



活动图

内容

- 概述
- 基本概念
- 建模方法



内容

➤ 概述

- 基本概念
- 建模方法



活动图（activity diagram）

- 活动图用来描述事情进行的流程，是UML中对系统行为建模的一种常用工具
- 活动图描述系统的控制流：从一种活动转换到另一种活动的整个过程
- 活动图可以对系统的工作流程建模，也可以对具体的操作建模



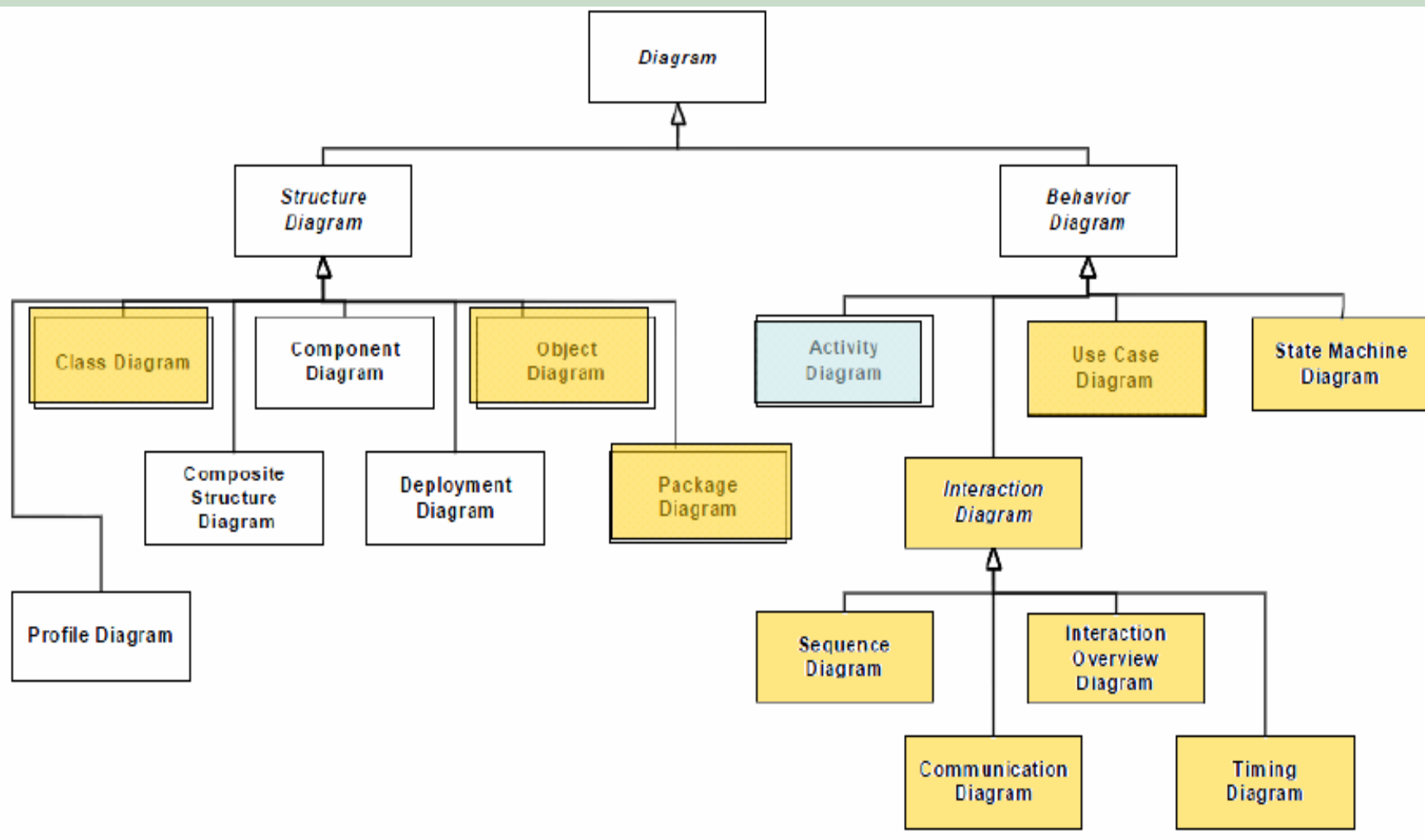
活动图与流程图

活动图与流程图类似，但有区别：

- 活动图能够表示并发活动，而流程图不能
- 活动图是面向对象的，而流程图是面向过程的
- 流程图着重描述处理过程，活动图着重表现系统行为



活动图

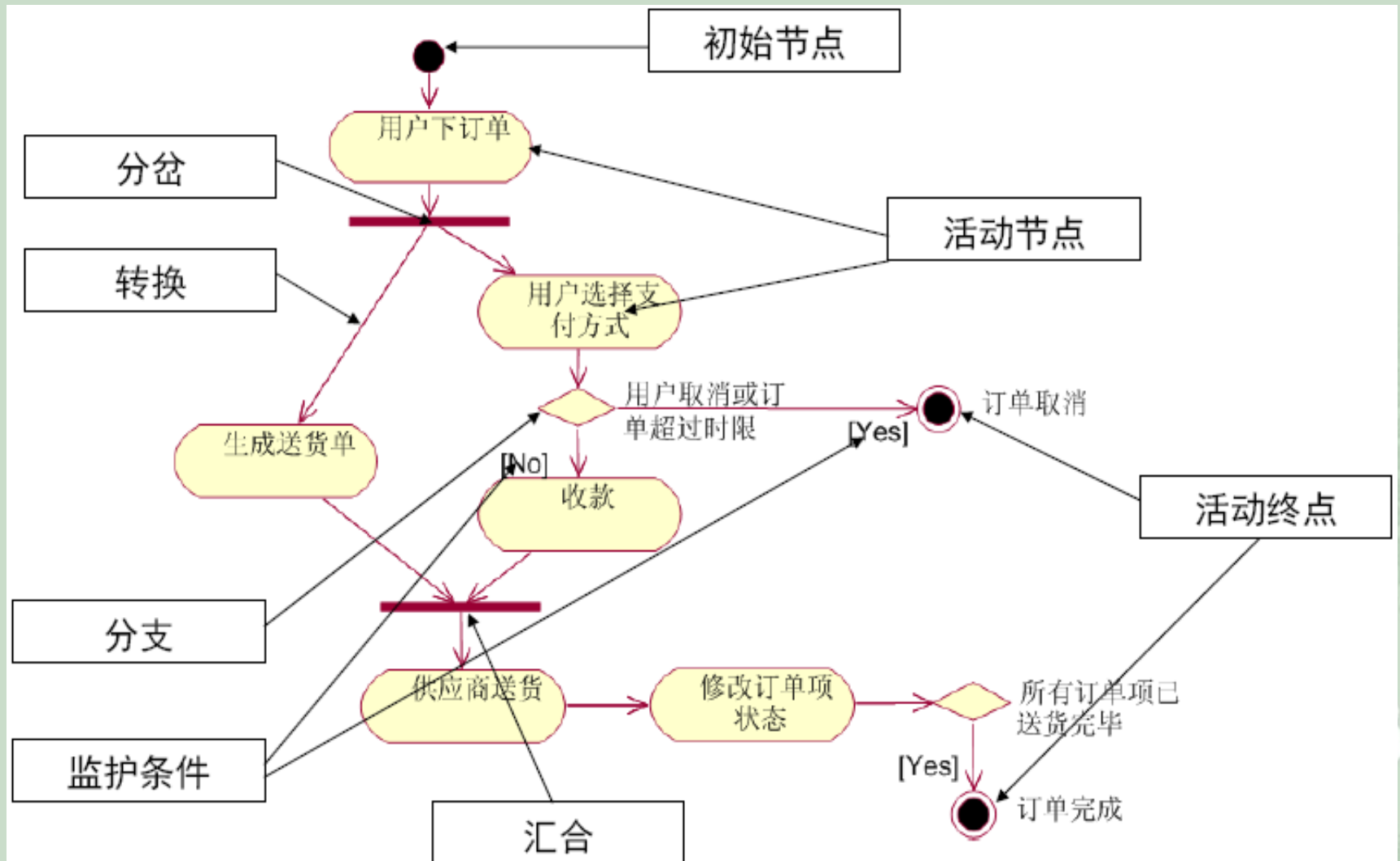


活动图的基本构成元素

- 活动
- 转换
- 分支、监护条件和合并
- 分岔与汇合



活动图示例



内容

- 概述
- 基本概念
- 建模方法



活动图的构成元素

- 活动
- 转换
- 分支、监护条件与合并
- 分岔与汇合
- 泳道
- 对象流



活动

- 活动描述事情进行的状态
- 活动表示流程中任务的执行或算法过程中语句的执行
- 活动用圆端矩形表示



选择课程

The diagram shows a yellow rounded rectangle with a dark red border, containing the text '选择课程' (Select Course). This shape is used to represent an activity in a flowchart or algorithm.




初始节点（initial node）

- 初始节点标记活动图的开始，表示事情处理过程的开始
- 在一个不包括子图的活动图中有且仅有一个初始节点
- 用实心圆来表示




活动终点（activity final）

- 活动终点标记活动图的结束，表示事情处理流程完成
- 在活动图中，可能包含多个活动终点
- 用一个圆圈加实心圆来表示 



转换

- 当一个活动执行结束，控制流就会马上传递给下一个活动节点，在活动图中称之为“转换”
- 与状态图不同，活动图的转换一般不需要特定事件的触发
- 转换用一条带箭头的直线来表示 



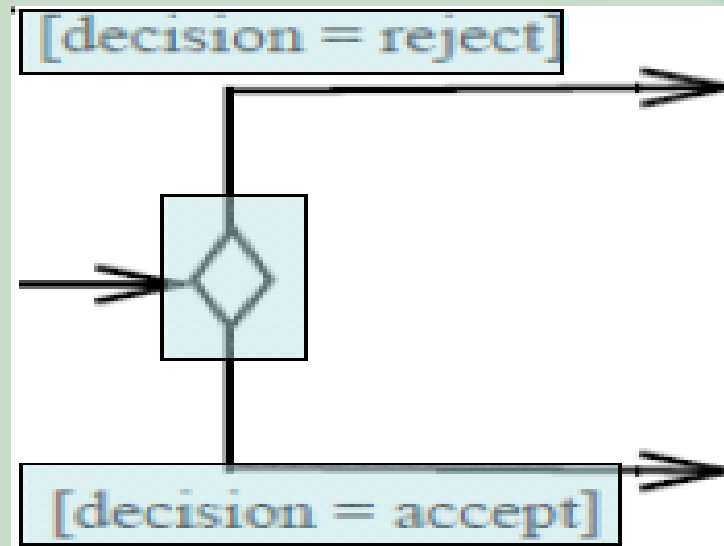
分支、监护条件与合并

➤ 分支（Decision node）

- 有条件的工作流需要根据条件来判断动作的转换方向，UML活动图中条件行为用分支和合并表达
- 分支用菱形表示，它有一个进入转换（箭头从外指向菱形），一个或多个离开转换（箭头从菱形指向外）
- 当两条选择性路径重新合并时，相同的符号适用

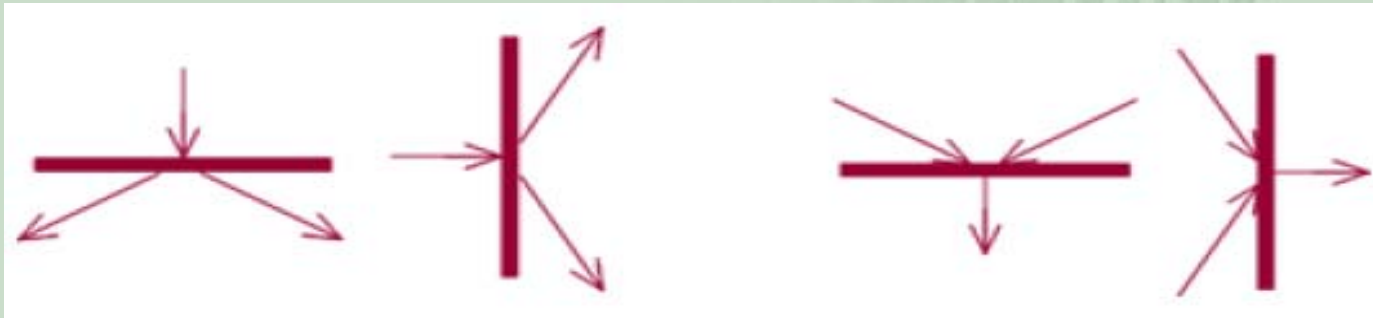
分支、监护条件与合并

- 监护条件（Guard condition）
 - 在分支的每个离开转换上都会有一个监护条件，用来表示满足什么条件的时候执行该转换



分岔与汇合

- 对象在运行时可能会存在多个并发运行的控制流
- UML中采用分岔和汇合来对并发控制流建模
- 分岔用于将一个控制流分为多个并发运行的独立控制流，有一个进入转换，有多个离开转换
- 汇合用于同步这些并发控制流，当所有控制流都到达汇合点后，控制才能继续运行，汇合有多个进入转换，有一个离开转换

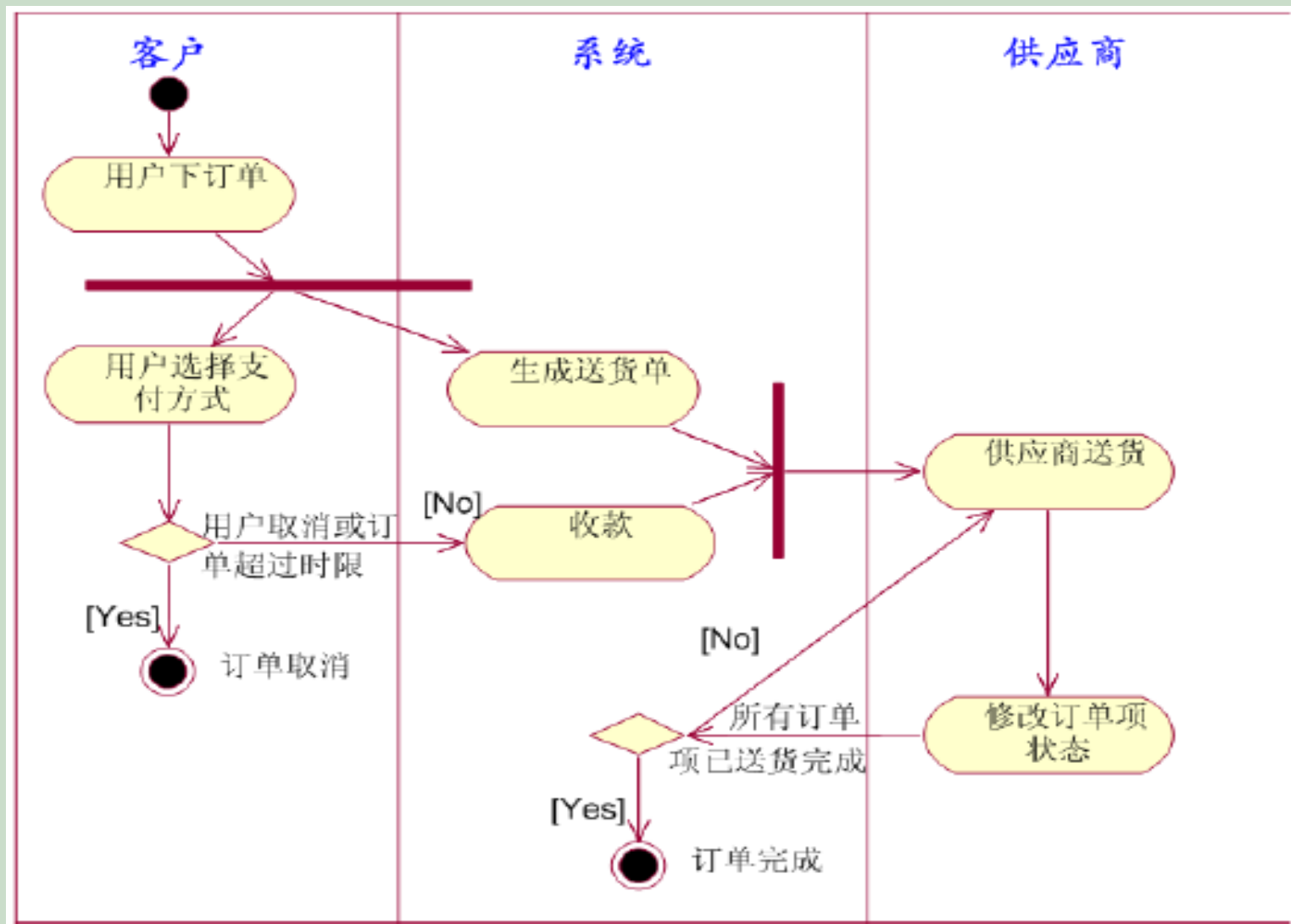


泳道

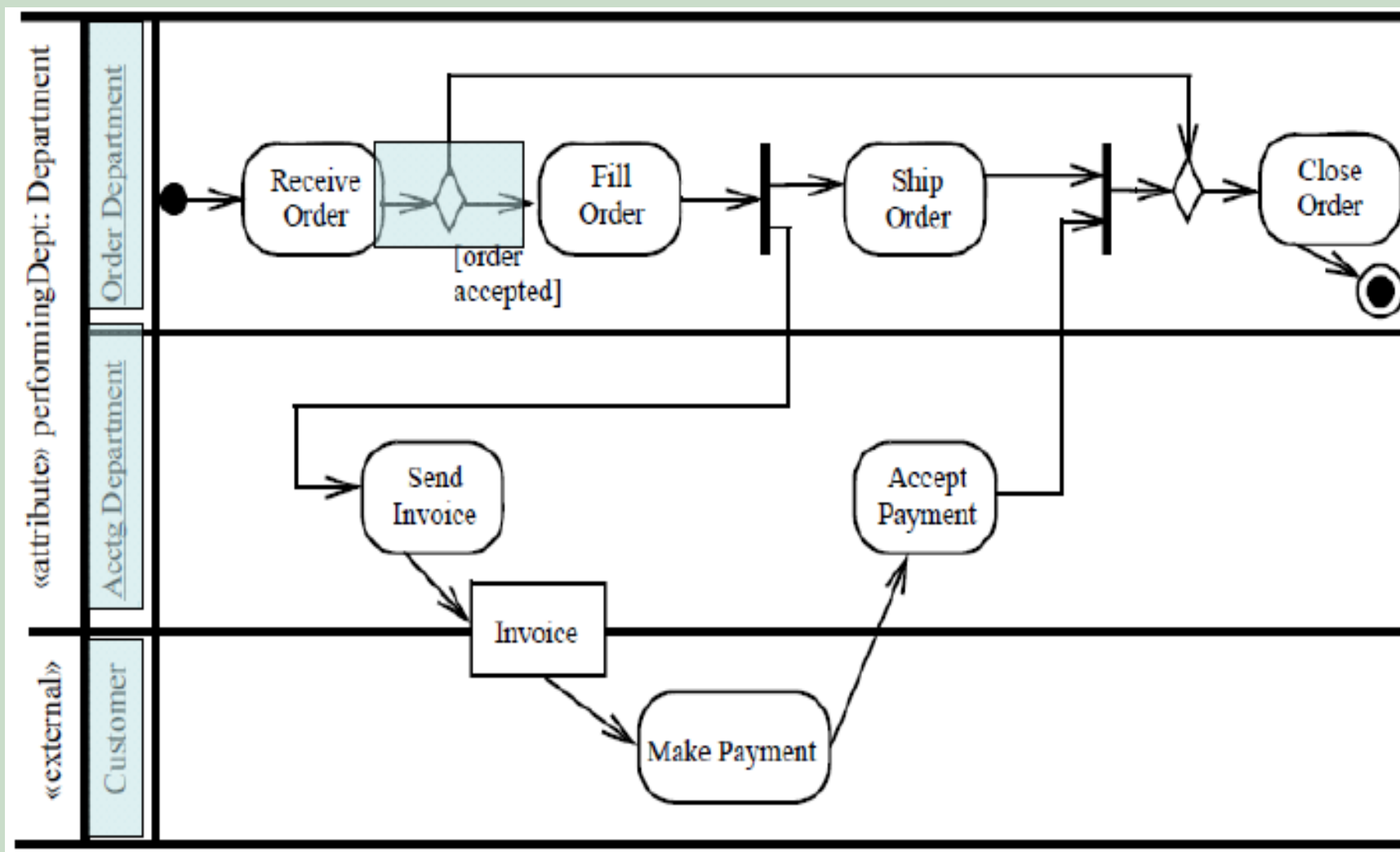
- 泳道将活动图中的活动分为若干区
- 泳道可以区分负责活动的对象，明确表示哪些活动是由哪些对象执行的
- 每个泳道必须有一个名称
- 每个活动只能明确地属于一个泳道
- 转换、分岔、汇合和对象流允许跨越泳道
- 分支只属于一个泳道



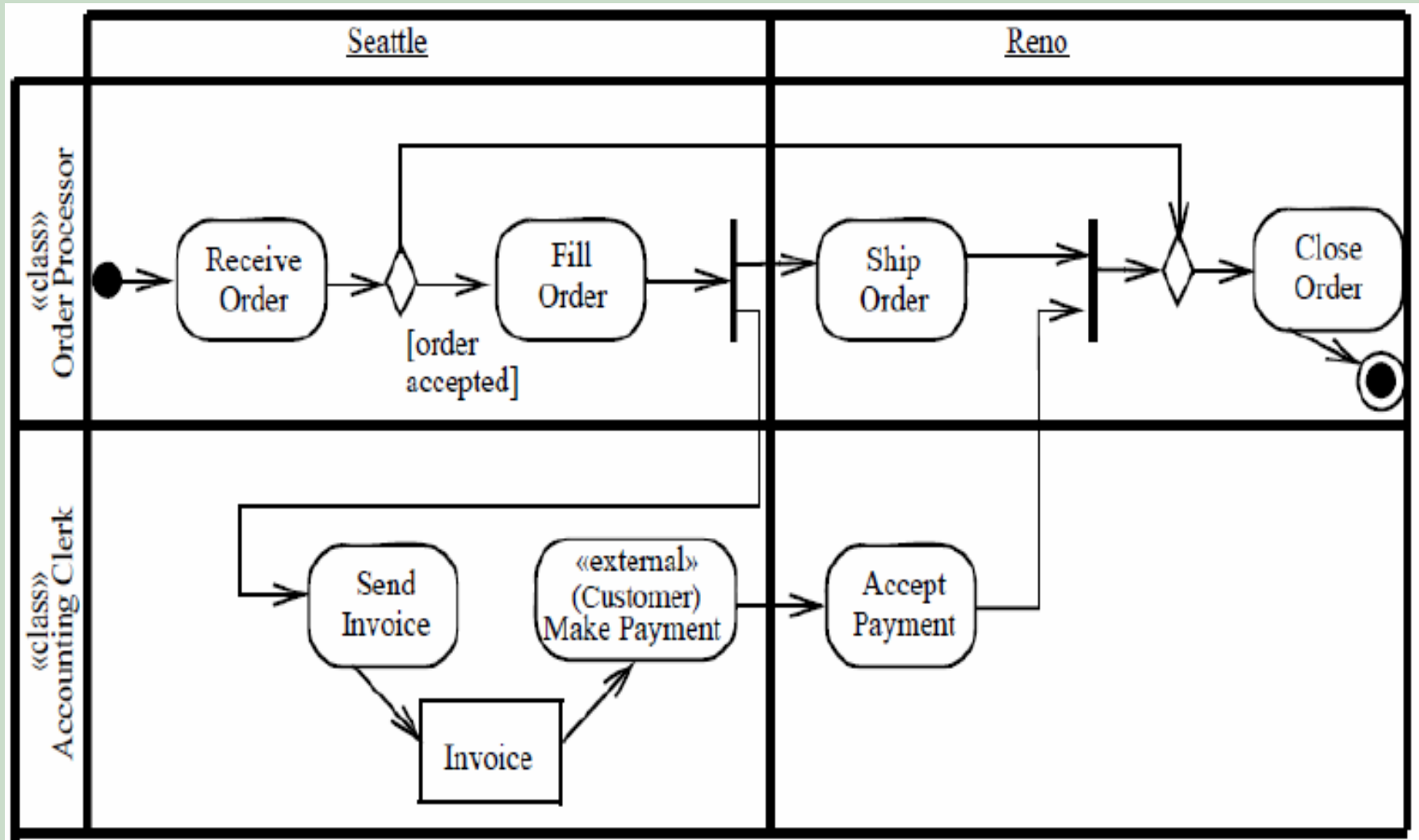
带泳道（swim lane）的活动图



带泳道的活动图示例

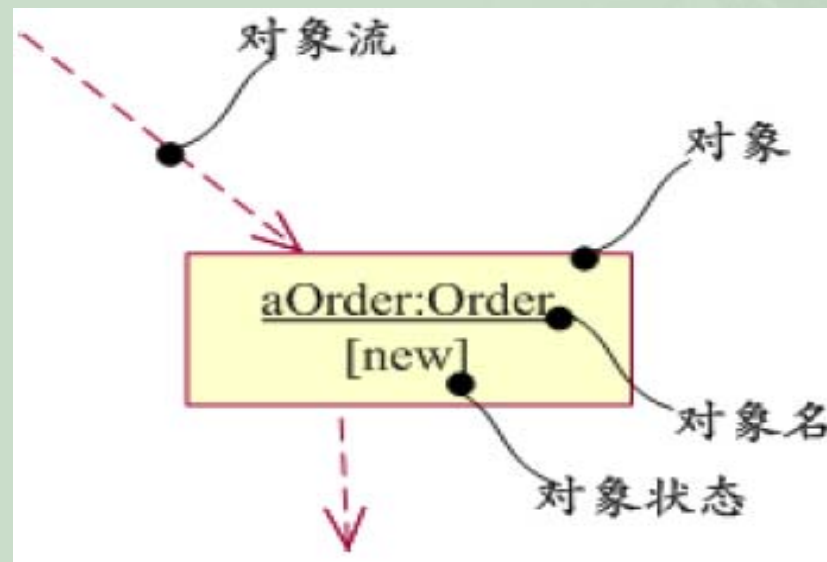


带泳道的活动图示例

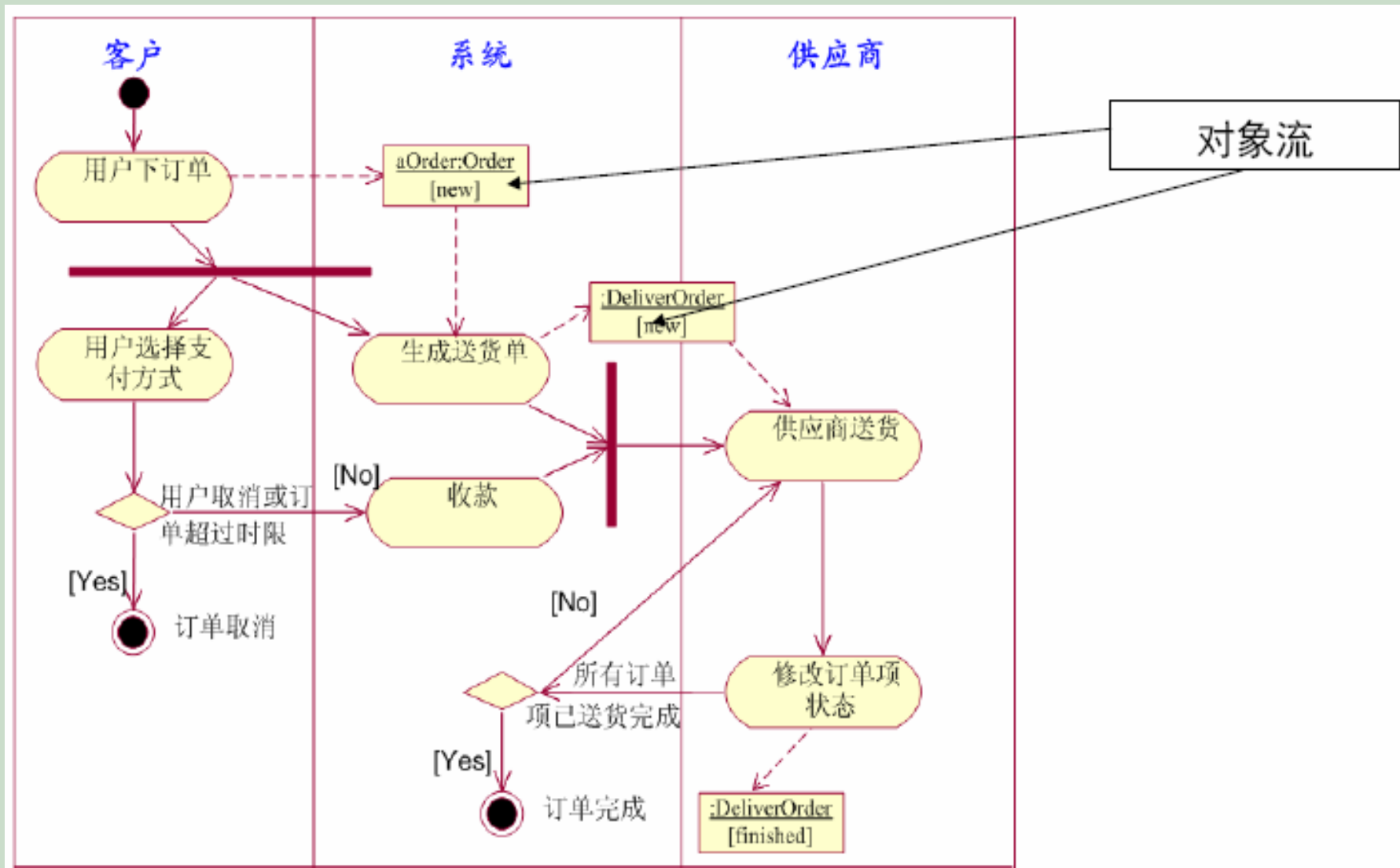


对象流

- 活动可以输入和输出对象，在活动图中可用对象流表示
- 对象流还可以显示活动间对象角色、状态和属性值的变化



带对象流的活动图示例



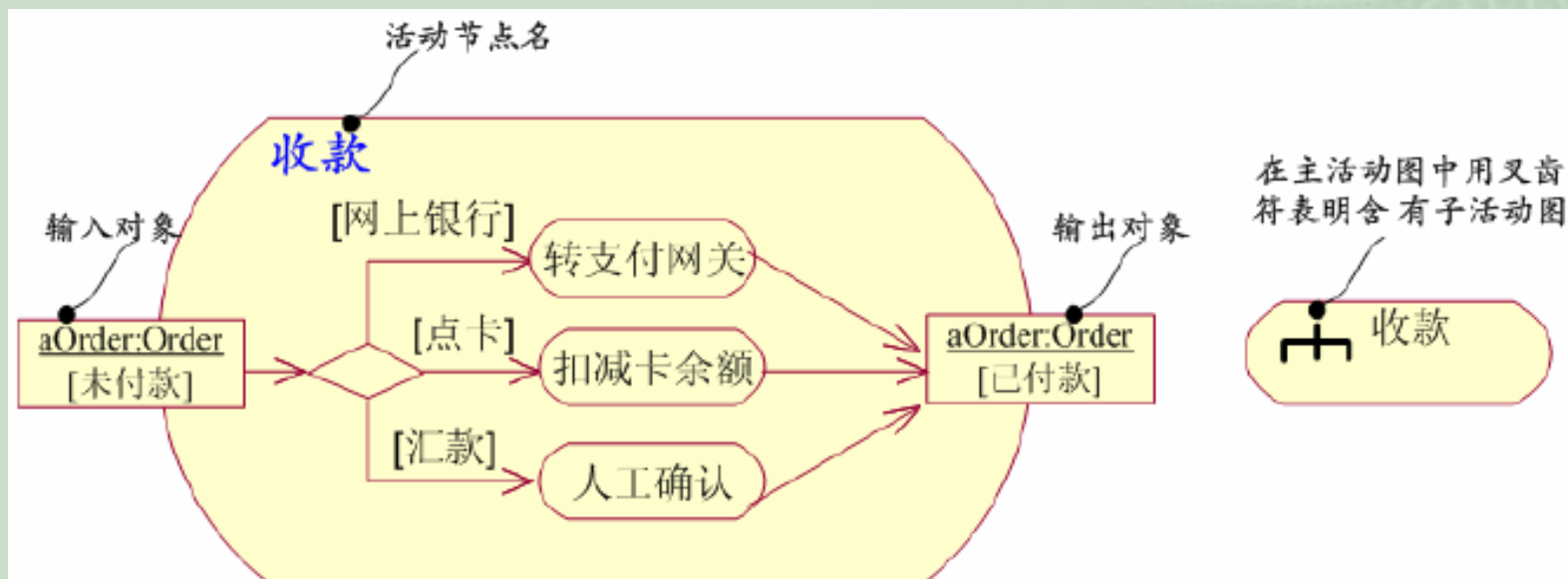
复杂活动图

- 辅助活动图
- 汇合描述
- 发送信号与接收信号
- 引脚
- 扩展区

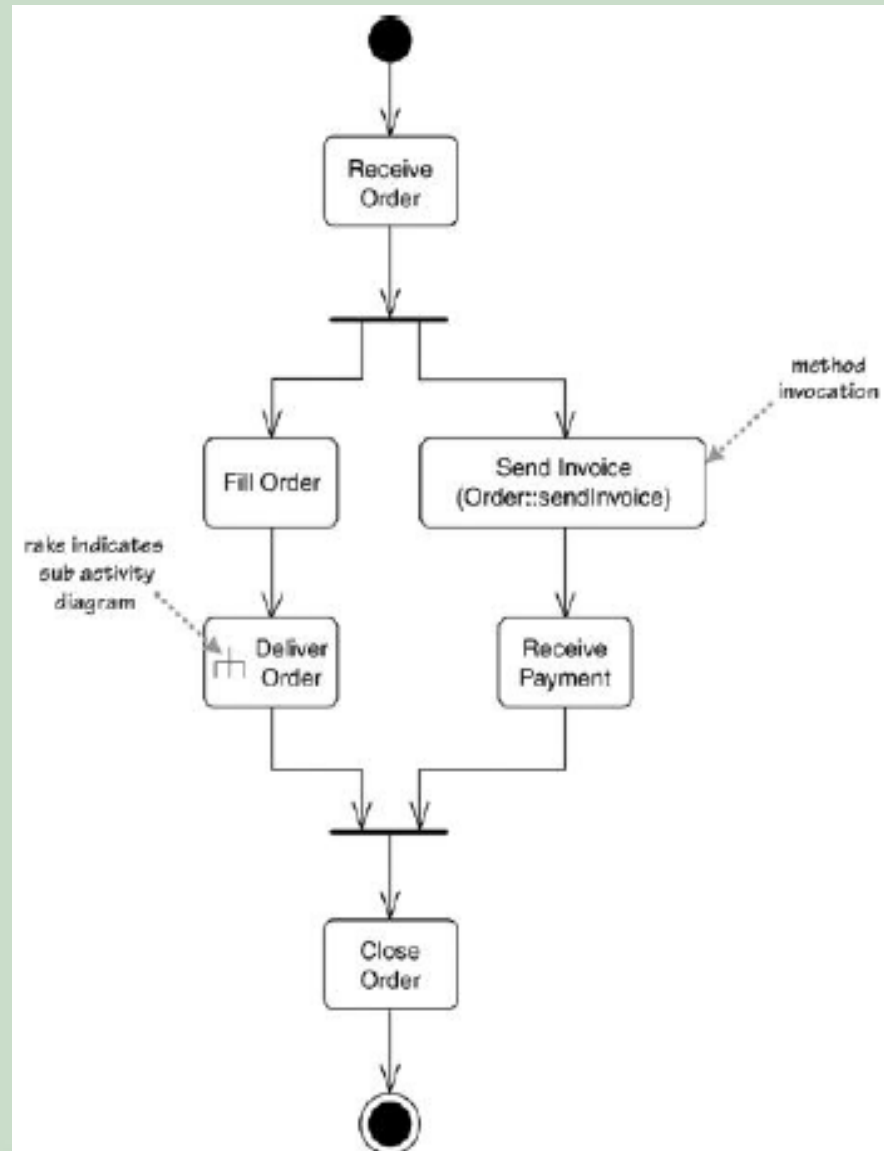
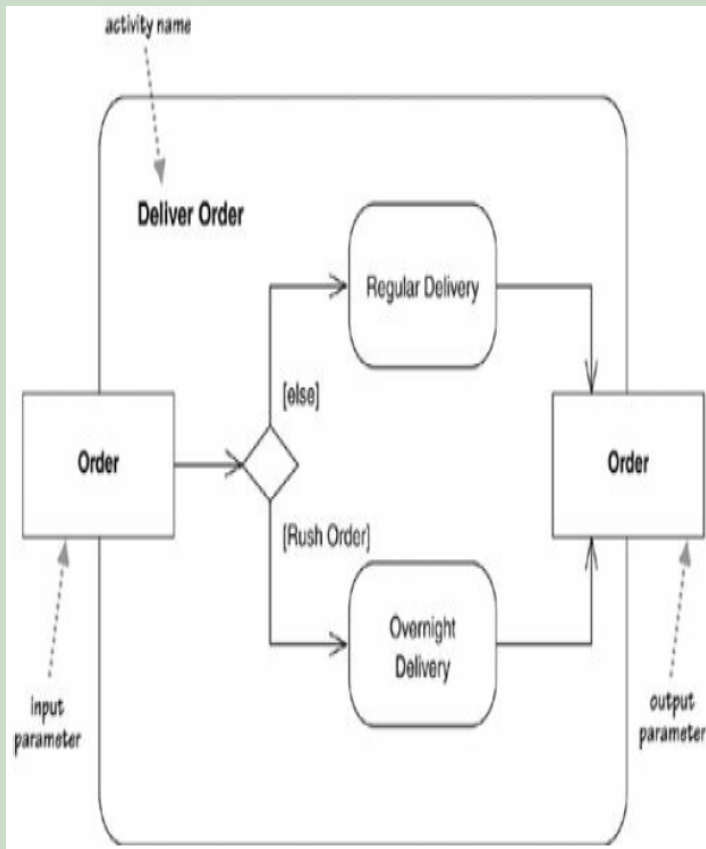


辅助活动图

- 活动图中包含的子活动图
- 当一个活动图很复杂，可以把其中的一组相关活动看做是一个子活动图，用子活动图的简图代替子活动图

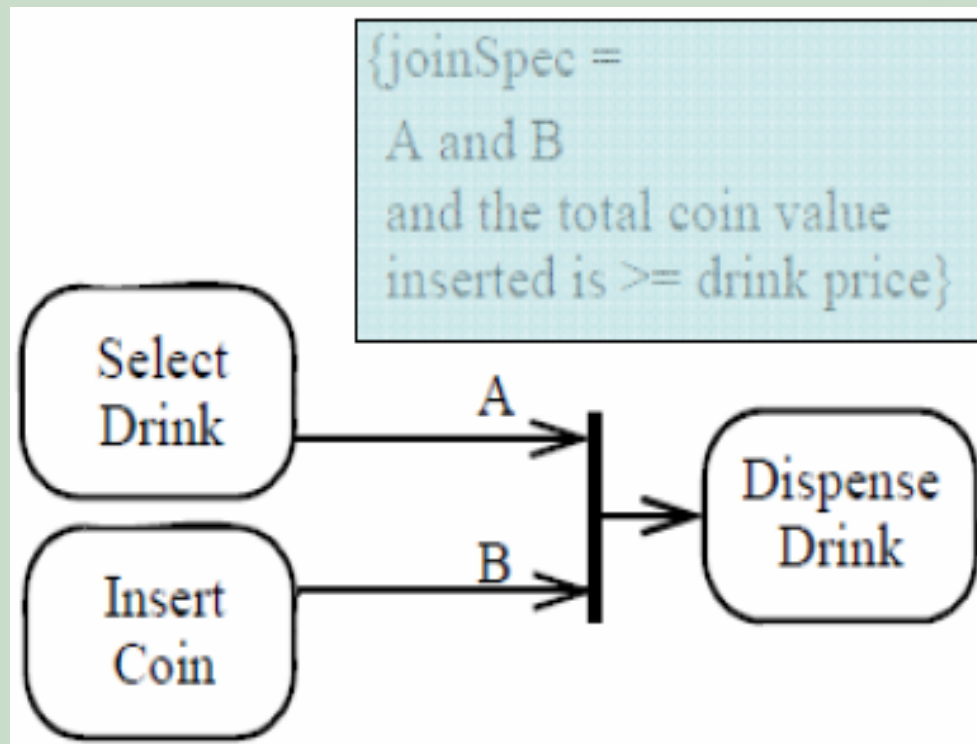


辅助活动图



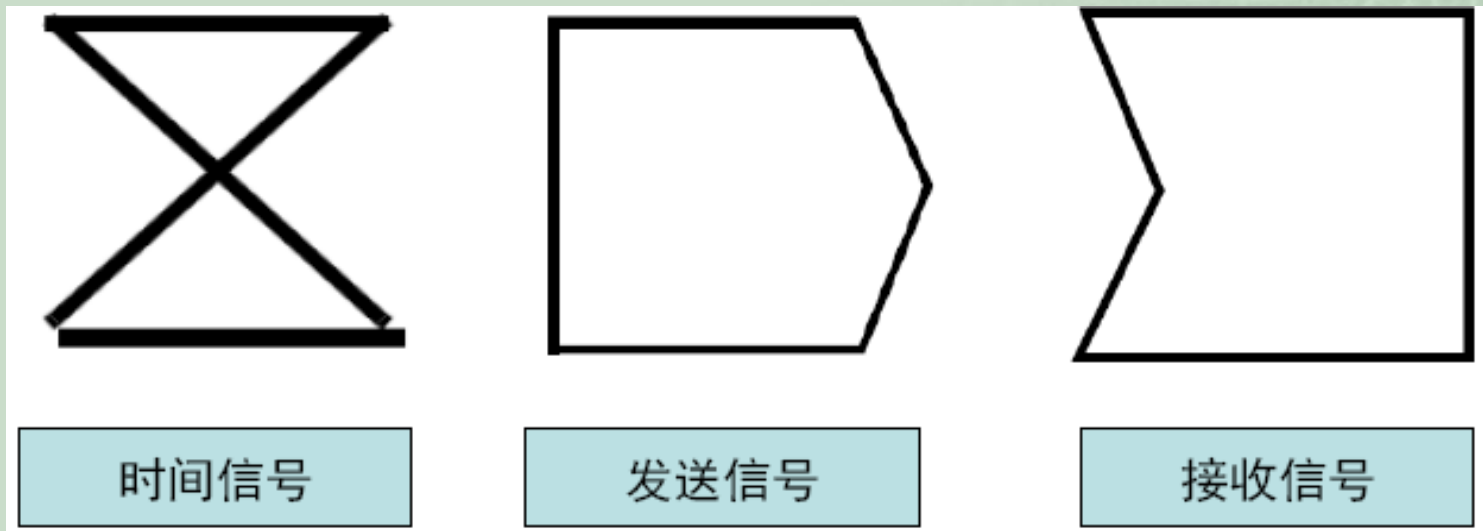
汇合描述 (join specification)

- 在汇合点，可以通过汇合描述对其加以约束

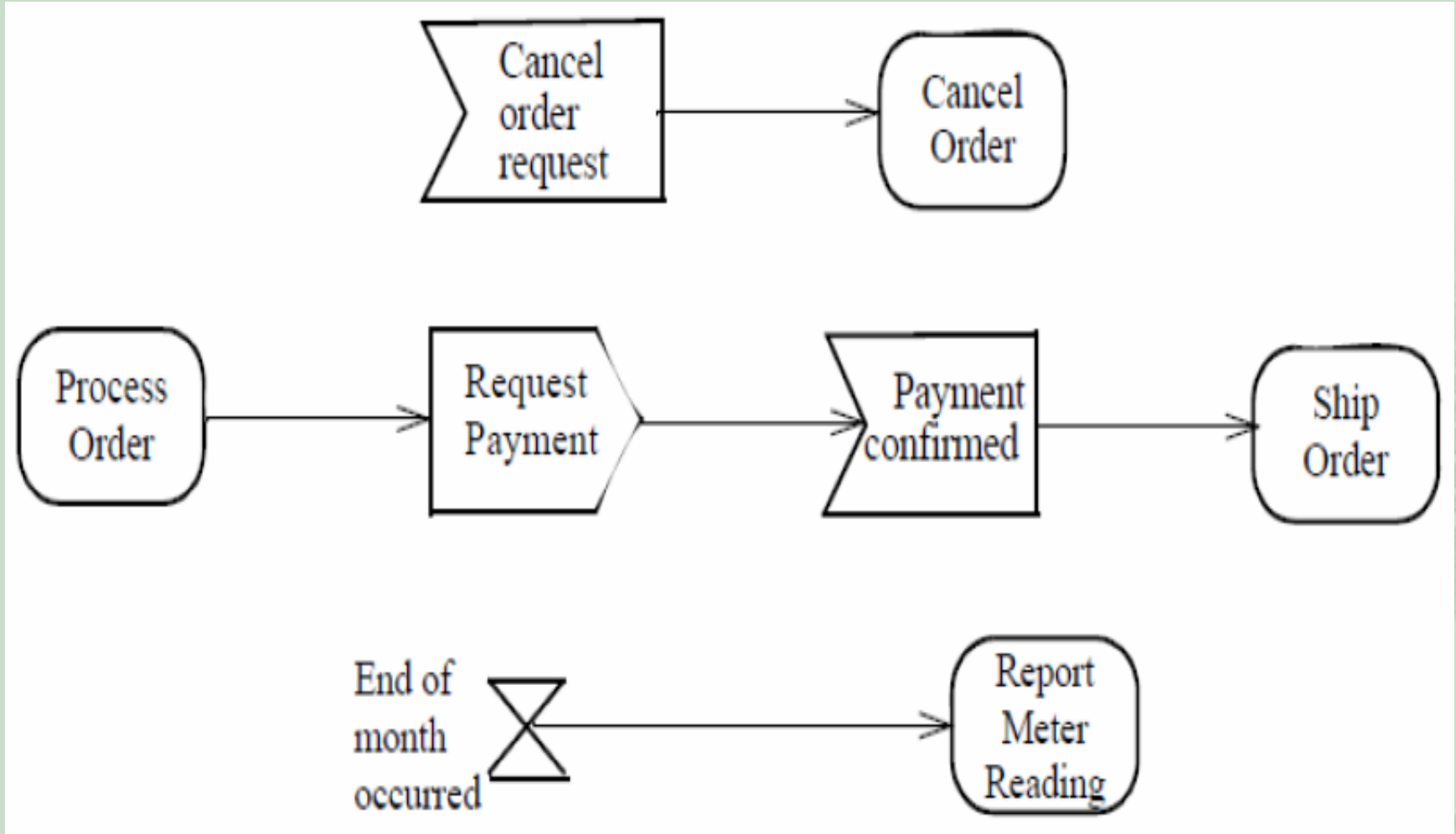


信号（signal action）

- 信号是表示两个对象之间异步通信信息包的方法，当一个对象接收到一个信号时，信号事件将发生

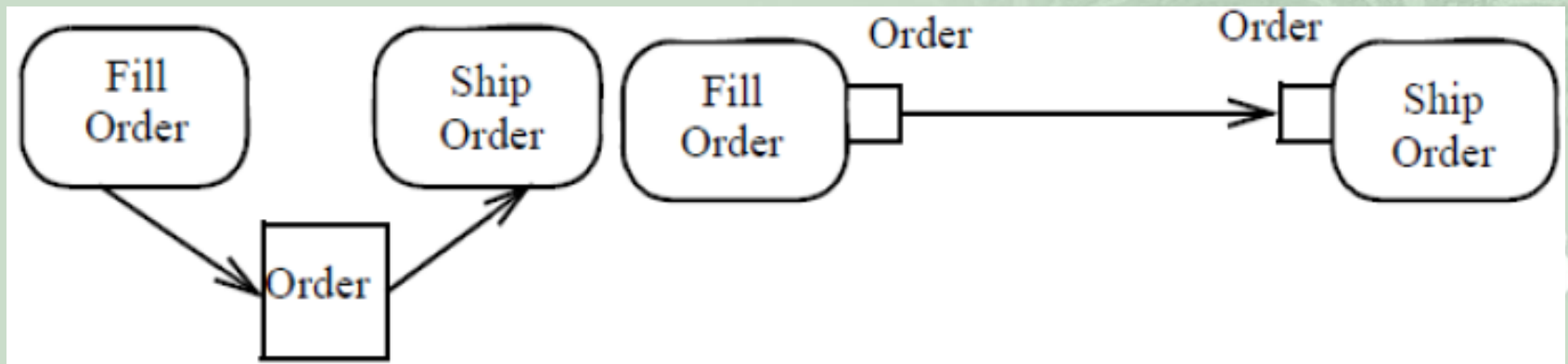


信号节点示例

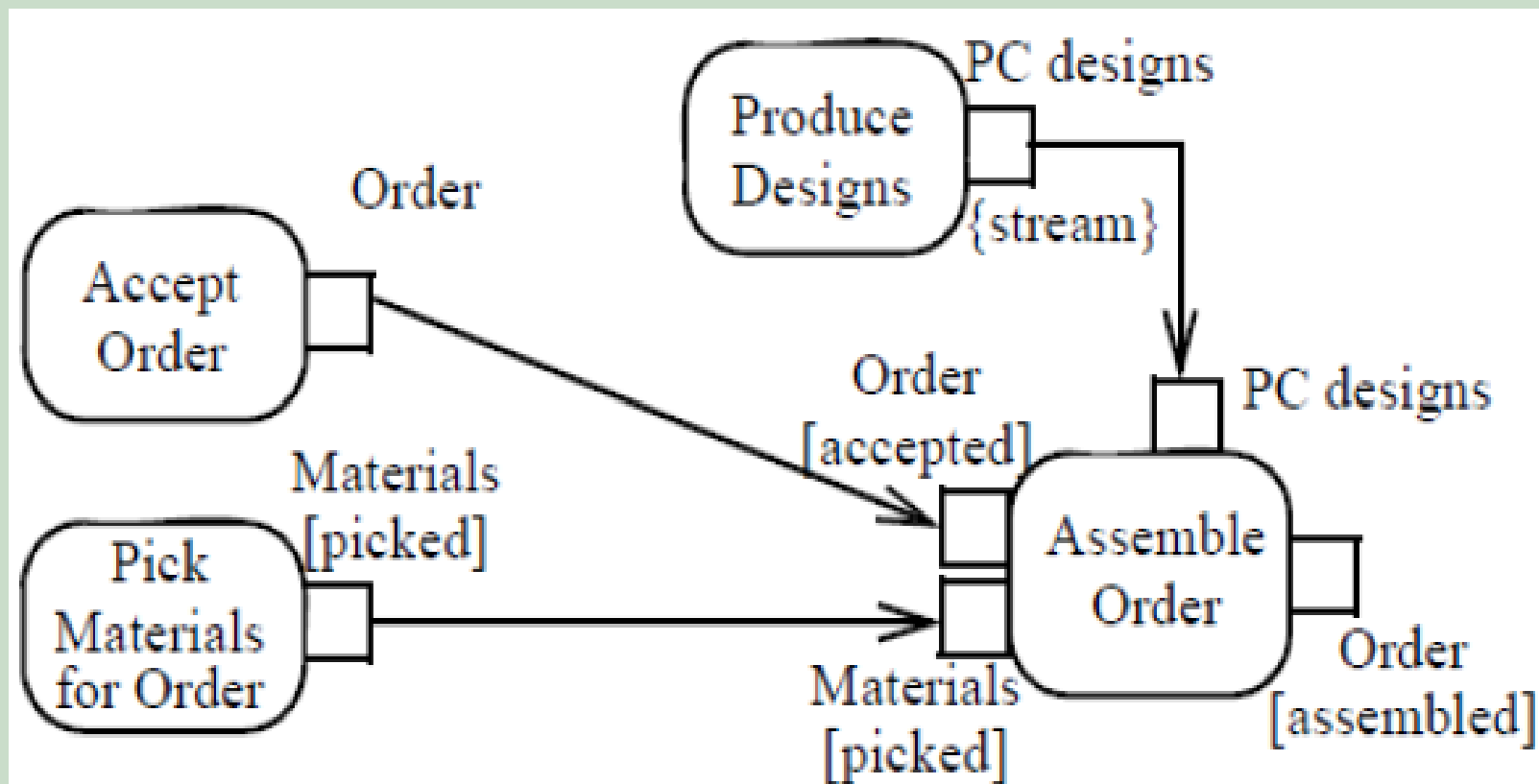


引脚 (pin)

- 可以用引脚可以表示活动节点所需的数据以及将要产生的数据
- 在UML中，引脚用小矩形表示，放在活动节点的外边界上
- 引脚是对象节点

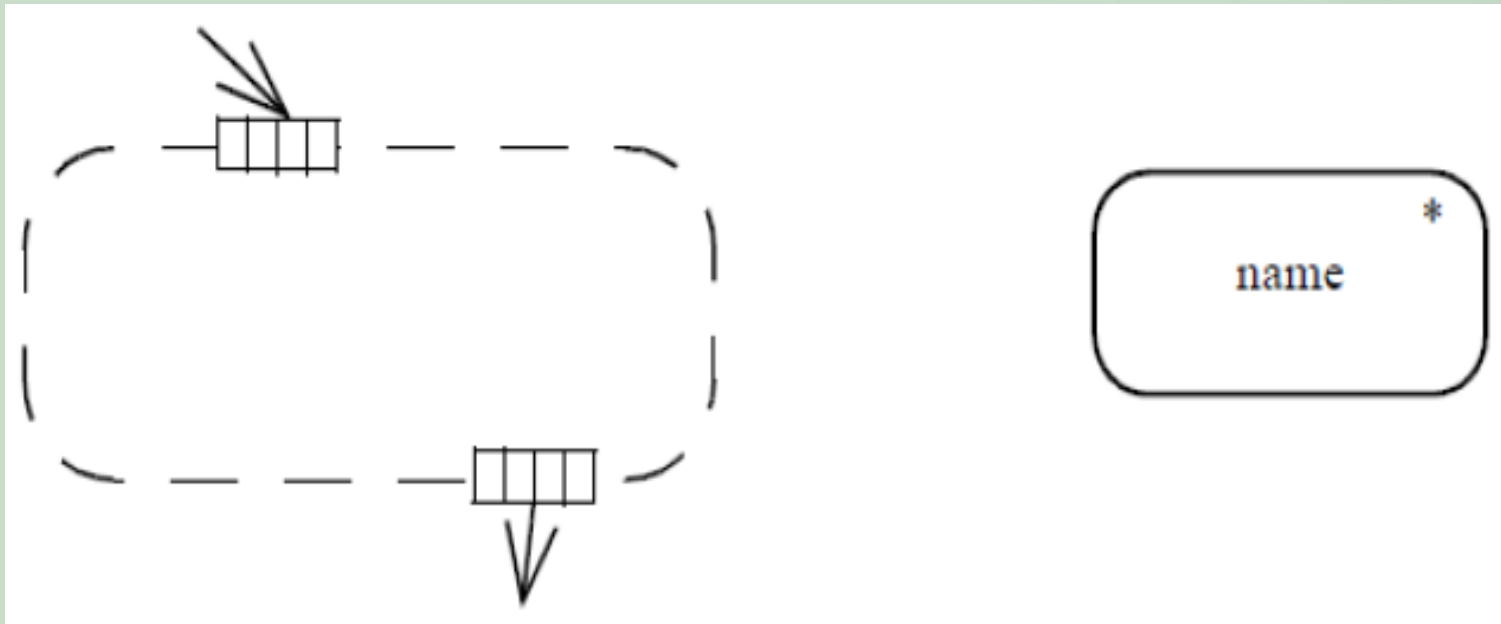


引脚示例

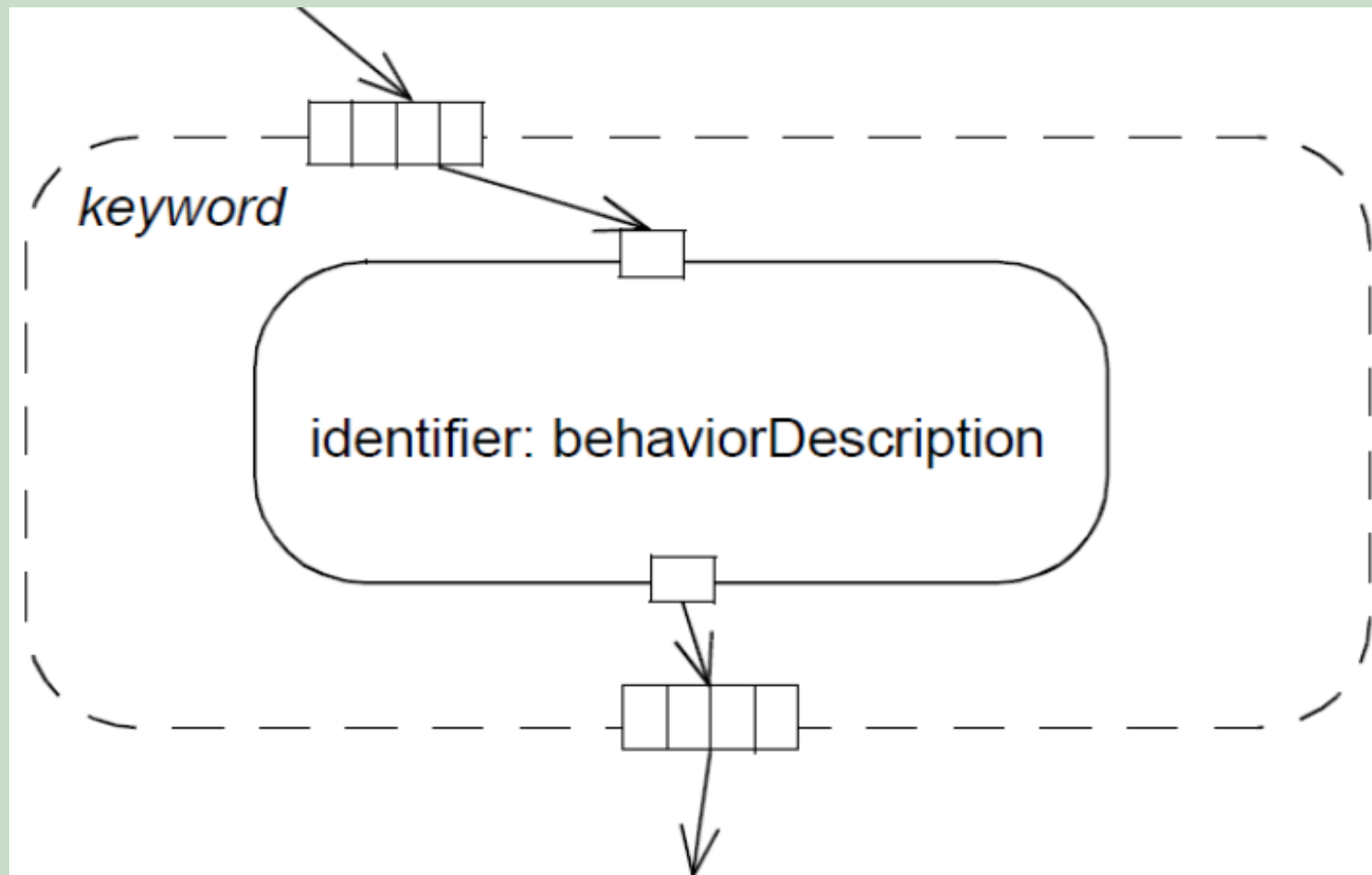


扩展区 (ExpansionRegion)

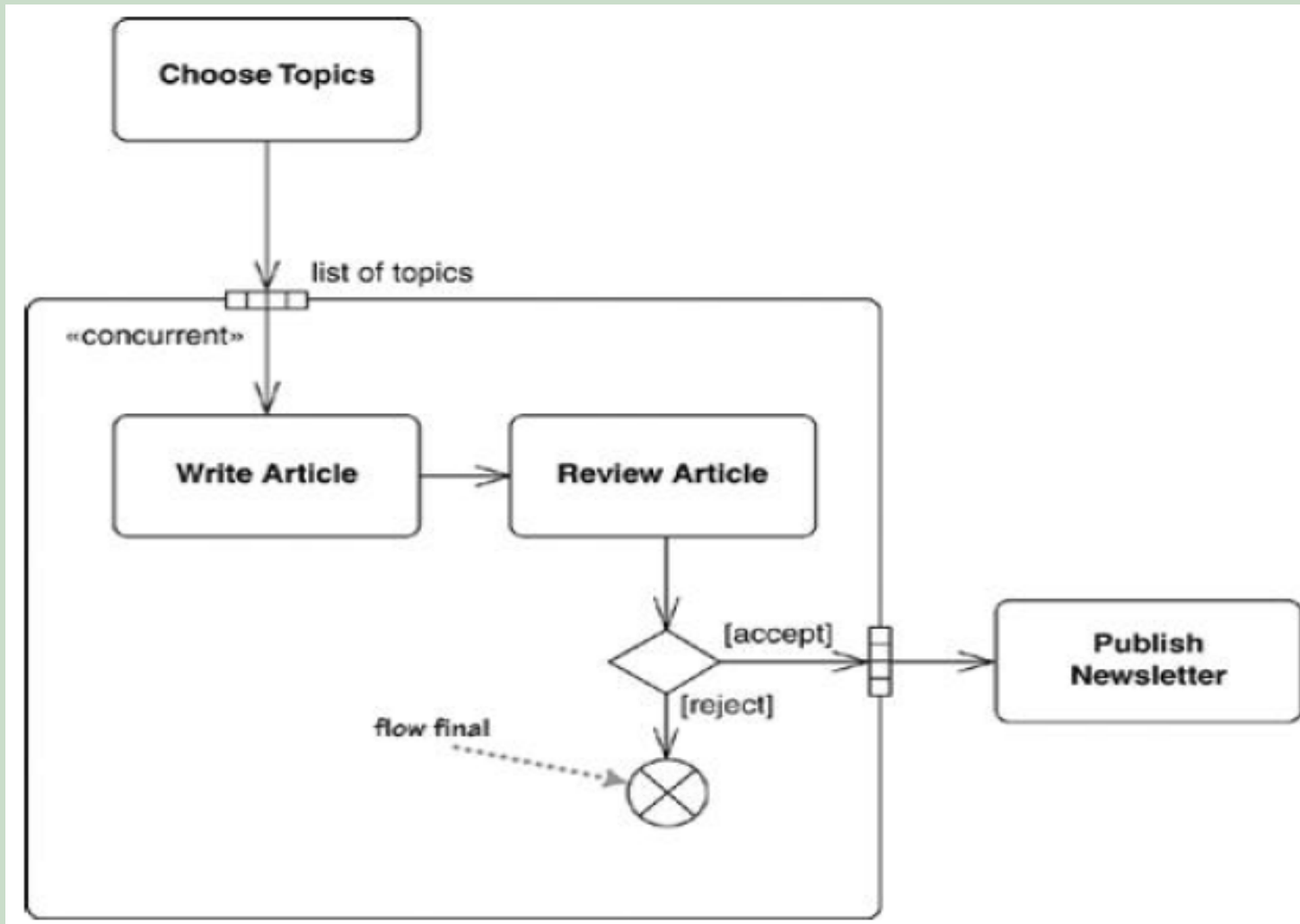
- 在需要表达一个活动需要多次执行的时候，可以采用扩展区



扩展区示例



扩展区示例



内容

- 概述
- 基本概念
- 建模方法



对 workflow 建模

- 为重要的职责部门创建泳道
- 从初始节点开始找出随时间推移发生的动作和活动，并标识为活动节点
- 将复杂的活动或多次出现的活动用子活动图的引用节点表示，并绘制子活动图
- 找出连接活动节点的顺序转换、分支、分岔和汇合
- 如果工作流中存在重要的对象，加入对象流
- 如果工作流中有重复执行的活动，则使用扩展区来表示循环活动

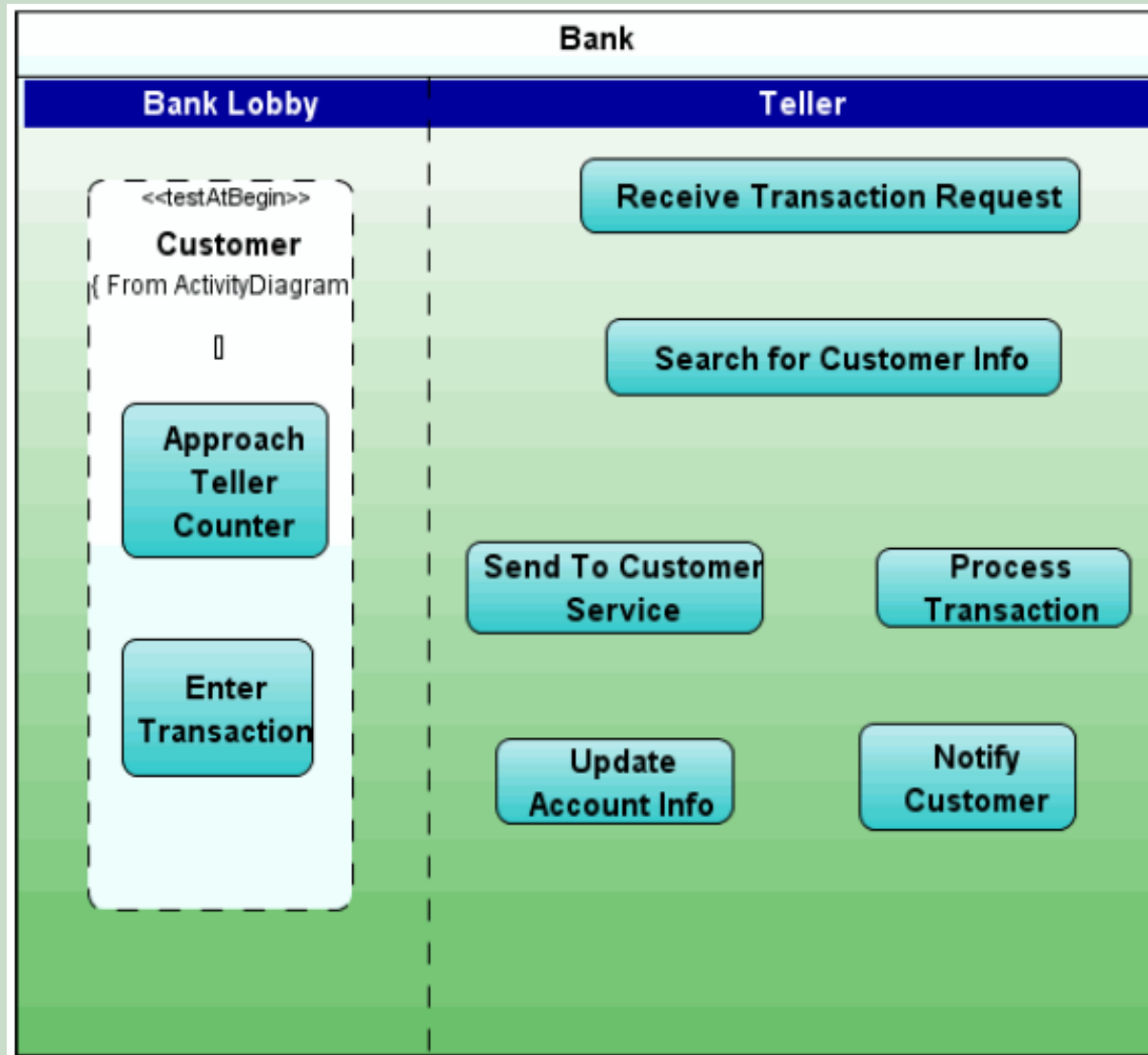
绘制活动图的注意点

- 尽量使用分支、分岔和汇合等基本建模元素绘制活动图
- 根据活动图中是否要体现出不同的实施者，来决定是否采用泳道
- 根据需要，加入对象流以及对象的状态变化，只选取重要的对象
- 根据需要，加入高级的建模元素（比如辅助活动图、汇合描述、信号节点、引脚及扩展区）

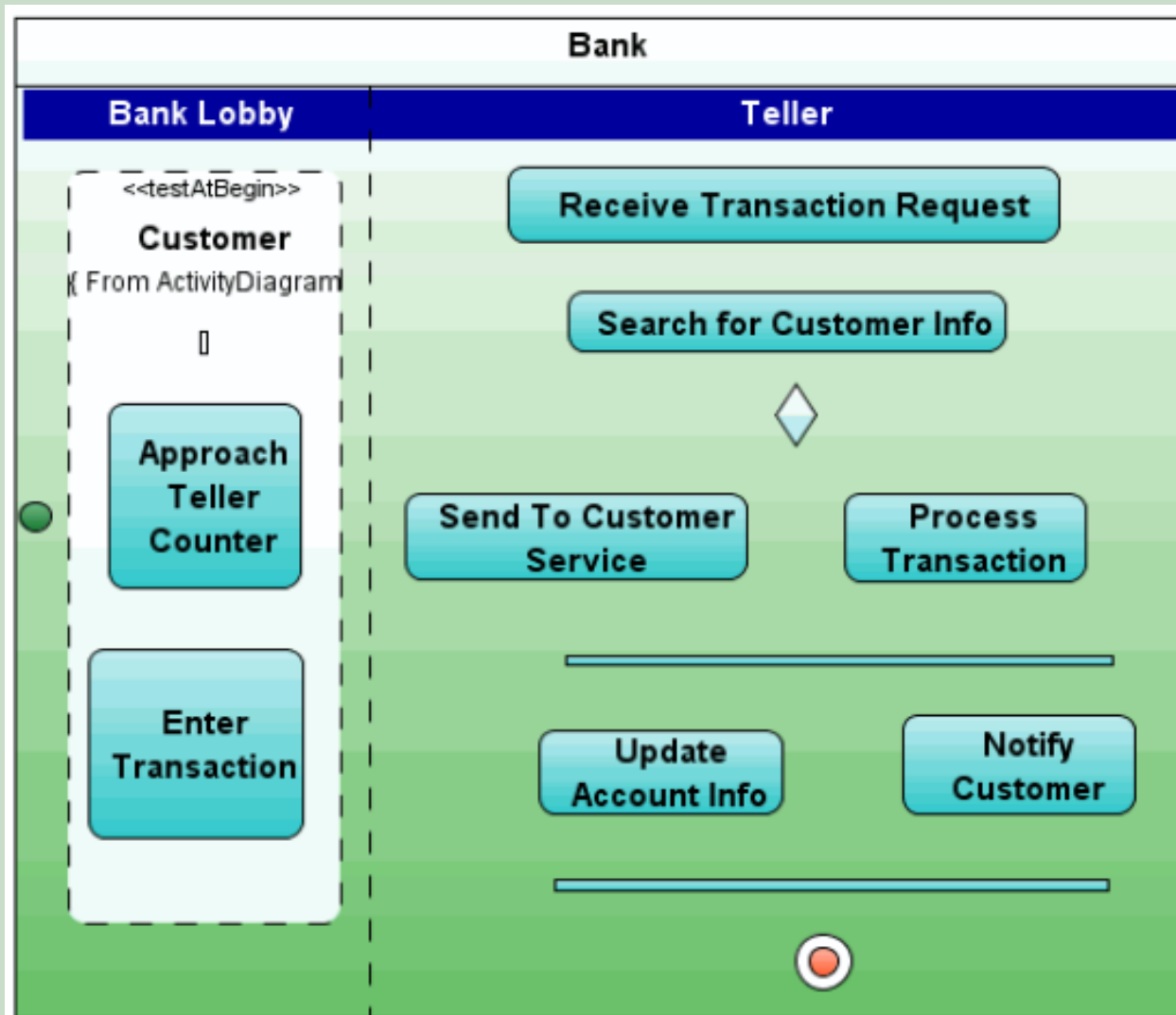
活动图建模示例



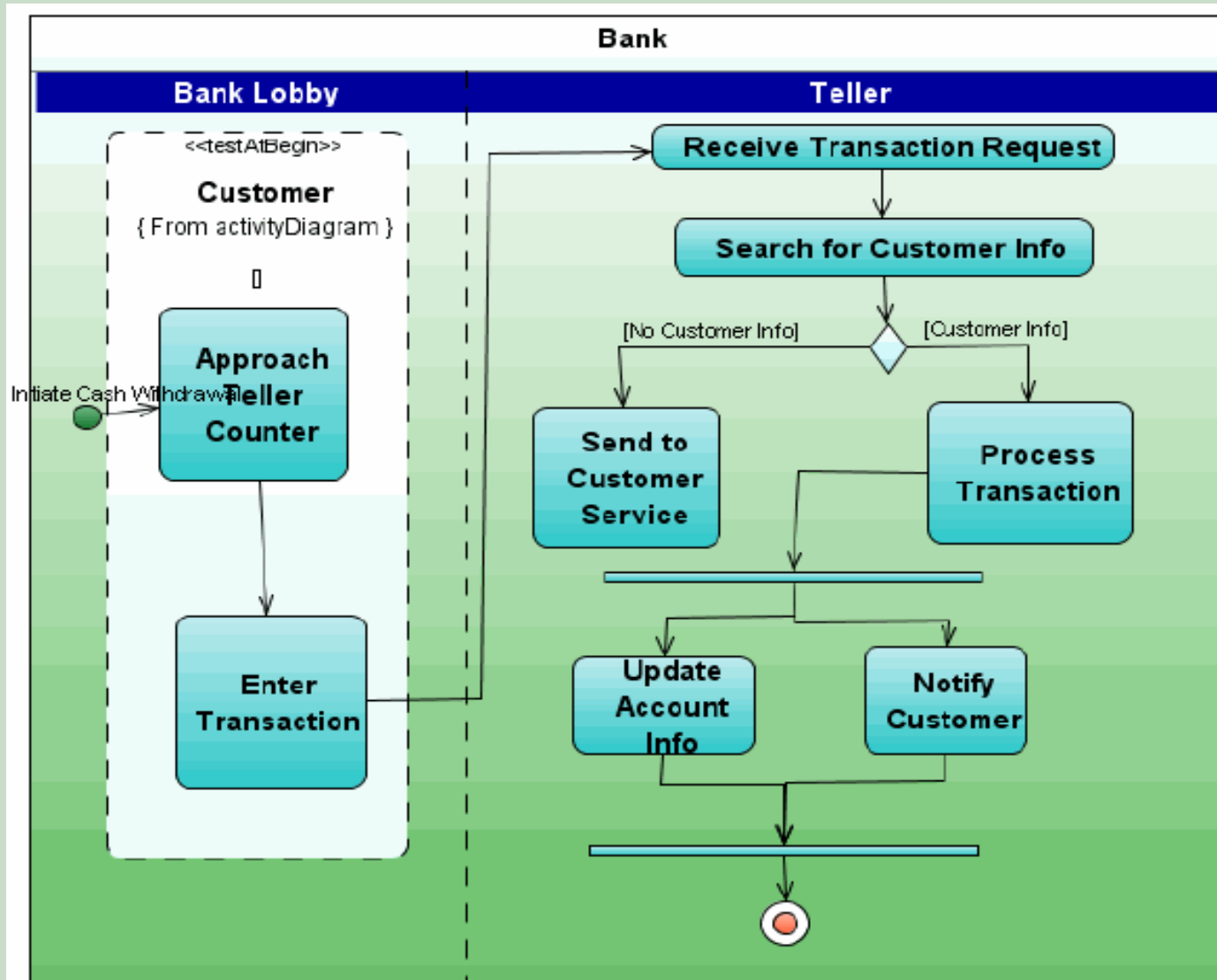
活动图建模示例



活动图建模示例



活动图建模示例



活动图建模风格

- 确保离开决策点每个转移都有一个监护条件
 - 这可以确保建模人员已经考虑到这个决策点的所有可能情况
- 确保决策点上的监护条件形成一个完备集
 - 不管何种情况，要一定能够从决策点离开
 - 比如，如果只有两个转移，像 $x < 0$ 和 $x > 0$ 这样的监护条件是不完备的，当 x 等于0时怎么办？
- 监护条件不要重叠
 - 离开决策点的转移上的监护条件应该保持一致
 - $x \geq 0$ 和 $x \leq 0$ 这两个监护条件不一致



活动图建模风格

- 确保每一个分岔都有对应的汇合
 - 当并行流完成的时候，总体的过程要从并行流结束的地方开始
- 确保每个分岔只有一个进入转移
 - 当发现需要几个转移到同一个分岔时，表明建模人员或者遗漏了一个活动，或者在该点事实上并不需要引入并发活动
- 确保每个汇合只有一个退出转移
 - 当某个汇合点需要多个退出转移时，表明仍然需要并发的活动，因此，把汇合点沿着总体活动过程向前移动