

第3章 活动图

(Activity diagram)

活动图的模型元素

◆ 活动图用来描述系统或者某个业务的处理流程

- 业务相关的工作流的描述
- 用例的事件流的描述
- 程序算法的描述

◆ 活动图由以下模型元素组成：

■ 活动

- 相当于构成业务和处理的一个单位。用圆角长方形来表示。

■ 控制流

- 用箭头表示，连接两个活动，表示活动的先后顺序。

■ 开始节点

- 用实心圆圈表示一个流程的开始。

■ 终了节点

- 用实心圆圈外加一圆圈来表示一个流程的结束。

■ 判断节点和监护条件

- 判断节点用菱形框来表示，根据判断结果的不同，处理流向不同的分支。
- 监护条件用来表示处理分支的条件。

■ fork节点

- 表示一个处理流程分割为多个并发流程。

■ join节点

- 表示对多个并发子流程进行同步。（※在UML2.0中，fork节点和join节点合称为同步条）

■ 参与者

- 在UML1.X中也称为泳道。

activity(活动)

- 活动代表工作流中的一个任务的执行。活动可以分解成更多的动作。

说明：

- Action 和activity的区别：
 - Action是原子的，而activity可进一步分解。

Swimlane (泳道)

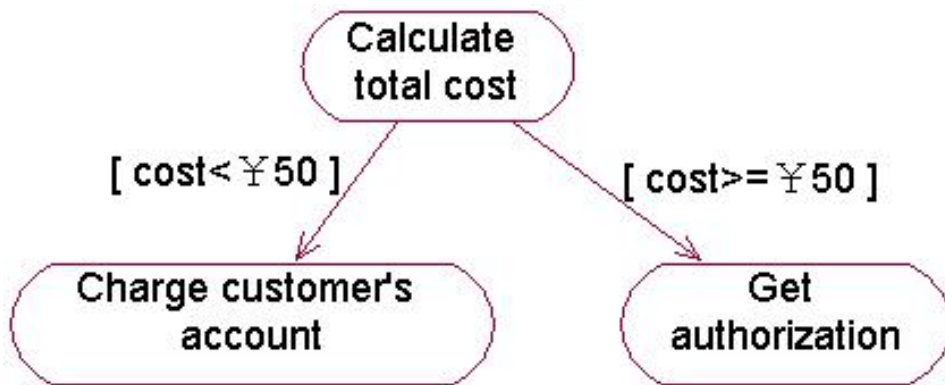
- 由组织内的某个人或者某些人执行的活动图。

说明：

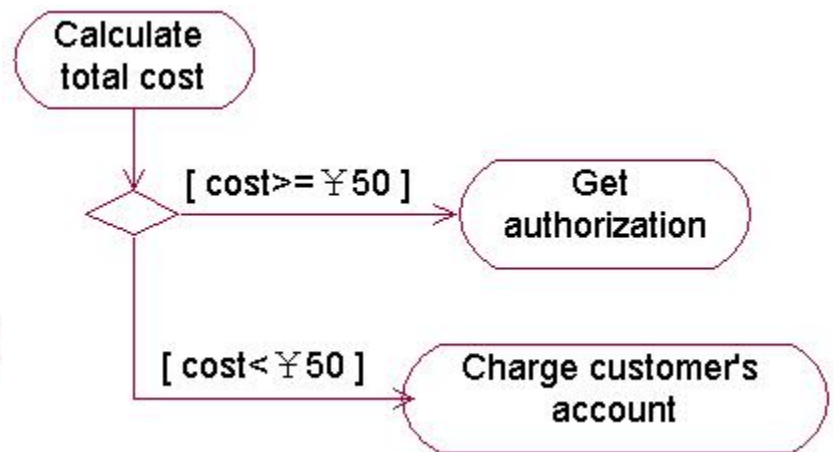
- 每个泳道代表一个责任区。

branch (分支)

- A **branch** is an element in a state machine in which a single trigger leads to more than one possible outcome, each with its own **guard condition**.
- 表示分支的两种方法：



方法1

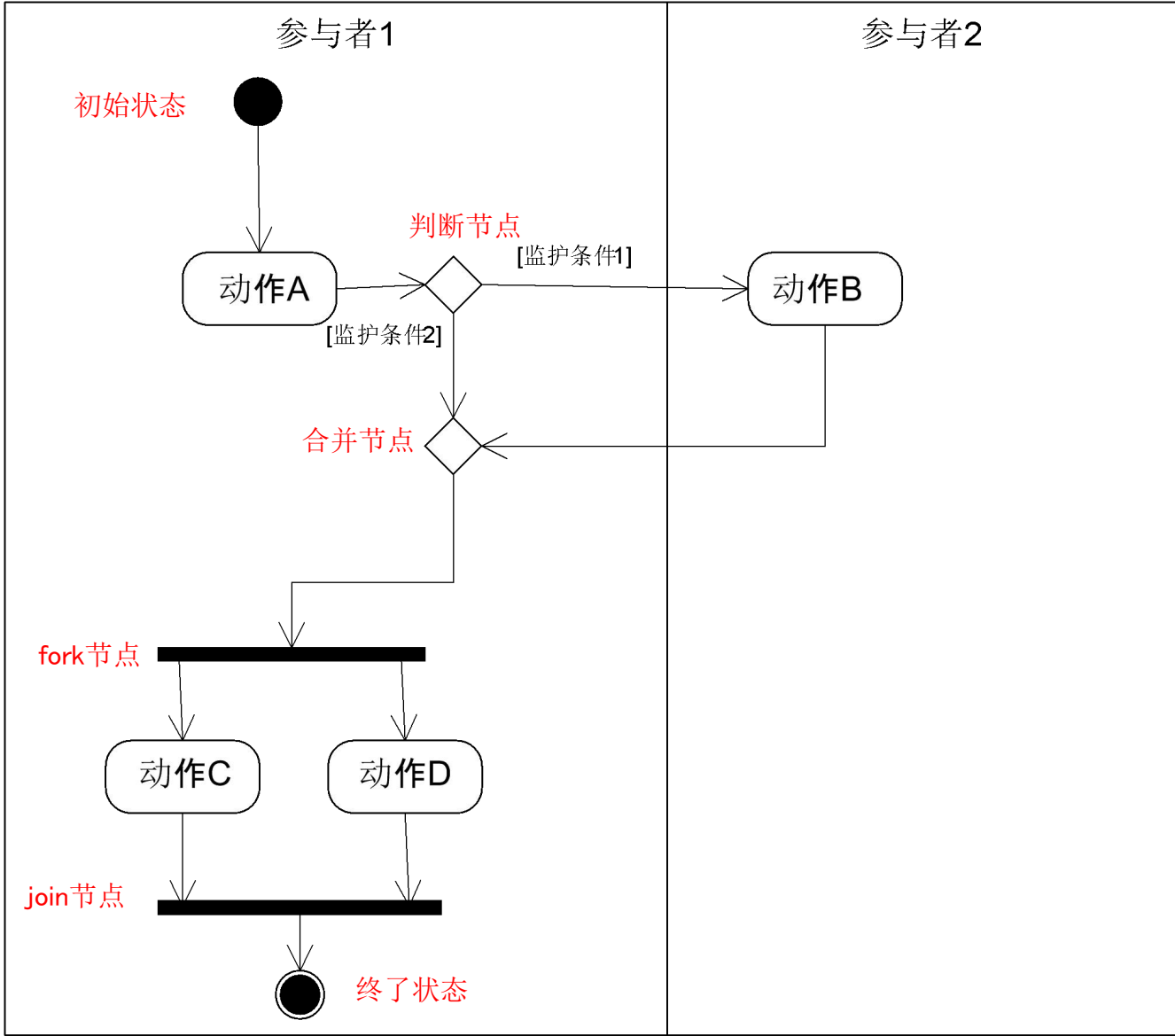


方法2

fork和join(分叉和汇合)

- A **fork** is a complex transition in which one source state is replaced by two or more target states, resulting in an increase in the number of active states.
- A **join** is a transition with two or more source states and one target state.

活动图的模型元素



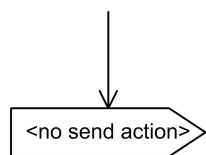
对象节点, 发送信号活动和接受信号活动

- ◆ 活动图中用对象节点来表示活动之间的数据交互

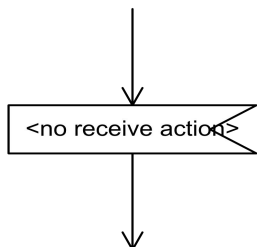


- ◆ (参考) 发送信号活动和事件受理活动

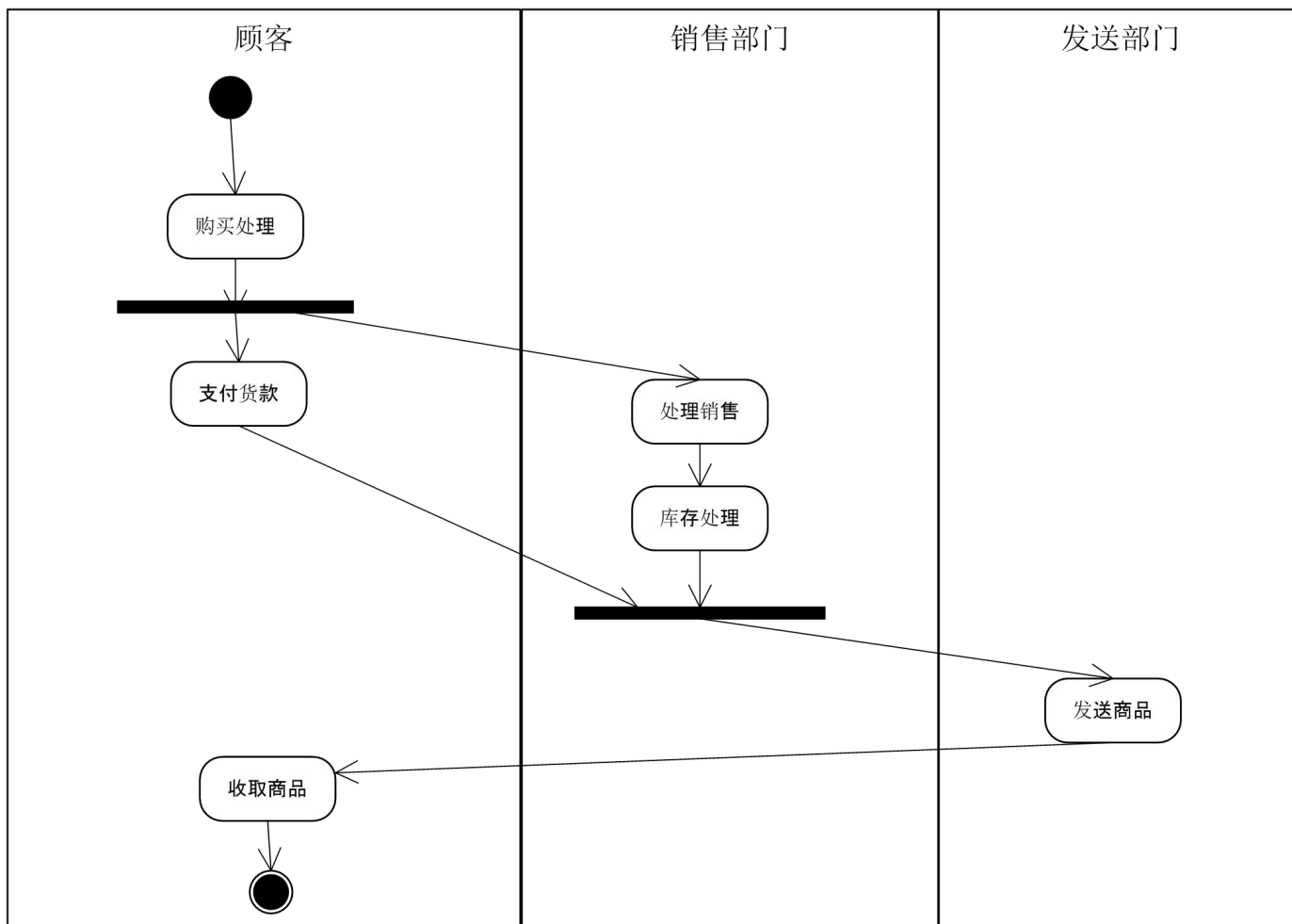
发送信号活动：表示的是让外部发生某种事件的活动。在活动图中如下表示：



事件受理活动：表示的是对于外部发生的事件进行接收的活动。在活动图中如下表示：



活动图的例子



绘制活动图的要点

■整理出场人物，抽取出角色

- ◆角色是现在的业务中出现的人、组织、系统等。
- ◆角色对应着活动图中的活动分区(Activity Partition)。

■整理角色间交换的信息

- ◆这些信息对应着活动图里的对象节点(Object Node)。

■找出开始节点和终止节点

- ◆一个活动图只有一个开始节点。
- ◆如果存在多个分支，可能会有多个终止节点。

■找出各个角色的活动

- ◆分析业务描述中的动词，获得各个角色的活动。

■做成活动图

- ◆信息、开始/终止节点、活动放置到合适的地方，并连接起来。

绘制活动图的例子

■图书馆借书的业务说明

- ◆读者来图书馆借书，可能先查询书库的图书记录。查询可以按书名、作者、图书编号、关键字查询。查询有两种结果，如果查到则记下书号，交给工作人员，然后等候办理借书手续。如果该书已经被全部借出，则可做借书登记，等待有书时被通知。如果图书馆没有该书的记录，则做缺书登记。
- ◆办理借书手续时先要出示图书证，没有图书证则去申请图书证。如果借书数量超出规定，则提示“借书数量超限，不能继续借阅”。工作人员登记借阅人信息、借阅的图书信息、借出时间和应还书时间。系统自动修改书库的图书记录、读者库信息。

绘制活动图的例子

■整理出场人物，抽取出角色

- ◆读者

- ◆工作人员

- ◆系统

■整理角色间交换的信息

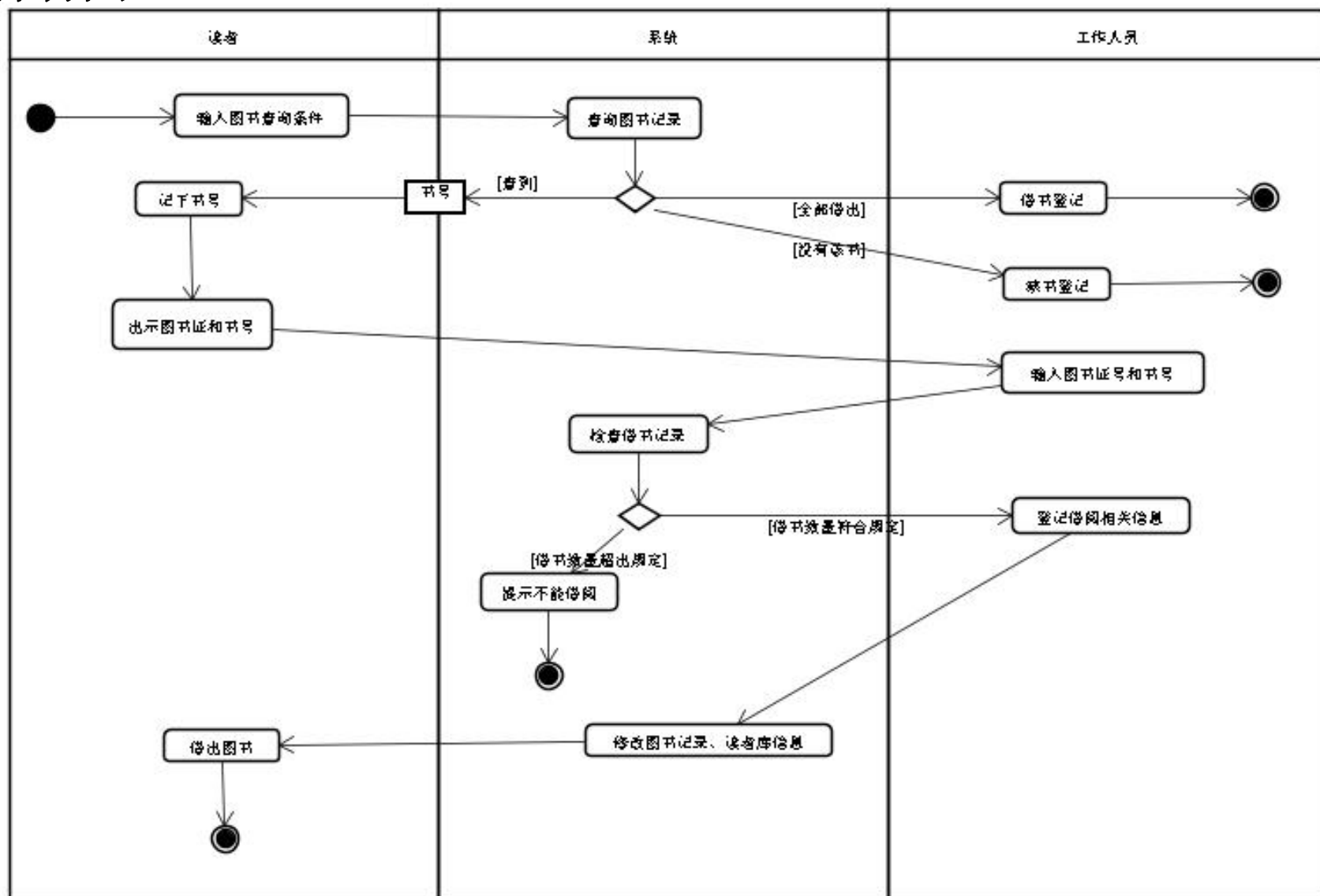
- ◆书名、作者、图书编号、关键字

- ◆书号

- ◆图书证

绘制活动图的例子

■ 活动图



活动图与状态图比较

1. 活动图 and 状态图描述的重点不同：
 - 活动图描述的是从activity到activity的控制流，而状态图描述的是对象的状态及状态之间的转移。
2. 活动图 and 状态图使用的场合不同：
 - 对于以下几种情况可以使用活动图：
 - 分析用例
 - 理解涉及多个用例的工作流
 - 处理多线程应用
 - 对于下面的情况要使用状态图：
 - 显示一个对象在其生命周期内的行为。

说明：如果要显示多个对象之间的交互情况，可用顺序图或协作图。

活动图建模风格

- 建模风格1：确保从决策点出来的每个转移都有一个警戒条件。
 - 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。
- 建模风格2：确保决策点上的警戒条件形成一个完备集。
 - 不管什么情况，要一定能够可以从决策点离开。
 - 例如，像 $x < 0$ 和 $x > 0$ 这样的警戒条件是不完备的，因为当 x 等于0时会发生什么并不清楚。
- 建模风格3：警戒条件不要重叠。

- 建模风格4：确保每个分叉只有一个进入转移。
 - 当发现需要几个转移到同一个分叉时，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并发活动。
- 建模风格5：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。

- 建模风格6：要小于5条泳道。
 - 泳道的一个缺点是使得建模人员无法自由地在图中排列活动以节省空间，因此会增加图的大小。泳道数目越多，这个问题越严重。

总结

◆ 活动图

- 可用来表示业务流程，用例图中某个用例的事件流，或者程序的算法描述。