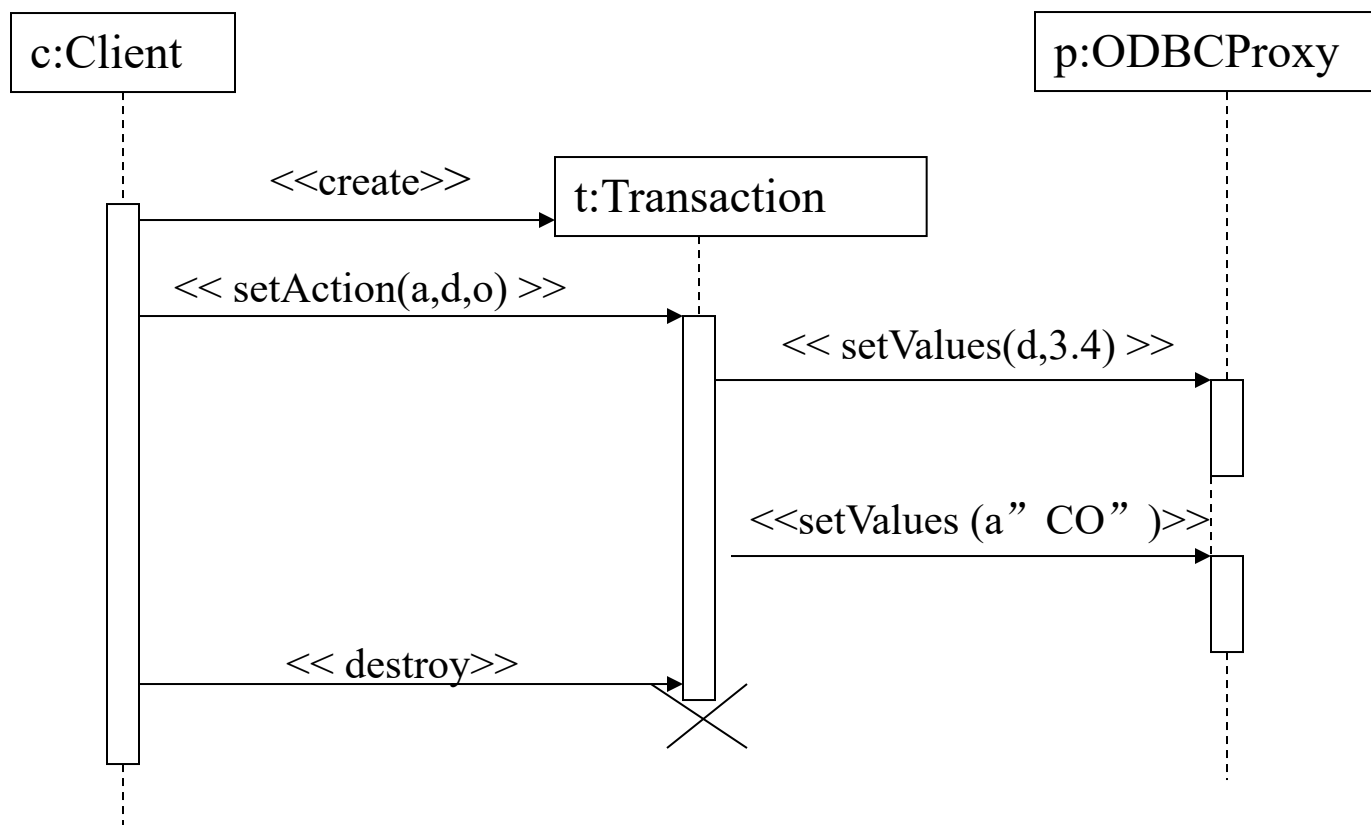
- **对象**：类的实例。对象是角色所属类的直接或间接实例，在协作图中，一个类的对象可能充当多个角色。
- **消息**：消息用来描述系统动态行为，它是从一个对象向另一个或几个对象发送信息，或由一个对象调用另一个对象的操作。
 - 由三部分组成：发送者，接收者，活动。
- **链**：表示两个或多个对象间的独立连接，是关联的实例。



协作图示例



与前页协作图在语义上等价的顺序图

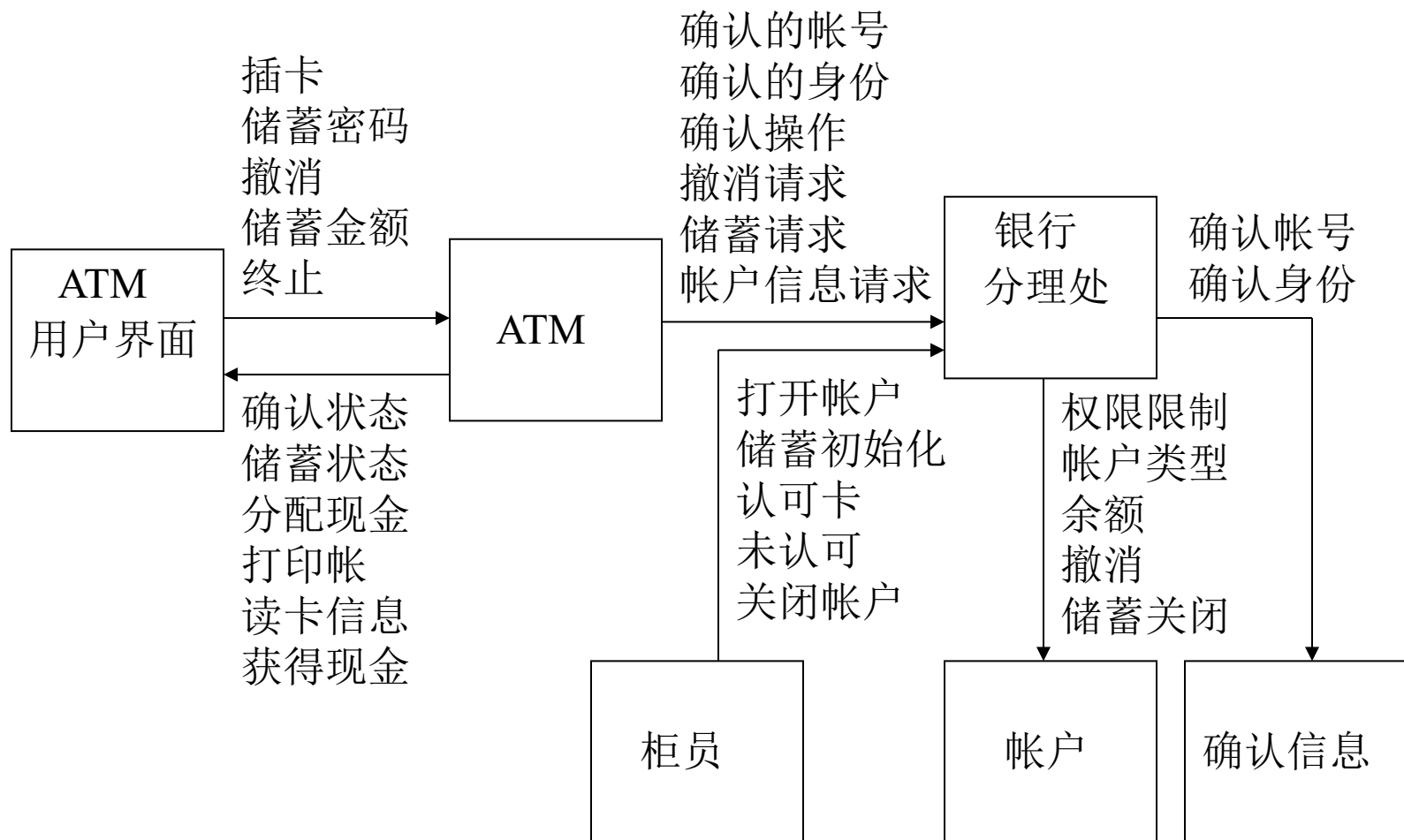


顺序图和协作图表示相同的模型语义，可以把顺序图和协作图从一种形式转换为另一中形式。但它们都可以表示对方不能表示的某些内容。

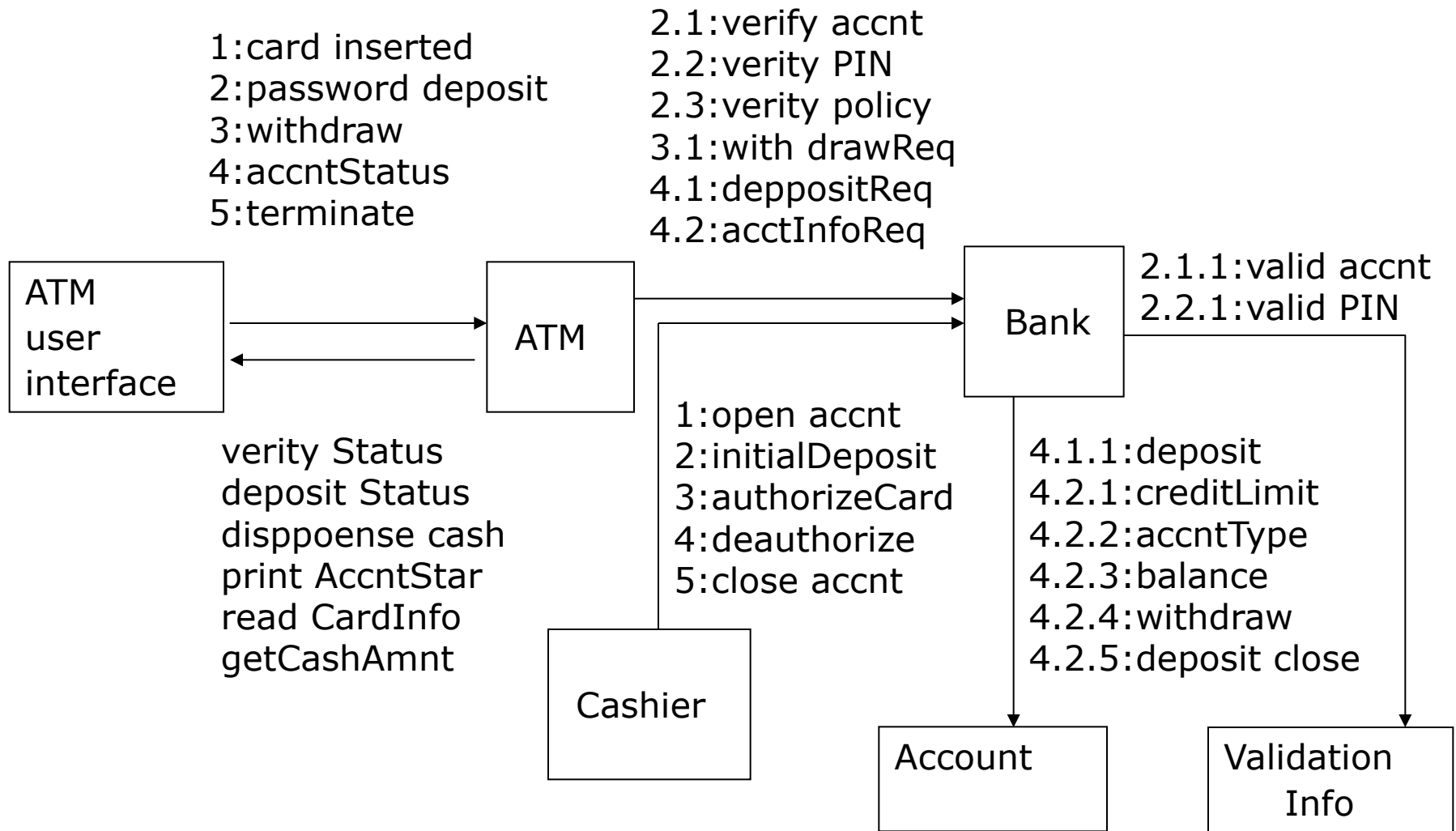
3.2. 协作图的用途

可用于分析对象行为的控制线程，规划系统分布及测试用例

例：用协作图规划储蓄应用的行为：



例：用协作图规划储蓄应用的行为：



面向对象分析与UML建模(4)

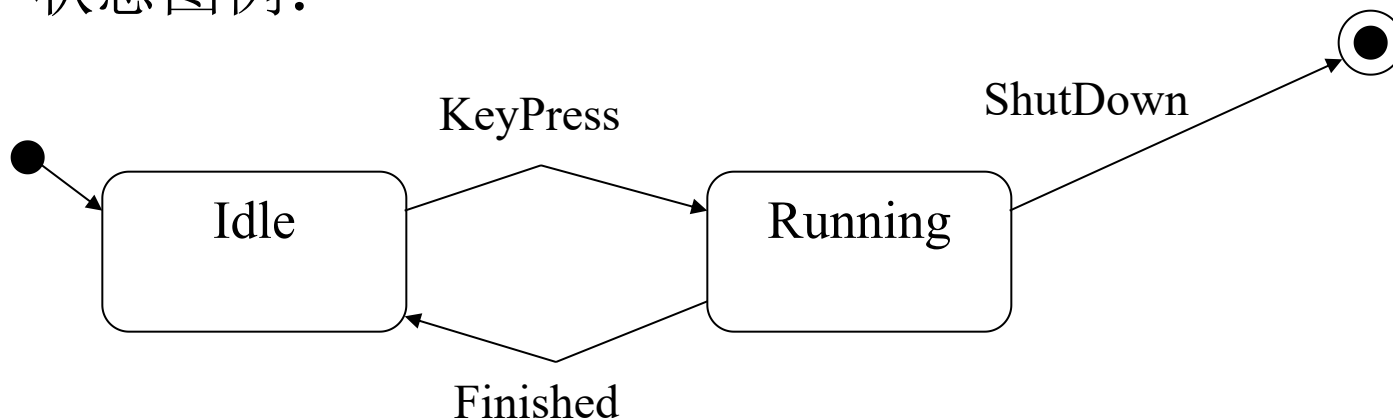
- 1. 动态模型
- 2. 顺序图
 - 2.1. 顺序图概念与表示
 - 2.2. 顺序图的建立
- 3. 协作图
 - 3.1. 协作图概念与表示
 - 3.2. 协作图的用途
- 4. 状态图
 - 4.1. 状态图概念与表示
 - 4.2. 状态图的建立
- 5. 活动图
 - 5.1. 活动图概念与表示
 - 5.2. 活动图应用
- 6. 建立动态模型示例

状态图概念与表示

状态图State Transition:

表示对象事物的行为，描述了一个对象在其生命周期内响应事件所经历的状态序列以及对这些事件所做出的反应。

状态图例：

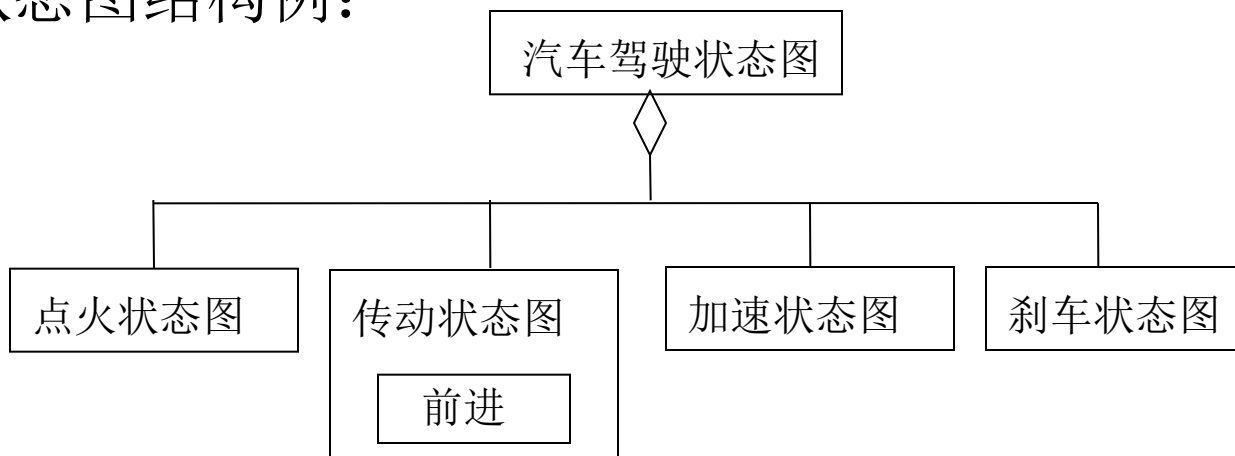


状态图的结构表示

状态图可以用组合状态表示其组织结构，表明由两个或多个子状态构成的状态。

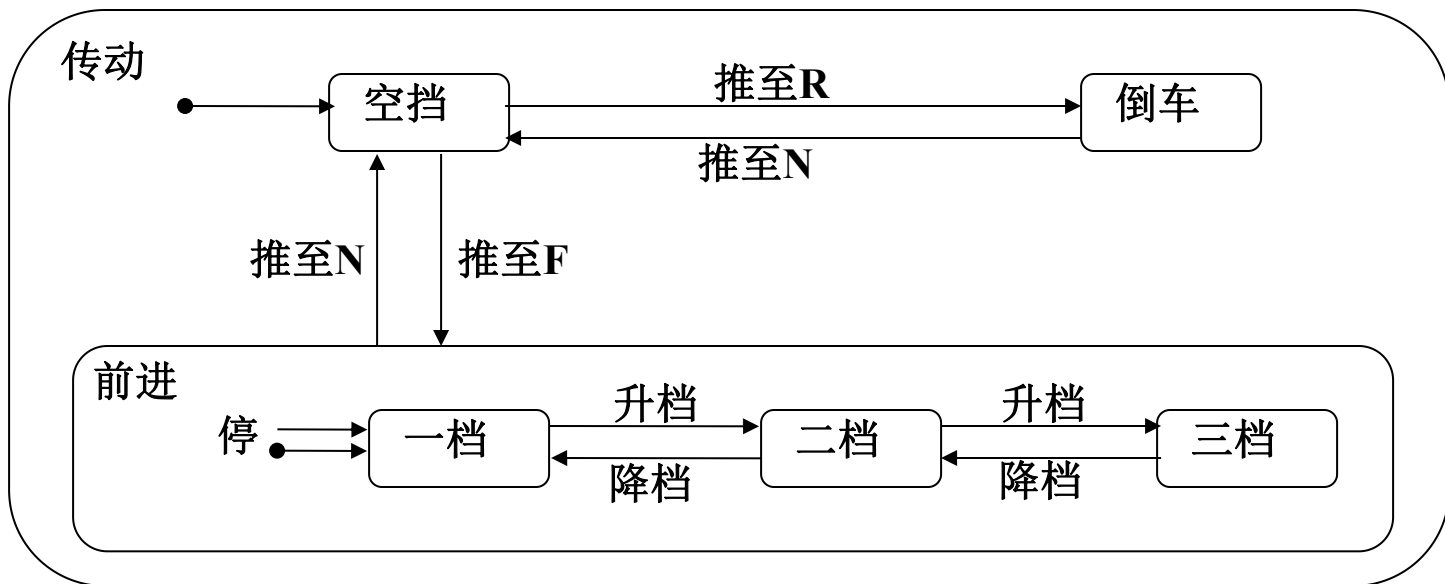
子状态包括：{ 顺序子状态
并发子状态
嵌套子状态

状态图结构例：

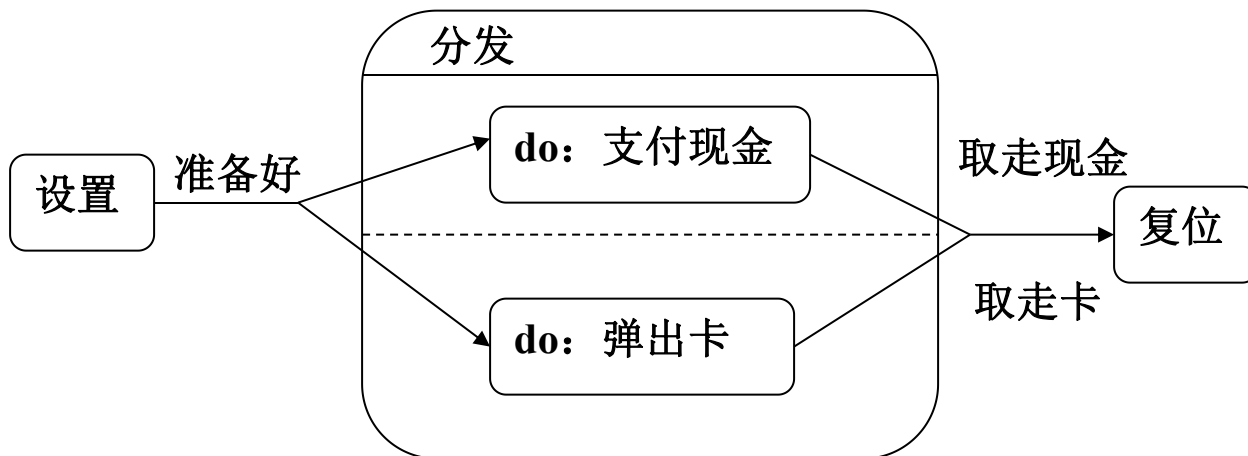


状态图的结构表示示例

顺序子状态例：



并发子状态例：



状态图的建立

建立策略：

- 1) 考虑某对象在特定语境中的交互行为；
- 2) 建立初始状态和终止状态；
- 3) 从属性值的范围和条件考虑对象所在的稳定状态；
- 4) 从对象的生命期开始，确定状态转换；
- 5) 决定对象可能响应的事件；
- 6) 用事件连接状态，给出事件名称、条件和动作。
- 7) 描绘各状态进入或退出的动作及保持状态的动作。
- 8) 从对象高层状态描绘可能的子状态（必要的时候）



面向对象分析与UML建模(4)

- 1. 动态模型
- 2. 顺序图
 - 2.1. 顺序图概念与表示
 - 2.2. 顺序图的建立
- 3. 协作图
 - 3.1. 协作图概念与表示
 - 3.2. 协作图的用途
- 4. 状态图
 - 4.1. 状态图概念与表示
 - 4.2. 状态图的建立
- 5. 活动图
 - 5.1. 活动图概念与表示
 - 5.2. 活动图应用
- 6. 建立动态模型示例

活动图概念与表示

活动图Activity Diagram:

描述系统的工作流程和并发行为，与状态图相比它强调流程的控制而不是状态在事件下的变化。

活动图的作用：

- 描述业务过程，特别能较好地表示并发流程，帮助理解涉及多个用户的工作流程
- 描述具体操作算法，与程序框图有相同的作用

活动图的概念接近结构化方法的流程图思想。



活动图元素

包括：

- 1) 活动---流程中的任务执行单元
- 2) 泳道---活动的区域划分
- 3) 分支---活动转向的分支
- 4) 分叉--- 并发控制流的分支
- 5) 汇合---分叉后的合并
- 6) 对象流---活动图中的控制流



活动图元素: 活动

■活动图中的“活动” Activity

包括: 动作状态 Action State

Calculate Total Cost

活动状态 Activity State

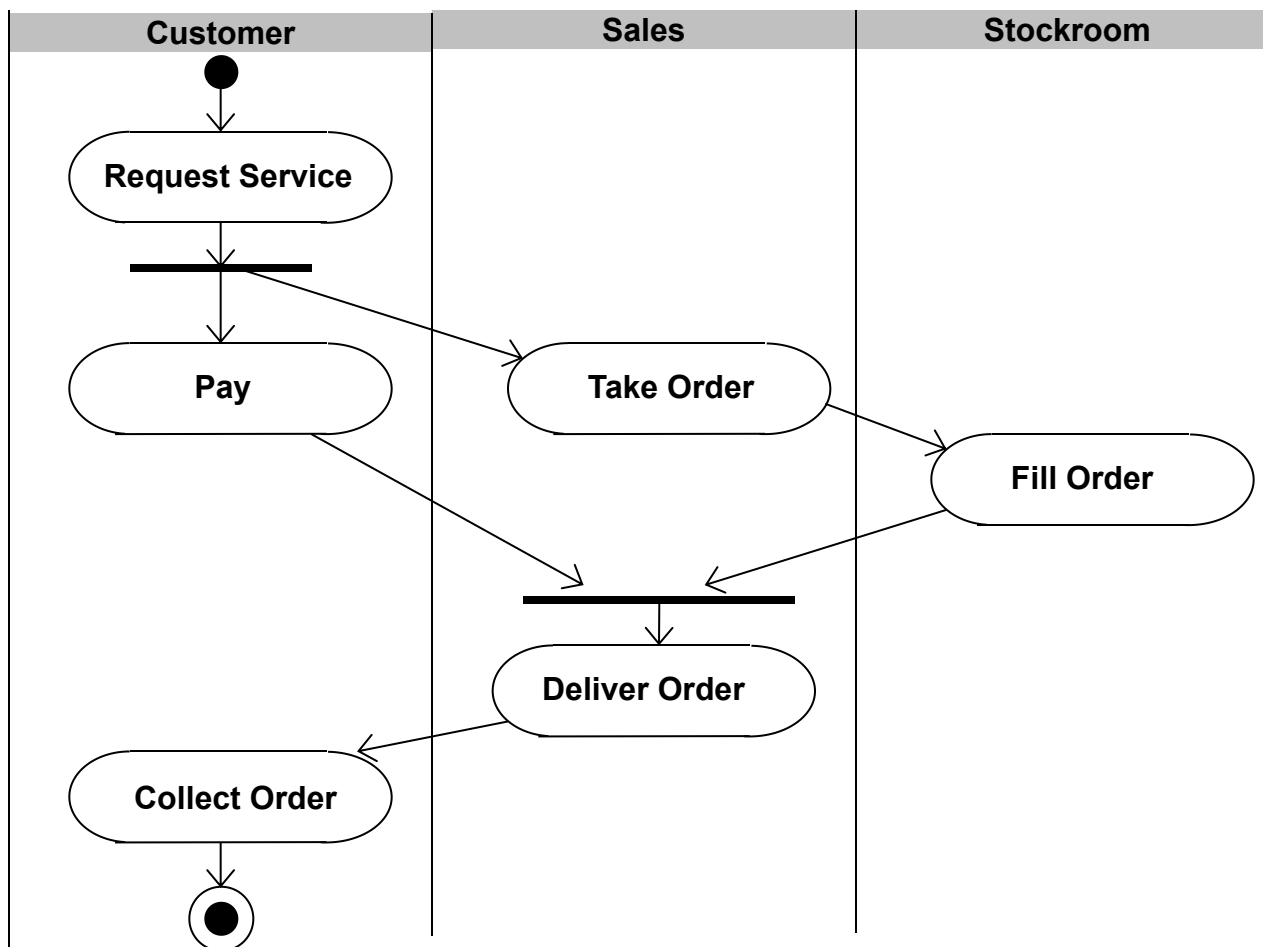
- **动作状态**: 没有内部转移和内部活动, 是原子的最小执行单元, 作用是表示执行进入动作后转向的状态, 执行时间可以被忽略。
- **活动状态**: 可分解的、非原子的, 并有一定持续时间的执行单元。



活动图元素：泳道

“泳道” swimlane：根据每项活动的**职责**，划分所有活动的责任区域

例：

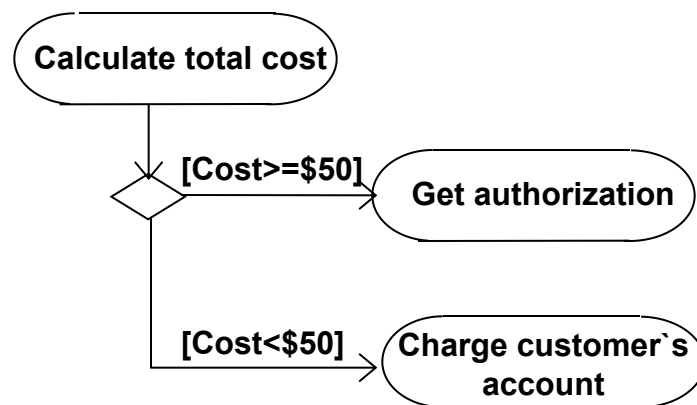
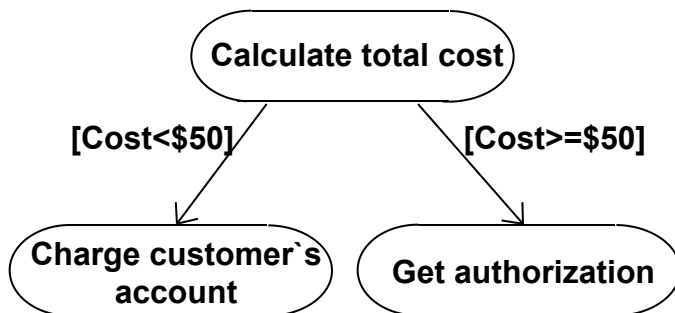


活动图元素: 分支

■活动图中的“分支” branch

在活动图中同一触发事件，可根据警戒条件转向不同的活动，有**两种表示方法**：

例：

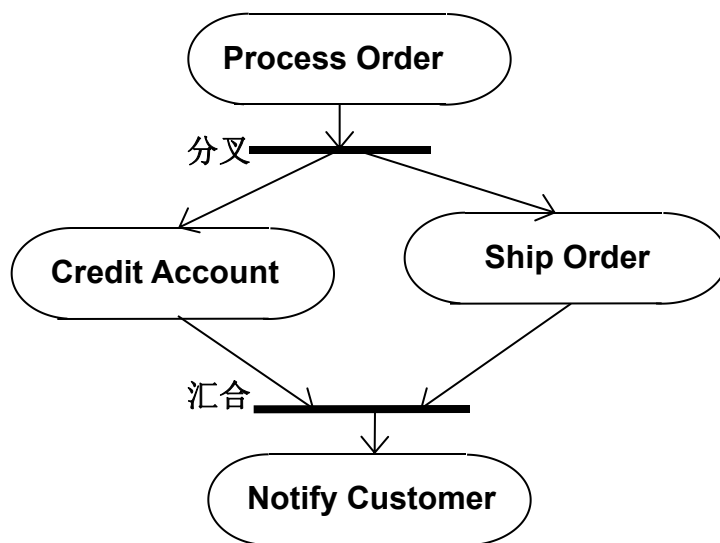


活动图元素: 汇合

■活动图中的“分叉和汇合” fork and join

在活动图中同一控制流被两个或多个控制流替代且并发，其转换点为分叉；汇合则与此过程相反。

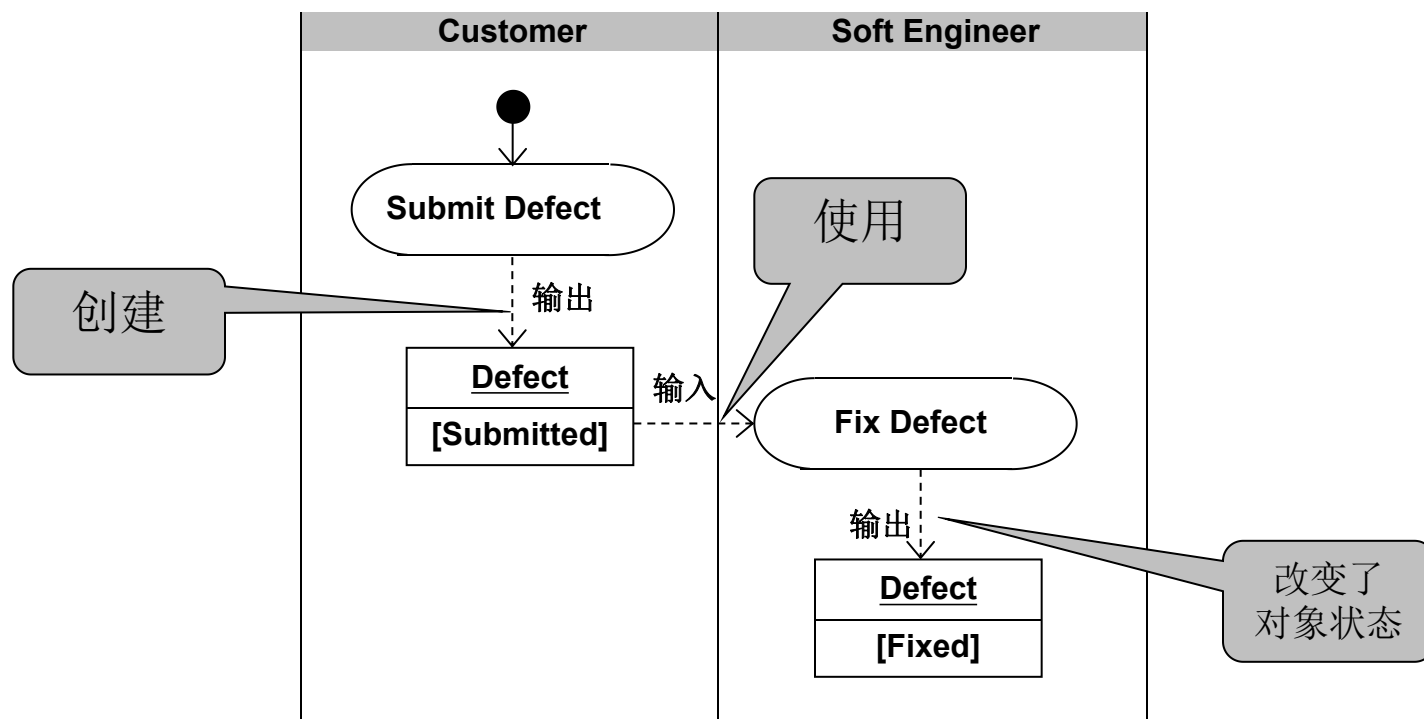
例：



活动图元素: 对象流

■活动图中的“对象流”

对象作为活动的输入或输出，对象流可表示对象与活动之间的行为，对象流属于控制流。



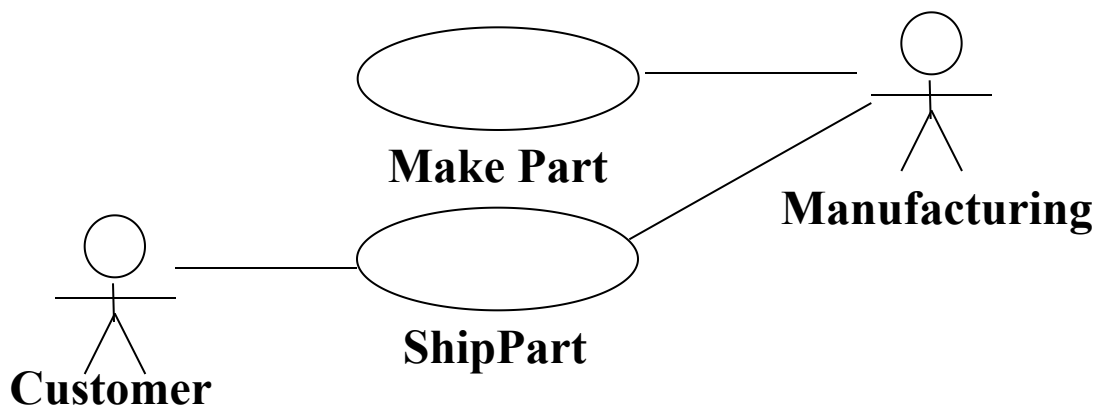
活动图应用

■用活动图对 workflow 建模

- workflow 建模：系统的业务过程的描述
- 用例图的局限性：

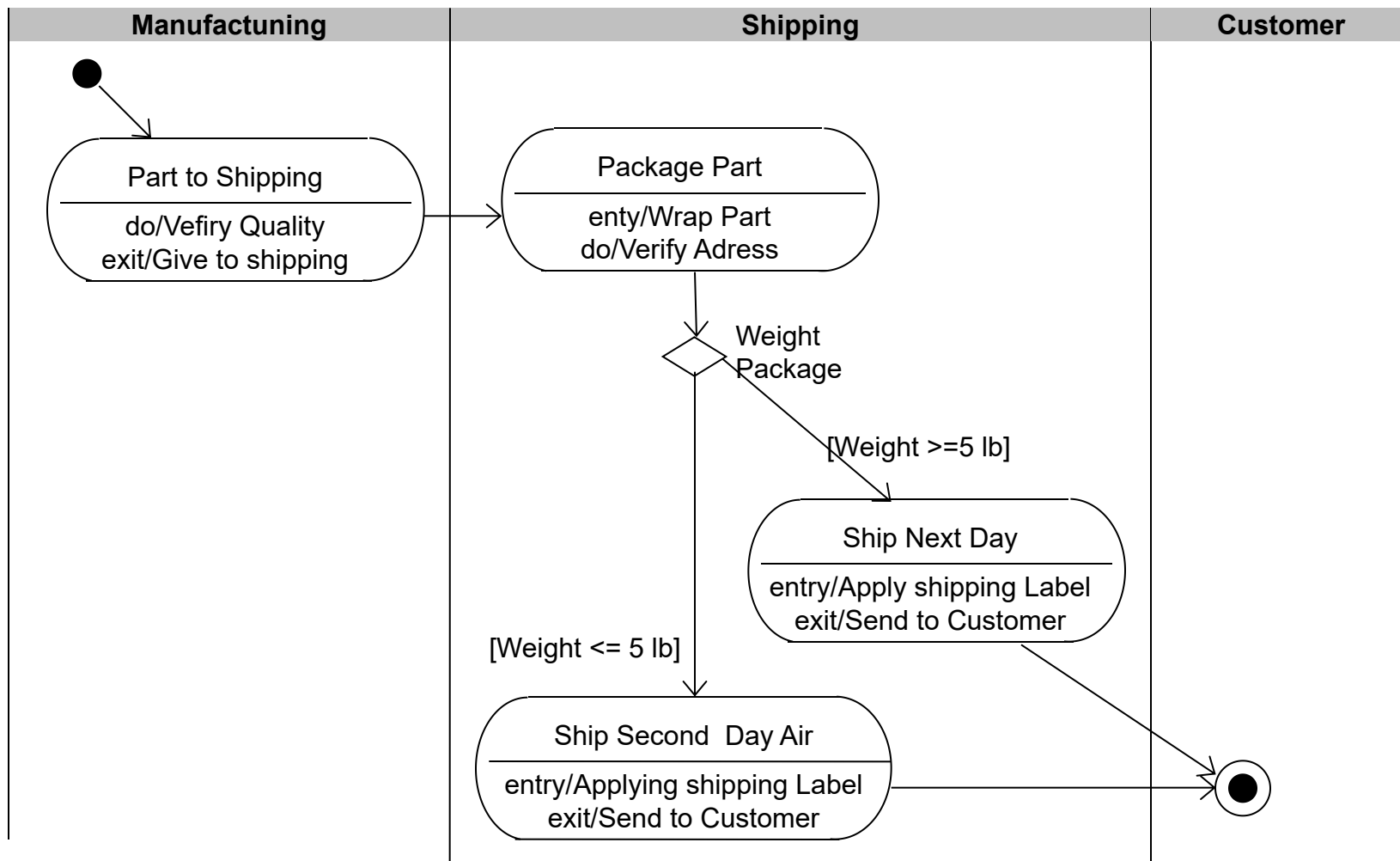
用例图是以系统的各独立功能为单位描述的，**并不涉及到整体业务过程**，在进行系统分析时，有时需要对业务过程进行必要的描述。

例：产品制造和发货过程在用例图中无法表示



活动图示例

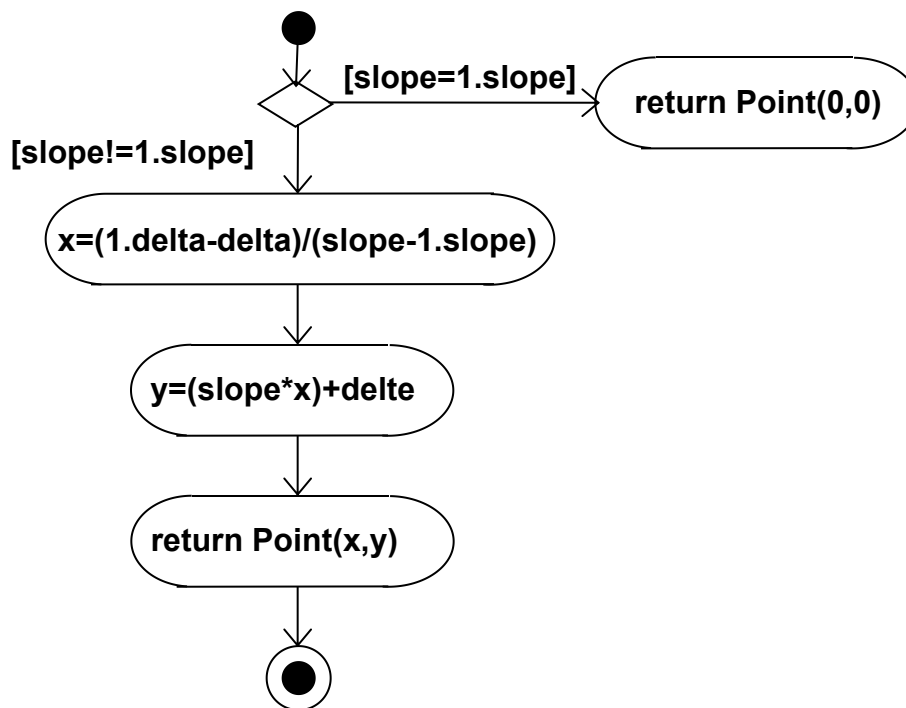
例：用活动图描述产品制造和发货过程



用活动图对具体操作建模

用活动图描述具体算法，类似于结构化分析时的流程图

例：Line类的求直线焦点的算法

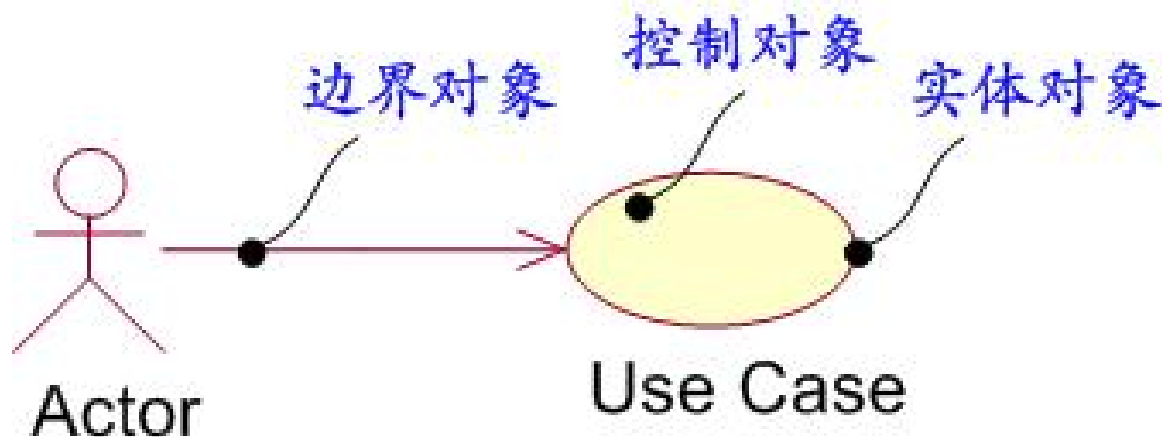


面向对象分析与UML建模(4)

- 1. 动态模型
- 2. 顺序图
 - 2.1. 顺序图概念与表示
 - 2.2. 顺序图的建立
- 3. 协作图
 - 3.1. 协作图概念与表示
 - 3.2. 协作图的用途
- 4. 状态图
 - 4.1. 状态图概念与表示
 - 4.2. 状态图的建立
- 5. 活动图
 - 5.1. 活动图概念与表示
 - 5.2. 活动图应用
- 6. 建立动态模型示例

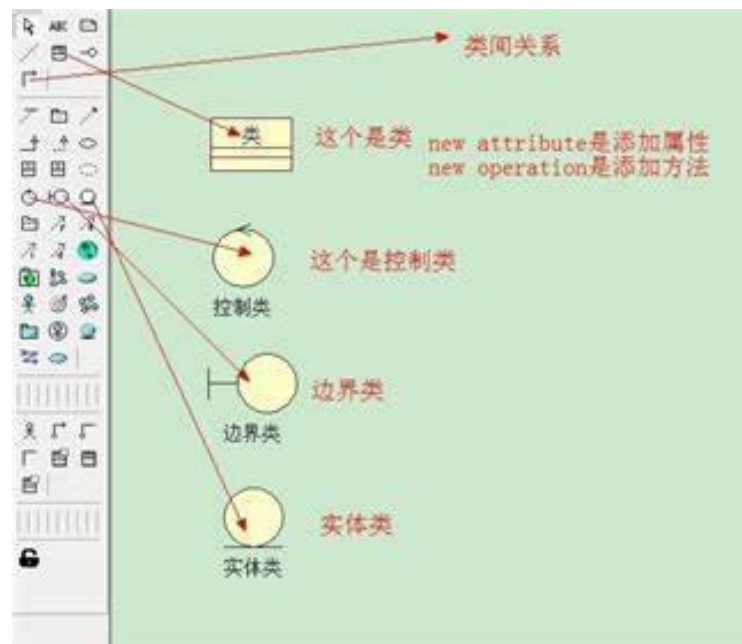
建立动态模型示例

- 从一个用例中抽取三类对象的方法:



建立动态模型示例

- 将应用边界类、控制类和实体类
 - 边界类：描述外部参与者与系统之间的交互
 - 实体类：存储和管理系统内部的信息(持久存储体的信息)
 - 控制类：描述一个用例所具有的事件流控制行为，控制一个用例中的事件顺序



从业务场景/事件流开

3.1 基本事件流

1. 图书管理员向系统发出“新增书籍信息”请求；
2. 系统要求图书管理员选择要新增的书籍是计算机类还是非计算机类；
3. 图书管理员做出选择后，显示相应界面，让图书管理员输入信息，并自动根据书号规则生成书号；
4. 图书管理员输入书籍的相关信息，包括：书名、作者、出版社、ISBN号、开本、页数、定价、是否有CDROM；
5. 系统确认输入的信息中书名未有重名；
6. 系统将所输入的信息存储建档。

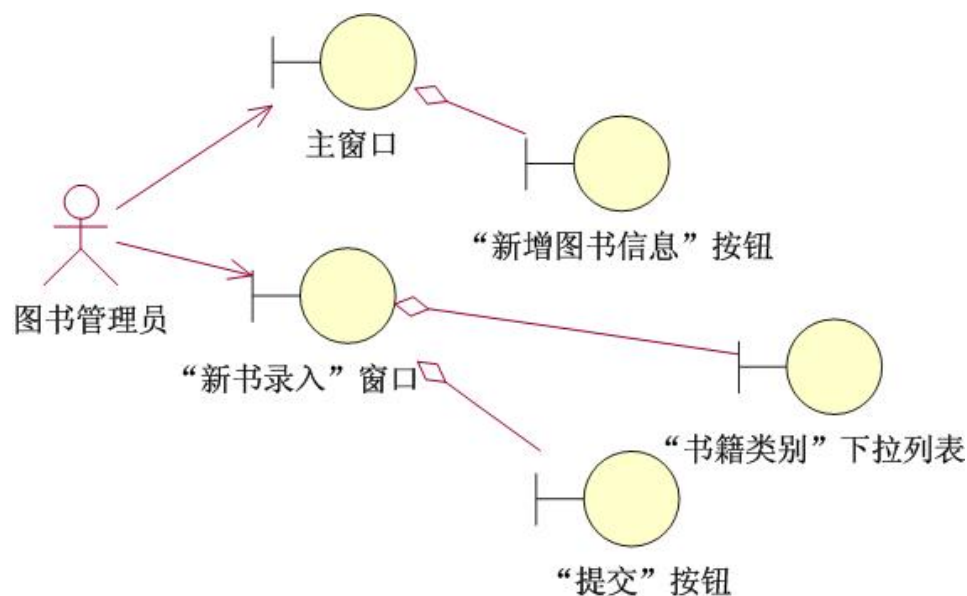
3.2 扩展事件流

- 5a) 如果输入的书名有重名现象，则显示出重名的书籍，并要求图书管理员选择修改书名或取消输入；
 - 5a1) 图书管理员选择取消输入，则结束用例，不做存储建档工作；
 - 5a2) 图书管理员选择修改书名后，转到 5)



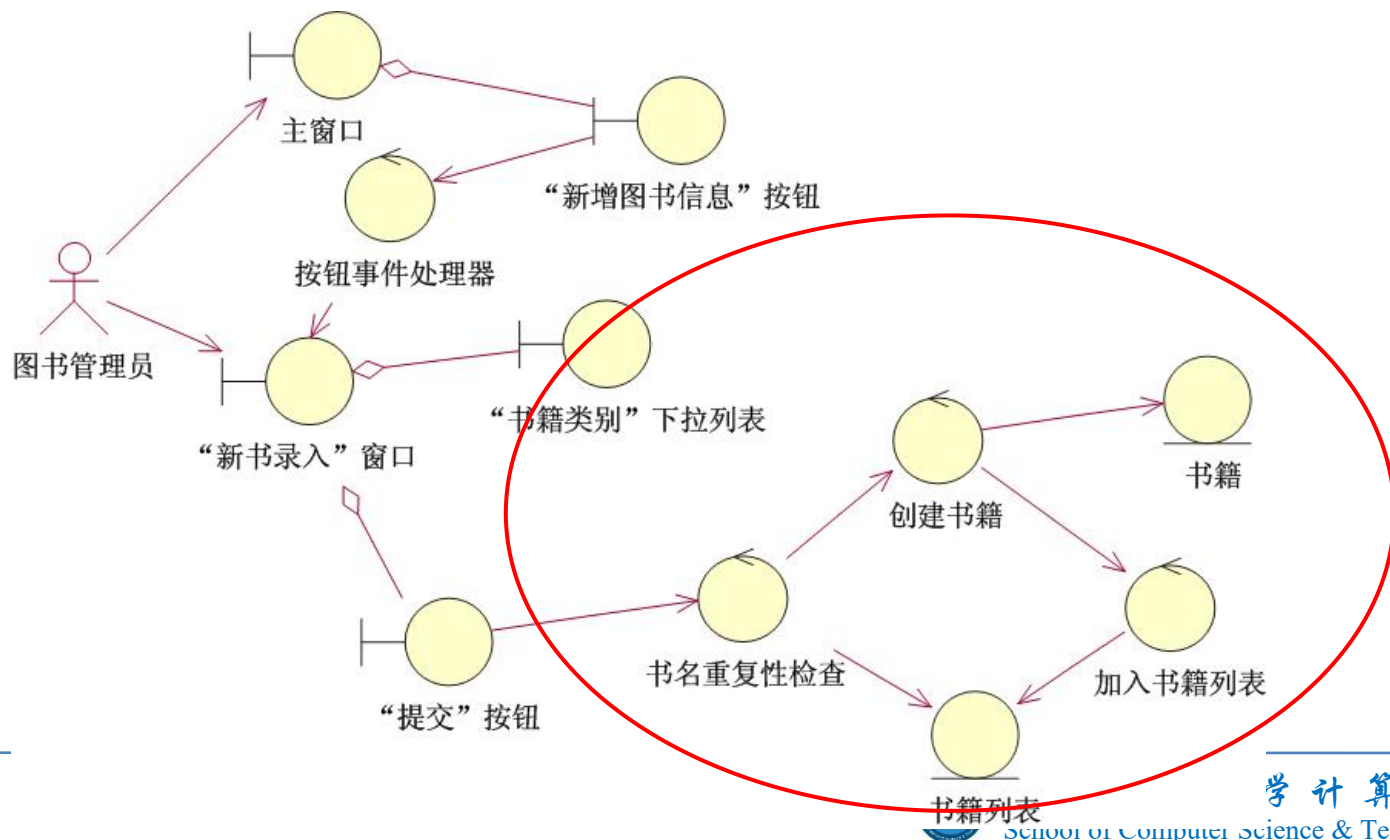
寻找边界对象

- 图书管理员向系统发出“新增书籍信息”请求——主窗口、“新增书籍信息”按钮
- 系统要求图书管理员选择要新增的书籍是计算机类还是非计算机类——书籍类别列表框。
- 图书管理员做出选择后，显示相应界面，让图书管理员输入信息，并自动根据书号规则生成书号——“新书信息录入”窗口及辅助的“提交”按钮



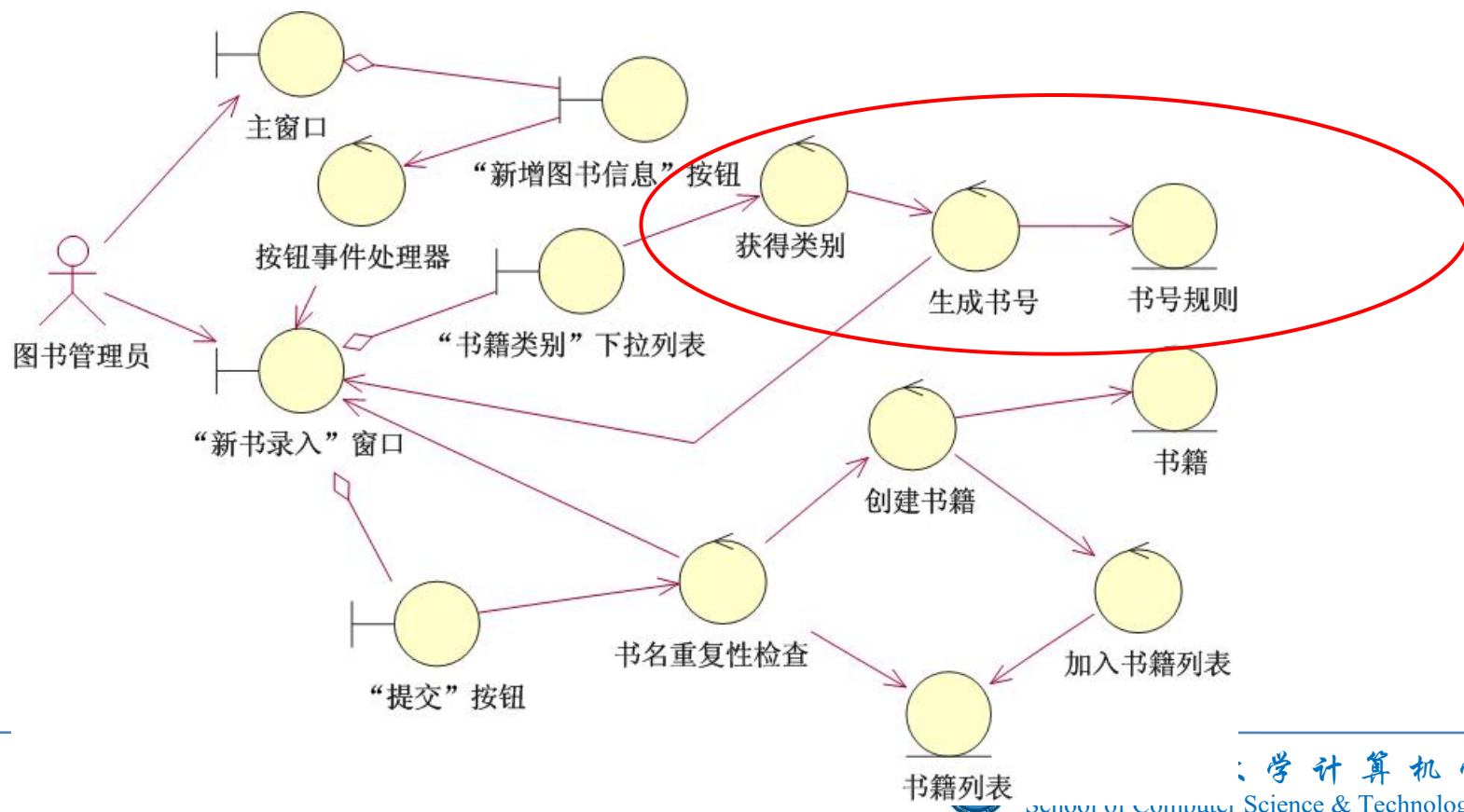
寻找控制对象和实体对象

- 根据事件流中的步骤5，以及扩展路径的描述，就可以在原图上增加相应的控制对象，得到更进一步的分析图

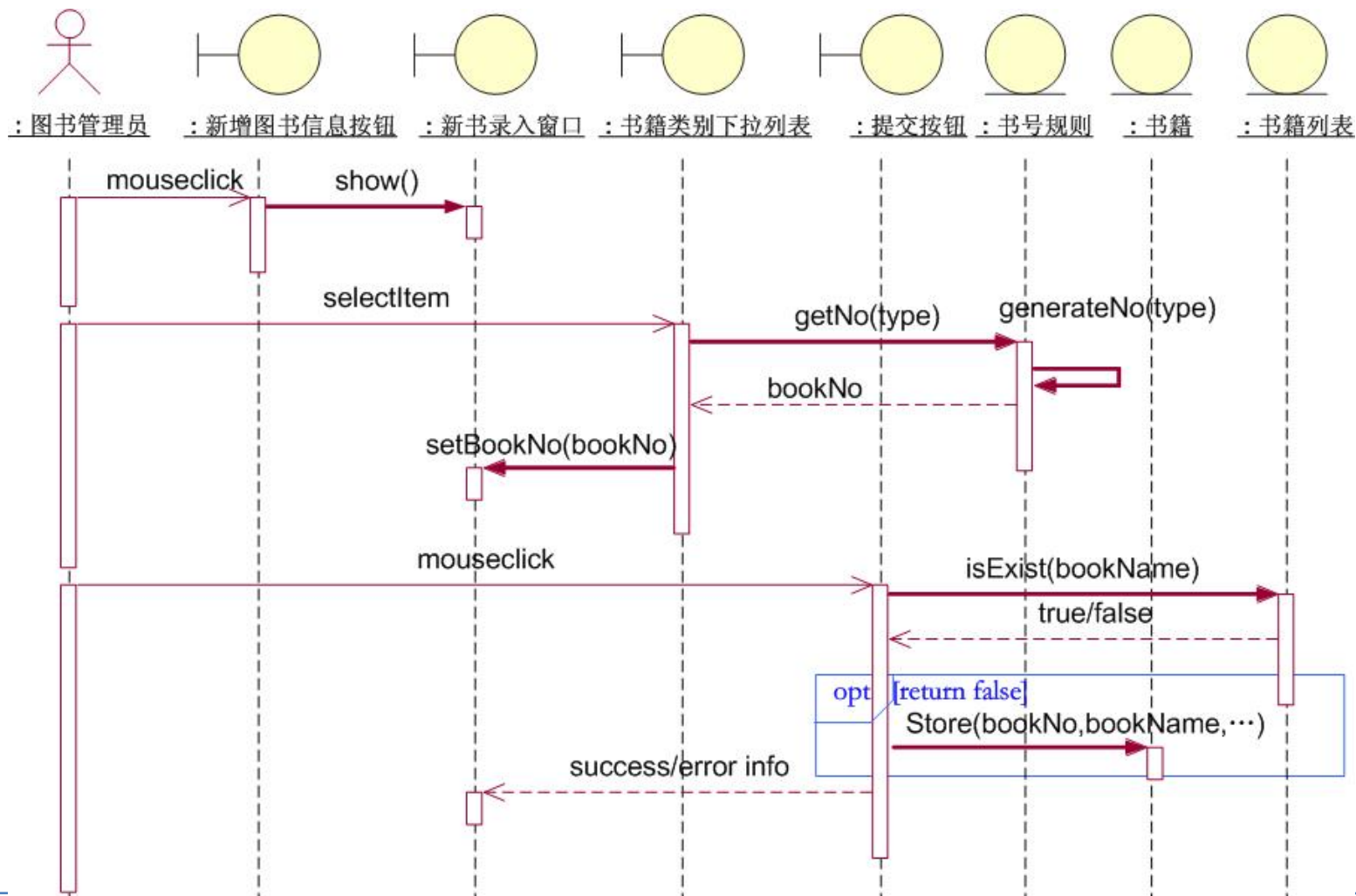


寻找控制对象和实体对象

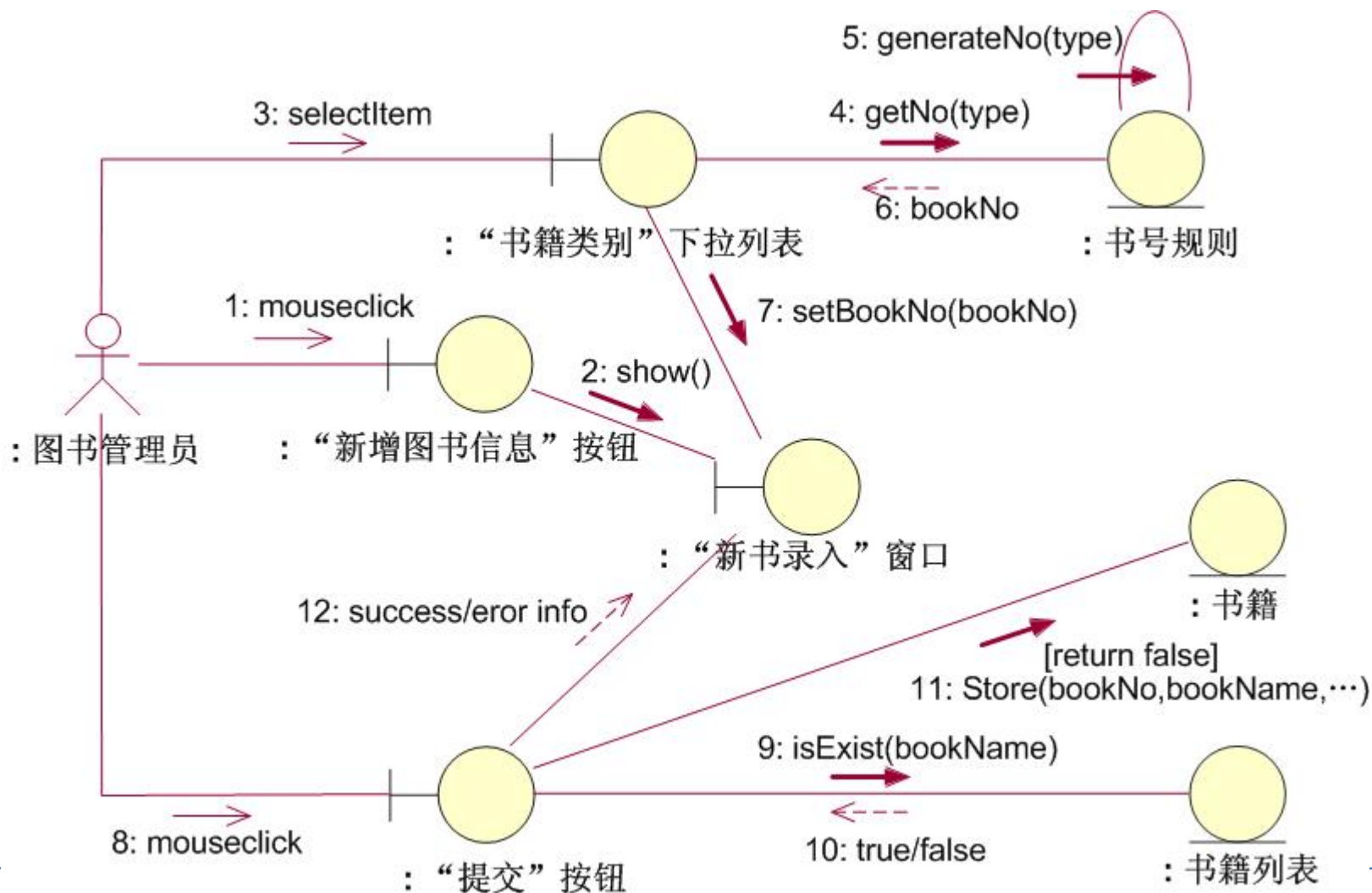
- 新添两个逻辑：一是基本事件流中的步骤2、3要求根据用户选择的类别，自动获得书号；二是当书名重复性检查没有通过（有重名），则应返回要求其重输入



构建交互模型: 顺序图



转换成协作图



本节小结

- 系统行为建模
- 交互图
 - 顺序图
 - 协作图
- 状态图
- 活动图