

状态图和活动图



内容提纲

- 状态图
 - 相关概念
 - 实例讲解
- 活动图
 - 相关概念
 - 实例讲解



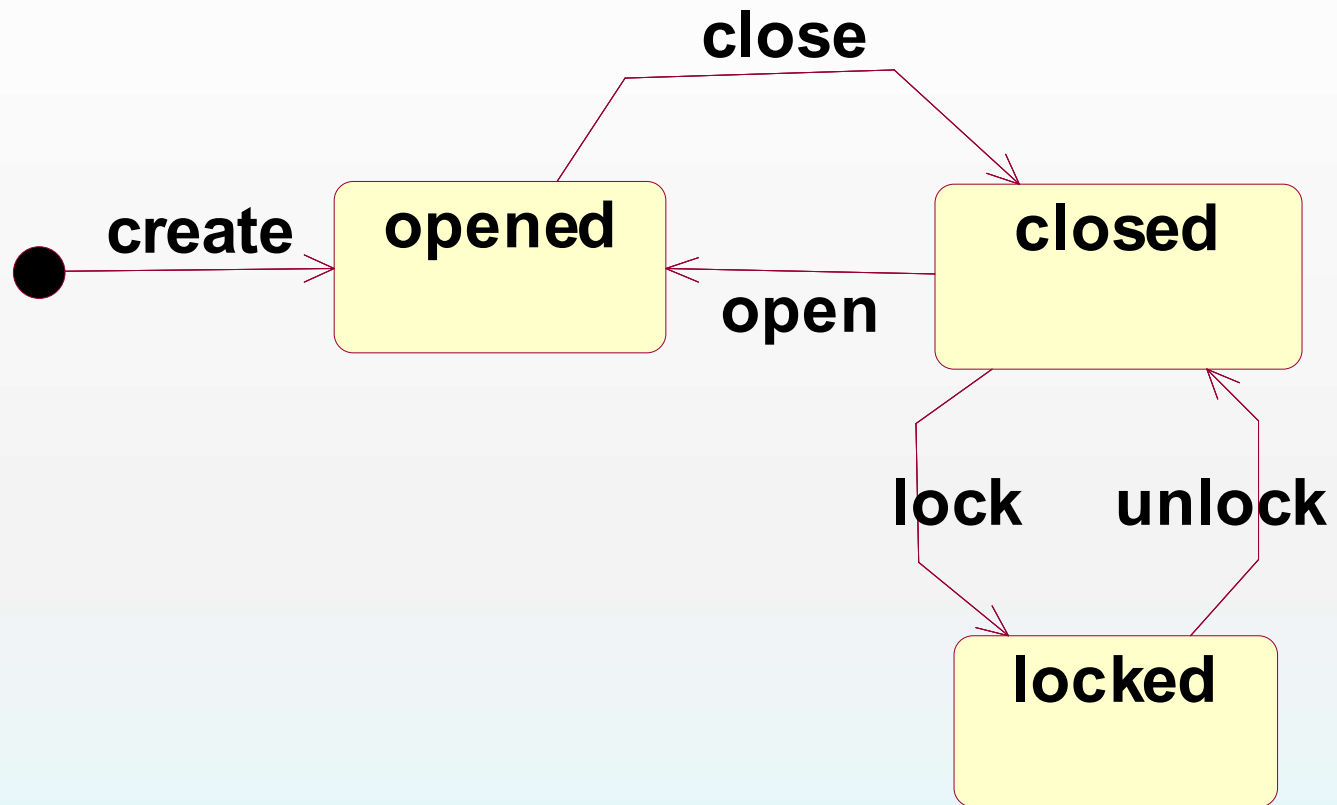
什么是状态图

状态图(Statechart diagram): 用于描述一个对象在其生存周期间的动态行为, 表现一个对象所经历的状态序列, 引起状态转移的事件(event), 以及因状态转移而伴随的动作(action).

对象: 类的实例、用例的实例、整个系统的实例



门的状态

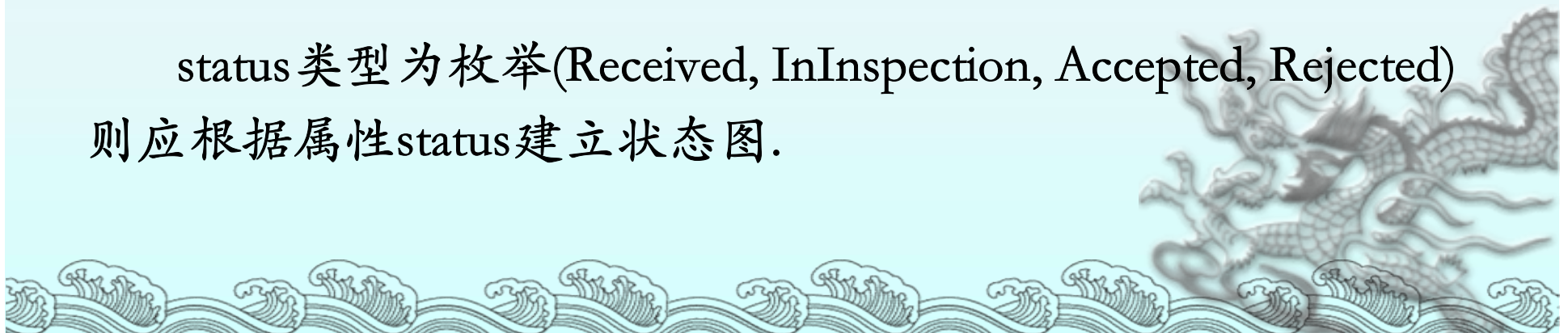


状态图描述的对象往往有 **多个属性**, 应该在具有如下特性的属性上建模:

- ◆ 属性拥有 **较少的可能取值**
- ◆ 属性在这些值之间的 **转移** 有一定的 **限制**

如: 类SellableItem中有2个属性: salePrice和status, 其中:
salePrice类型为Money, 取值范围为正实数;

status类型为枚举(Received, InInspection, Accepted, Rejected)
则应根据属性status建立状态图.



状态图的基本元素

状态, state

组合状态, composite state

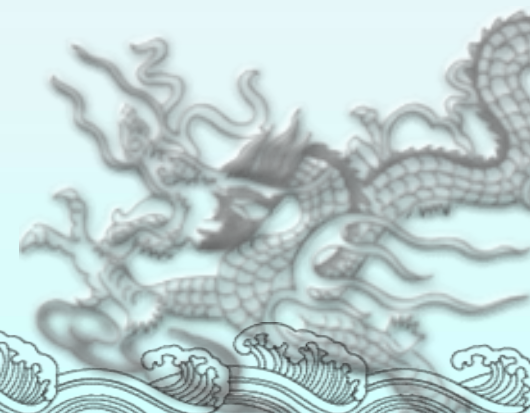
子状态, sub state

历史状态, history state

转移, transition

事件, event

动作, action

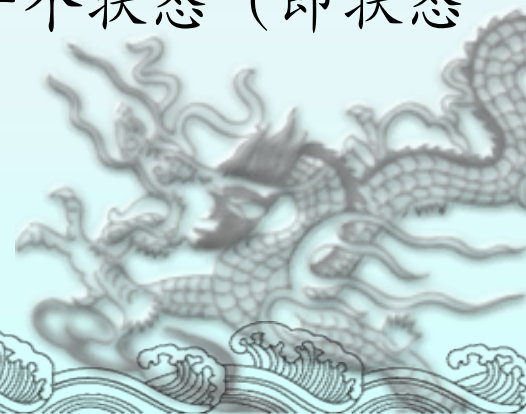


什么是状态 (state)

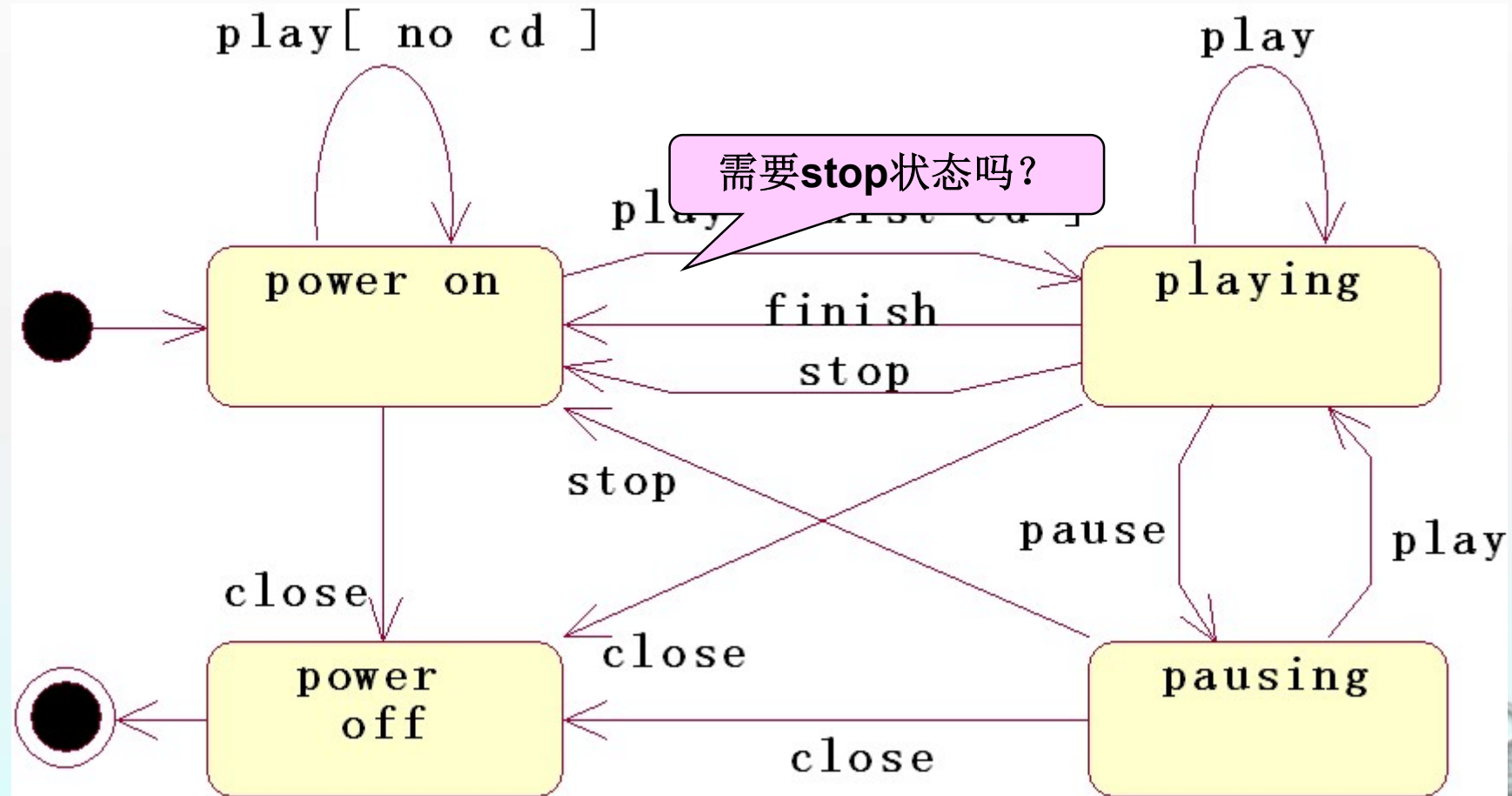
状态是在对象的生命期中的某个条件或状况,在此期间对象将满足某些条件、执行某些活动或等待某些事件.

◇ 理解

- ◆ 对象在任何时候都会处于某种状态中,所有对象都有状态
- ◆ 对象所处的状态决定了它如何响应所检测到的事件或所接收的消息。
- ◆ 通常,事件使对象从一个状态转向另一个状态 (即状态的转换)



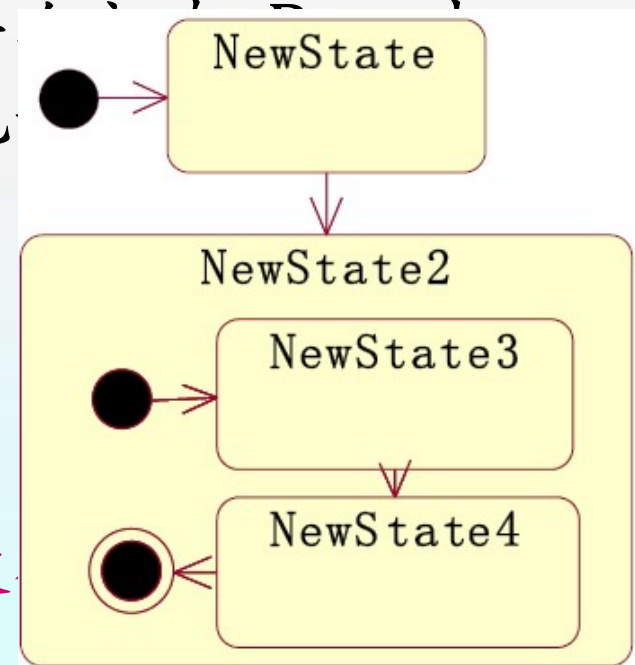
CD Player



初态和终态

初态 start state

- ◆ 显示状态图中状态机执行（execution）的开始
- ◆ 一个状态机**只能有一个**初态，因为每一个执行都是从相同的地方开始的。如果一个状态机用多张状态图描述，则多张图用一只允许画一个初态，其它地方拖
- ◆ 嵌套状态中可以使用新的初态



终态 end state

- ◆ 表示最后的或者终端状态；
- ◆ 终态**数目可以不确定，也可以没**

动作

动作

- ◇ 可执行的原子计算
- ◇ 不可中断，其执行时间可忽略不计
- ◇ 当转换被引发时，它对应的动作被执行。它一般是一个简短的处理过程。



动作分为入口动作和出口动作。

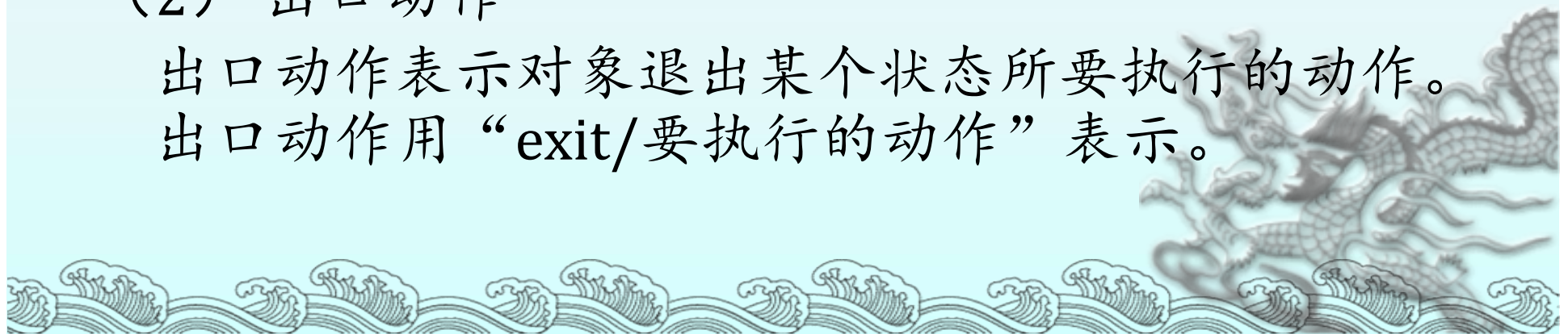
当转换发生时，进入某个状态时发生的动作称为入口动作；离开某个状态时发生的动作称为出口动作。

(1) 入口动作

入口动作表示对象进入某个状态所要执行的动作。
入口动作作用“entry/要执行的动作”表示。

(2) 出口动作

出口动作表示对象退出某个状态所要执行的动作。
出口动作作用“exit/要执行的动作”表示。



活动

- ◆ 当对象处于一个状态时，它一般是空闲的，在等待一个事件的发生。但是某些时间，对象正在执行一序列动作，即，对象做着某些工作，并一直继续到被某个外部事件的到来，才中断这些工作，我们把对象处于某个状态时进行的一序列动作称为活动。
- ◆ 如果对象处于某个状态进行一些动作，可能会需要一些时间，我们可以用活动来描述这一序列动作。表示活动的方法是，在状态的转换分栏中添加一行“活动描述”，其格式为“do/动作名”。
- ◆ 活动用来描述对象处于某个状态时，对象进行的一序列动作。



中间状态

中间状态的组成（除初态终态外，最常见的状态）

- ◇ 状态名（name）

- ◆ 是可以把该状态和其他状态区分开的字符串;状态也可能是匿名的，即没有名称。

- ◇ 入口/出口动作（entry/exit action）

- ◇ 内部转移（internal transition）

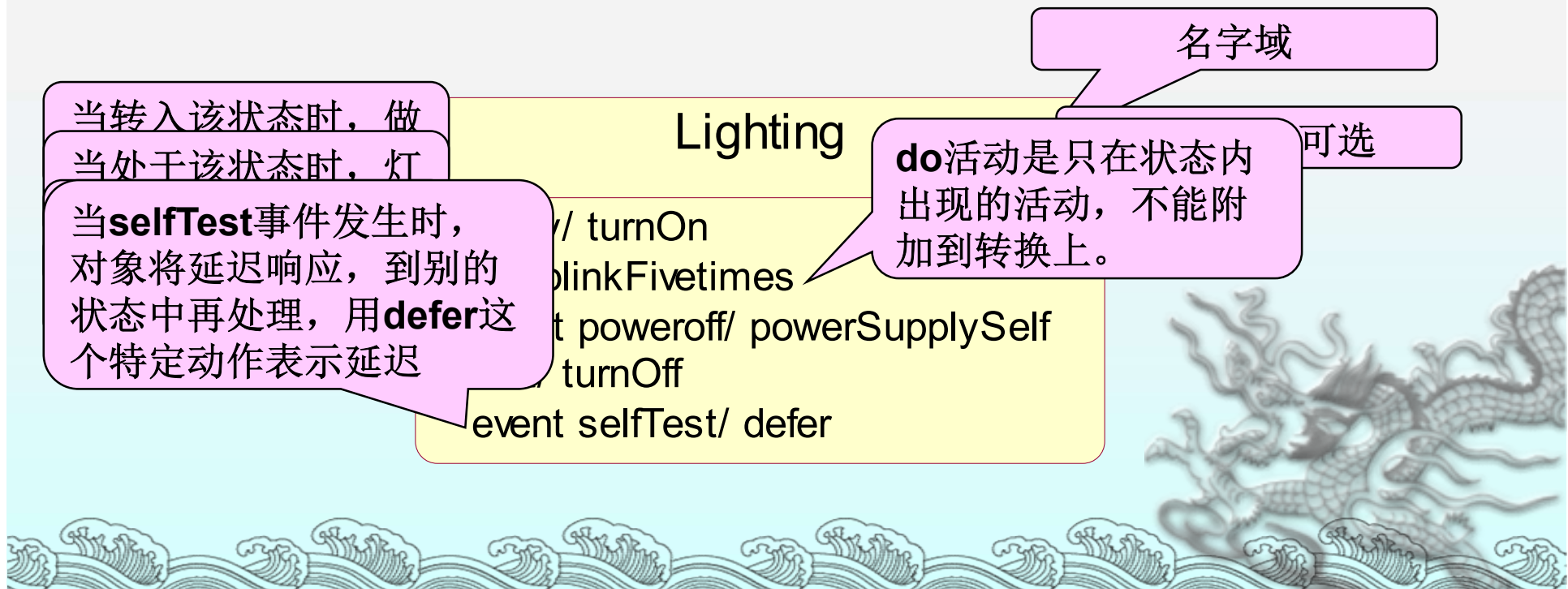


◇ 子状态 (substate)

- ◆ 嵌套在另外一个状态中的状态

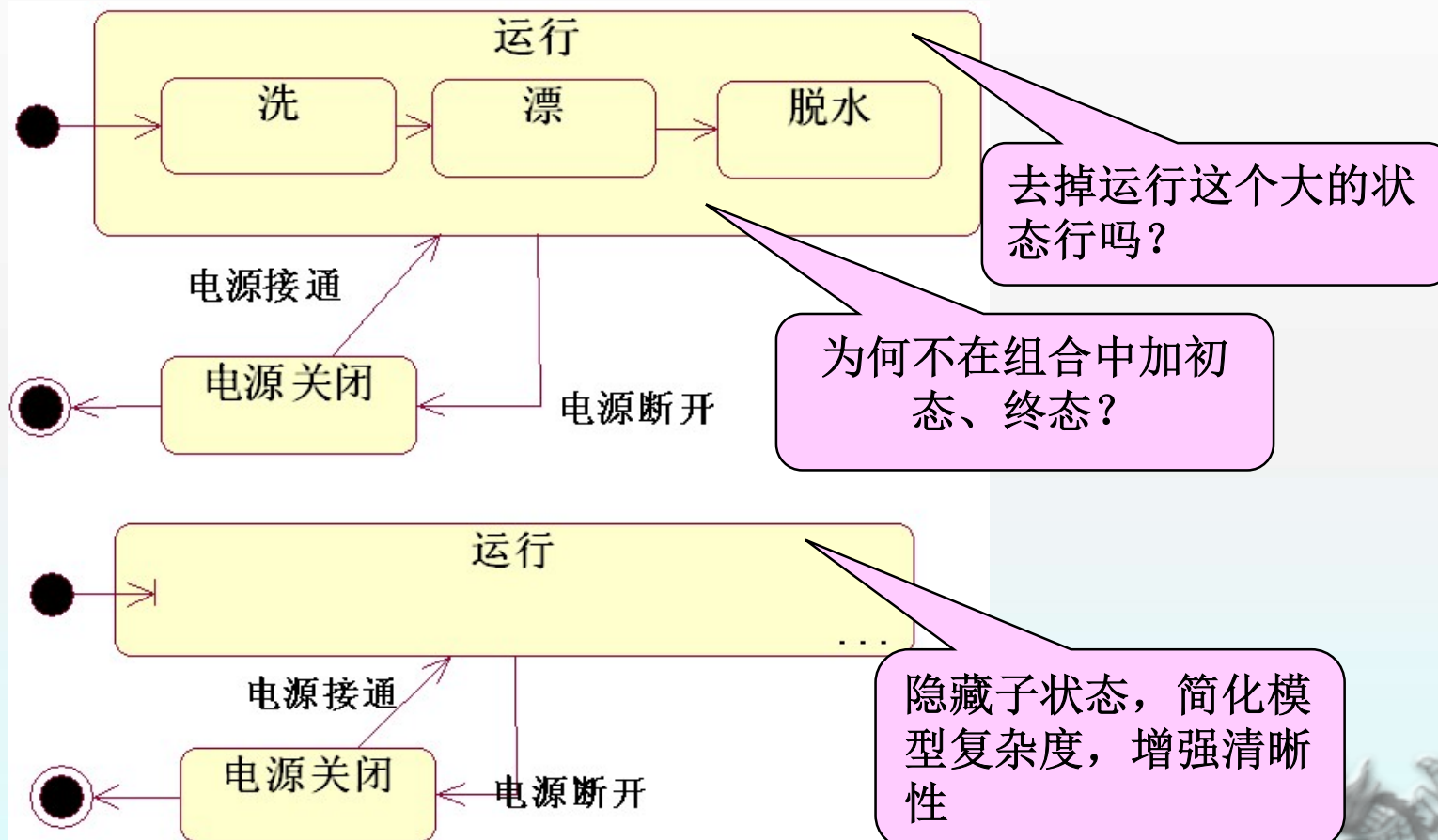
◇ 延迟事件 (deferred event)

- ◆ 是指在该状态下暂不处理，但将推迟到该对象的另一个状态下排队处理的事件



组合/嵌套状态

- ◆ 含有子状态的状态被称为组合或嵌套状态



- ◆ 组合状态可以使用“与”关系分解为并发子状态，或者通过“或”关系分解为互相排斥的顺序子状态。



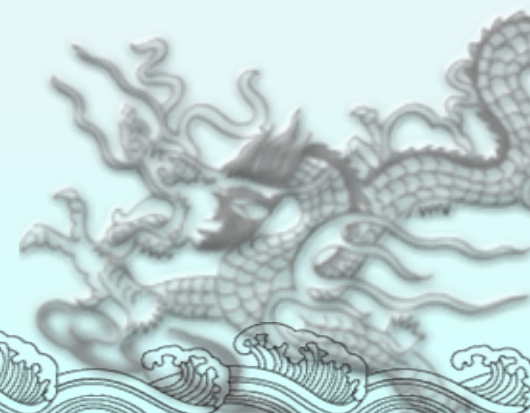
两种表示方法

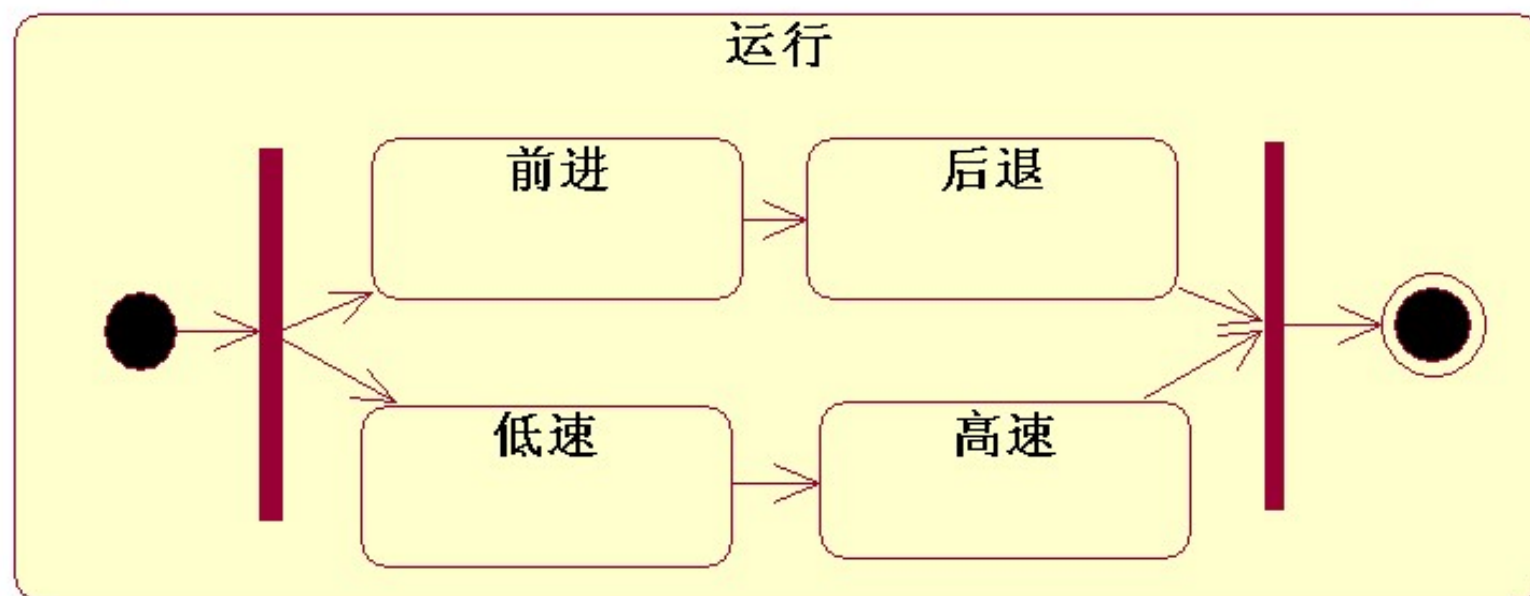
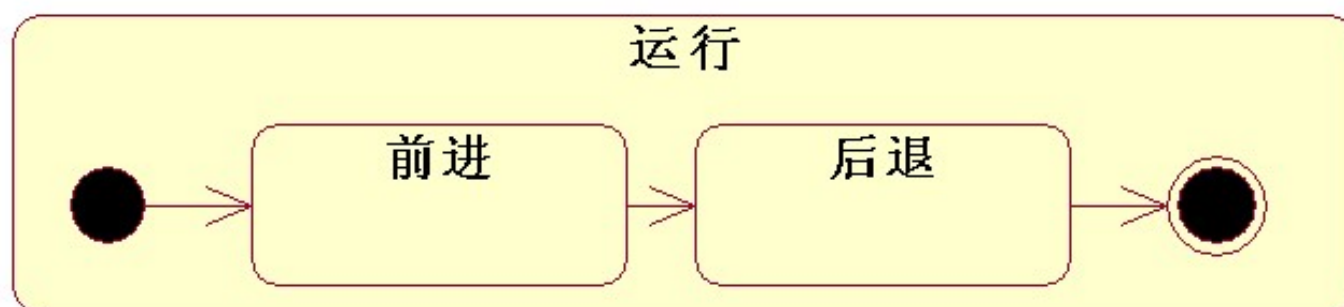
◇ 顺序子状态

- ◆ 如果一个组成状态的子状态对应的对象在其生命期内的任何时刻都只能处于一个子状态，即多个子状态之间是互斥的，不能同时存在，这种子状态称为顺序子状态。

◇ 并发子状态

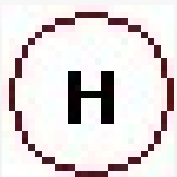
- ◆ 有时组合状态有两个或者多个并发的子状态机，此时称组成状态的子状态为并发子状态。



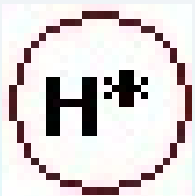


历史状态

历史状态是伪状态,其目的是记住从组合状态中退出时所处的子状态,当再次进入组合状态时,可以直接进入这个子状态,而不是再从组合状态的初态开始.



浅(**shallow**)历史状态,只记住最外层组合状态的历史.

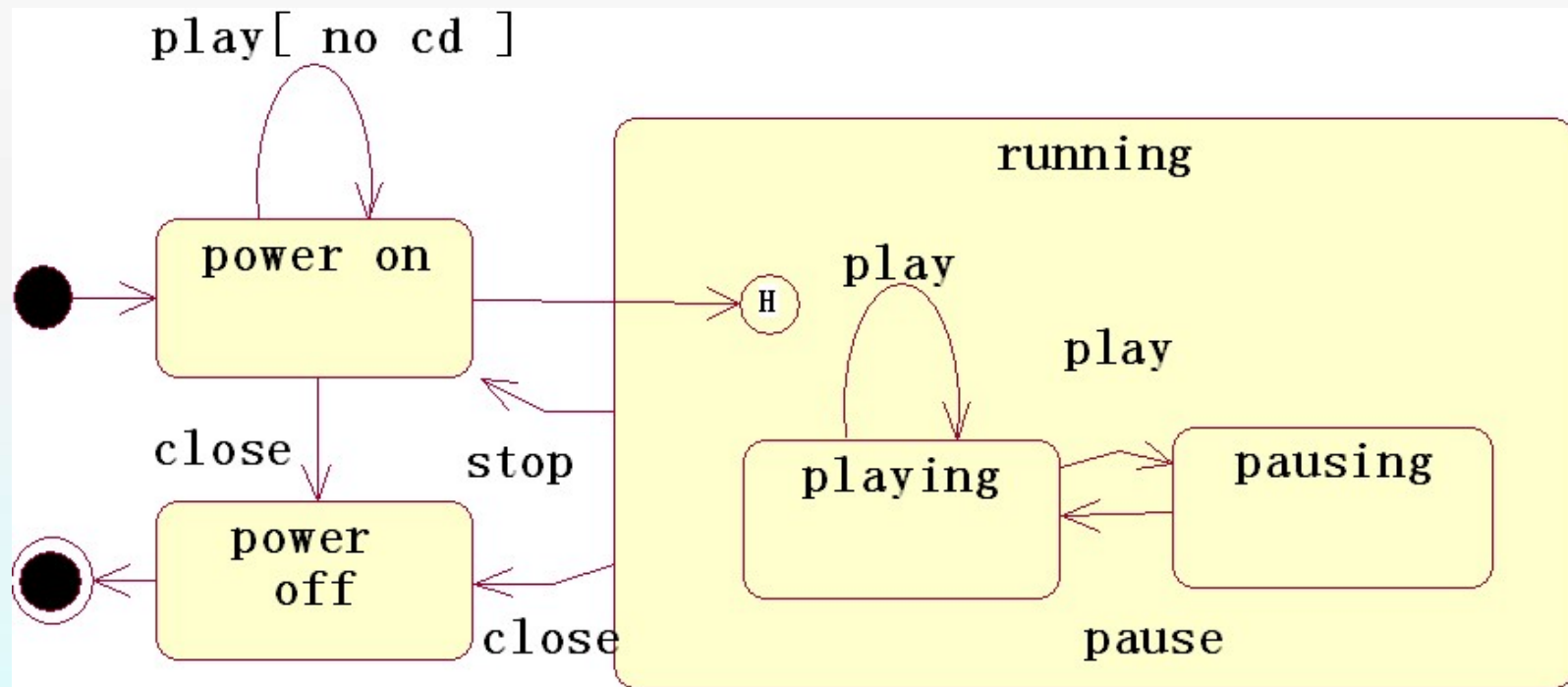


深(**deep**)历史状态,可以记住任意深度的组合状态的历史.



历史状态

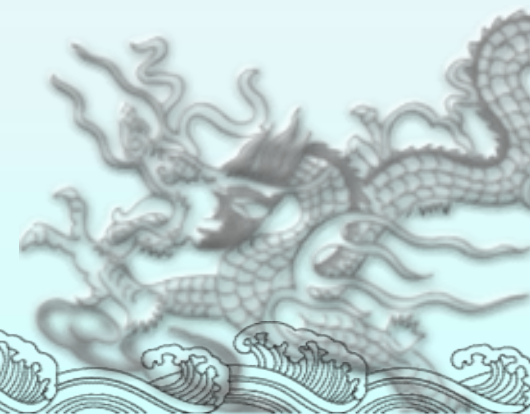
- ◇ CD Player, running状态被打断到power on状态, 再转回到running状态时, 希望直接进入历史状态



转移

◆ Transition

- ◆ 转移是两个状态间的一种关系，表示对象将在当前状态中执行动作，并在某个特定事件发生或某个特定的条件满足时进入后继状态。
- ◆ 每个转移只允许有一个事件触发，一个事件只允许有一个动作



转移的五要素（注意格式）

格式：事件（参数）[条件]/动作

◇ 源状态

◇ 目标状态

◇ 触发事件

- ◆ 如果箭头上不带任何事件名，表示是一个自动转换，当与源状态相关的活动完成时就会自动触发。

◇ 监护条件

◇ 动作



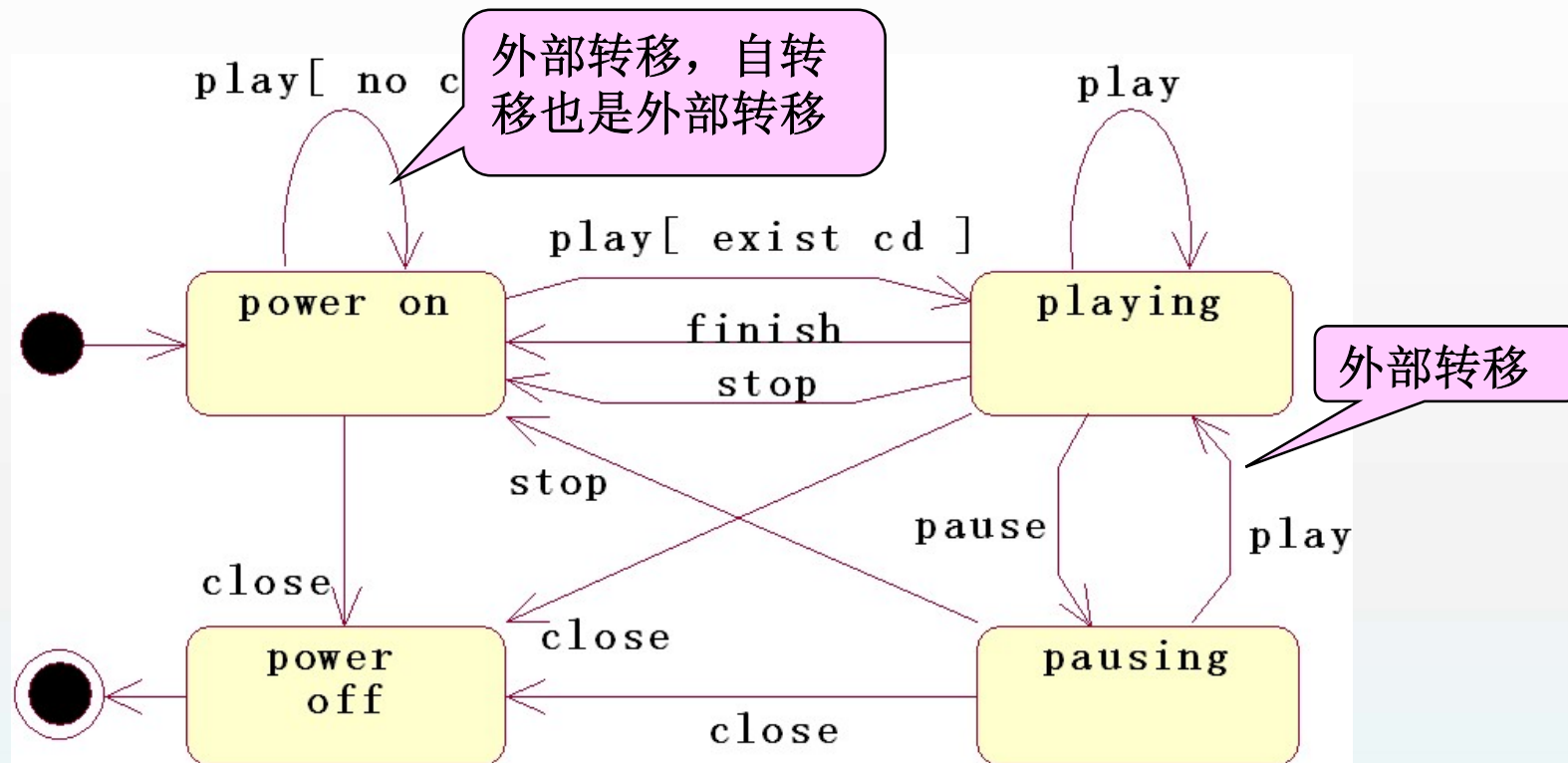
转移类型

外部转移

内部转移：不导致状态改变的转移，不会执行entry
和exit动作

自转移





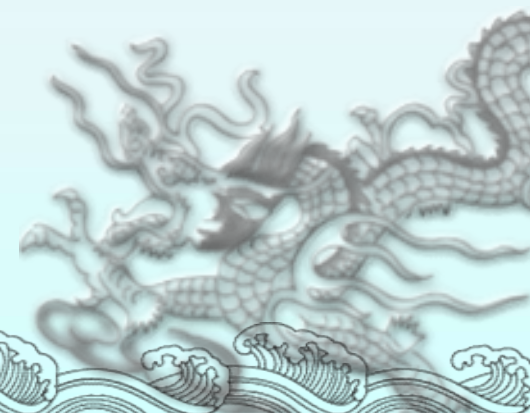
事件

Event

- ◇ 是对一个时间和空间上占有一定位置的有意义的事情的规格说明。
- ◇ 事件触发状态的转移

四类主要事件

- ◇ 信号事件
- ◇ 调用事件
- ◇ 变化事件
- ◇ 时间事件



信号signer事件

- ◆ 对象之间通过发送信号和接收信号实现通信。信号是一种**异步机制**。在计算机中，鼠标和键盘的操作均属于此类事件。对于一个信号而言，对象一般都有相应的事件处理器，如onMouseClicked（）等。



调用call事件

- ◆ 表示一个操作的调度。一个对象请求调用另一个对象的操作
- ◆ 信号是一个异步事件，而调用事件一般是同步的。就操作而言，发送者接收者转换到一个新的状态，控制返回给发送者。

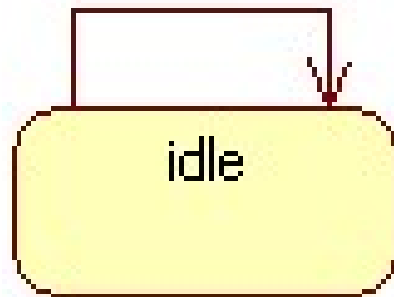


问题: 调用谁的操作?

变化change事件

- ◆ 用关键字When，后面跟布尔表达式
- ◆ When(temperature>120)/alarm()
- ◆ 变化事件的意图是要频繁测试表达式，只要表达式由假变为真，事件就会发生。

```
when(temperature>120) : / alarm()
```

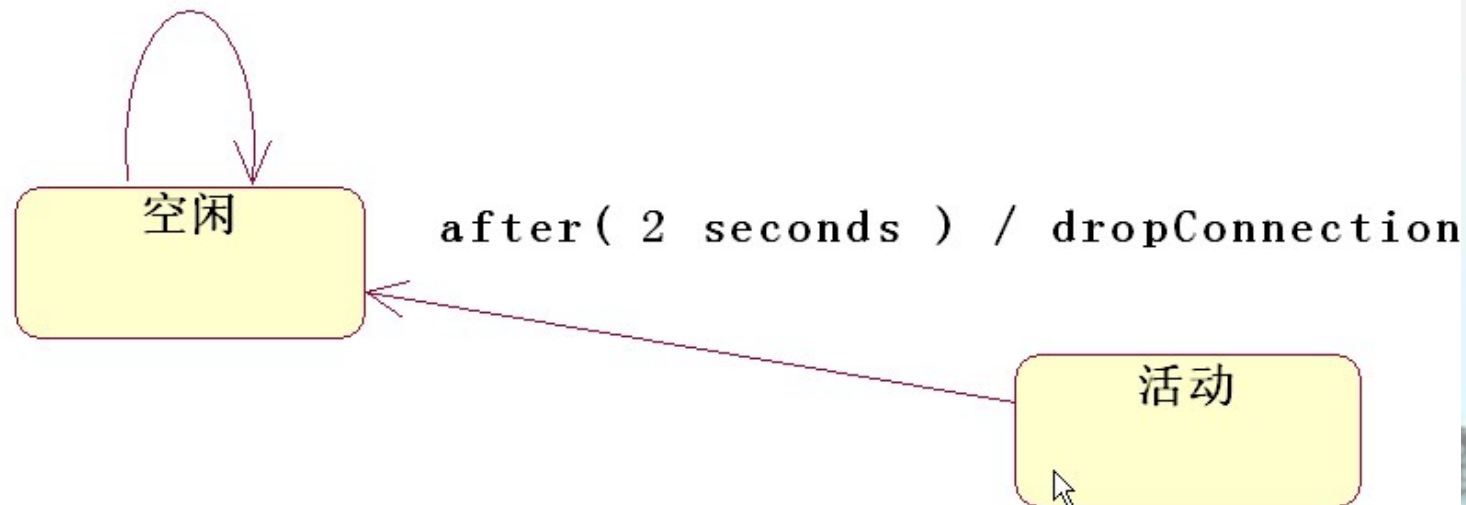


注意: 变化事件与监护条件的区别

时间 (time) 事件

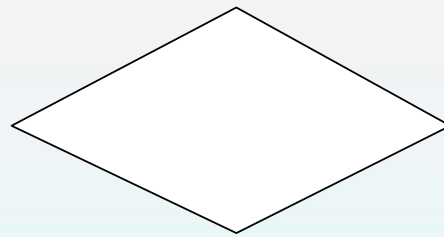
- ◆ 时间事件是指在绝对时间或在某个时间间隔内发生的事情所引起的事件。
- ◆ 例如到达某一时间或经过了某一段时间。用关键字 When 或 After 表示。

```
when( 11:35 ) / selfTest
```

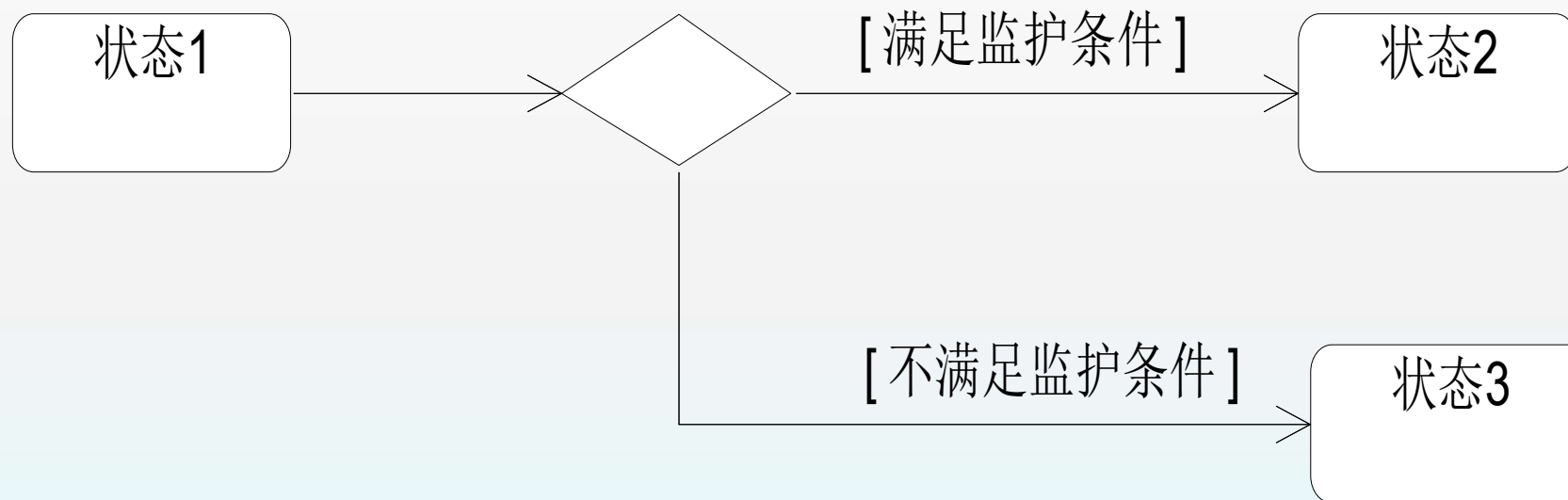


分支的表示

对象在外部事件的作用下，根据监护条件的不同值，转向不同的目标状态。即，对象的状态根据监护条件的取值而发生分支。分支用空心小菱形表示，如图所示。

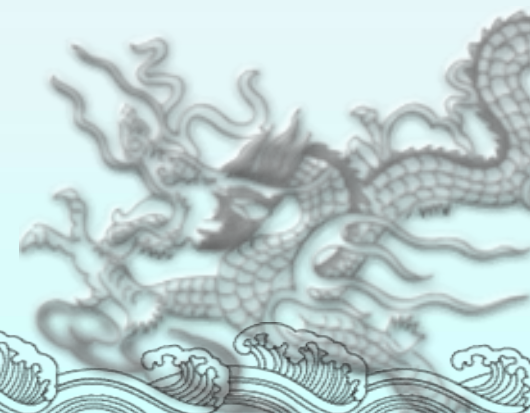


根据监护条件的真假可以触发不同的分支转换，如图所示。



内容提纲

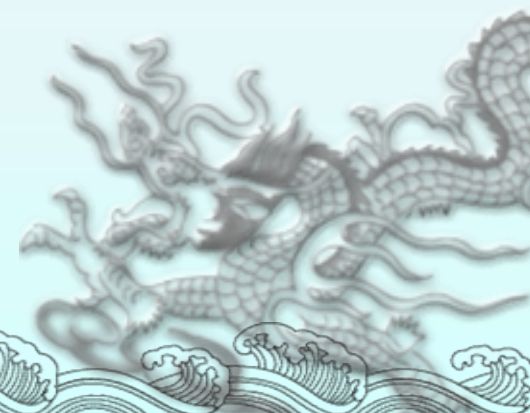
- 状态图
 - 相关概念
 - 实例讲解
- 活动图
 - 相关概念
 - 实例讲解



建立状态图

绘制状态图的一般步骤是：

1. 寻找主要的状态
2. 寻找外部事件，以便确定状态之间的转移
3. 详细描述每个状态和转移
4. 把简单状态图转换为复合状态图。

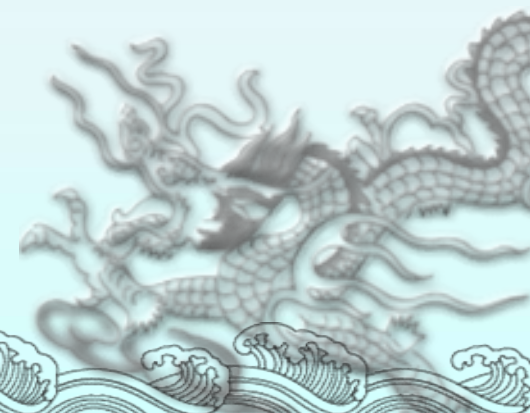


例子：航班机票预订

1. 确定状态

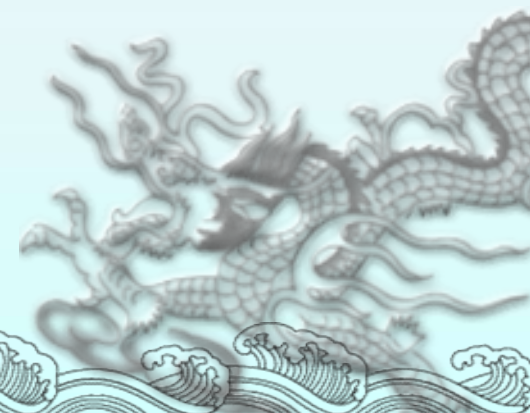
飞机票有以下4种状态：

- 无预订
- 部分预订
- 预订完
- 预订关闭



2.寻找外部事件

- (1) 预订：顾客预订机票。
- (2) 退订：顾客退订机票。
- (3) 关闭：机票管理员关闭订票系统。
- (4) 取消航班：飞机调度人员取消飞行计划。



请画出对应的状态间的转移

源目标	无预订	部分预订	预订完	预订关闭
无预订				
部分预订				
预订完				
预订关闭				



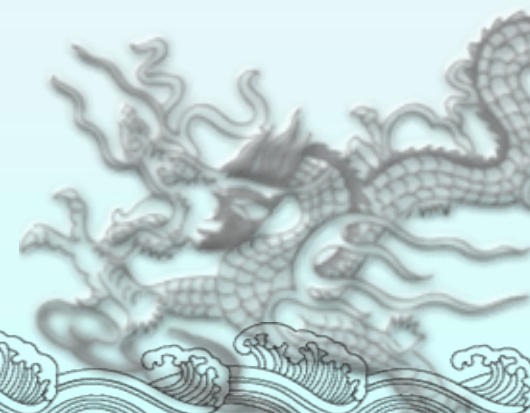
3.确定状态间的转移

源目标	无预订	部分预订	预订完	预订关闭
无预订		预订	不直接转移	关闭
部分预订	退订事件发生后，使预订人=0		预订，无空座	关闭
预订完	不直接转移	退订		关闭
预订关闭	无转移	无转移	无转移	



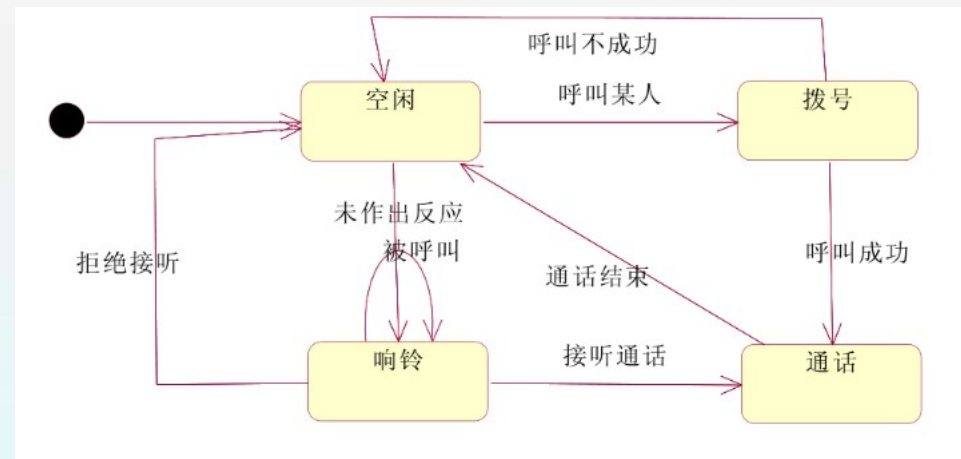
练习：

手机开机时，处于空闲状态；当用户开始呼叫某人时，手机进入拨号状态；如果呼叫成功，进入通话状态；如果呼叫不成功，重新进入空闲状态。在空闲状态被呼叫，进入响铃状态；如果用户接听，进入通话状态；如果一分钟不接听，重新进入空闲状态。



练习：

手机开机时，处于空闲状态；当用户开始呼叫某人时，手机进入拨号状态；如果呼叫成功，进入通话状态；如果呼叫不成功，重新进入空闲状态。在空闲状态被呼叫，进入响铃状态；如果用户接听，进入通话状态；如果一分钟不接听，重新进入空闲状态。



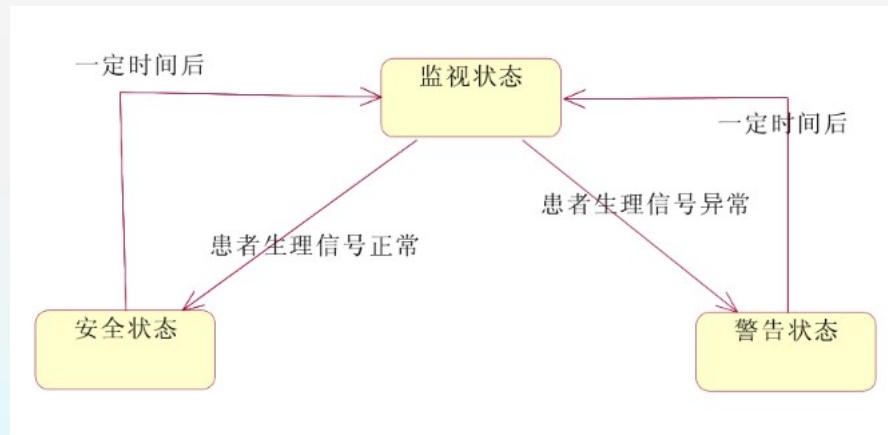
练习：患者监护系统

患者监护系统的要求是随时接收每个病人的生理信号（脉搏、体温、血压等），定时记录病人情况形成患者日志，当病人的生理信号超出医生规定的安全范围时向值班护士发出警告信息。请画出患者监护系统的状态图。



练习：患者监护系统

患者监护系统的要求是随时接收每个病人的生理信号（脉搏、体温、血压等），定时记录病人情况形成患者日志，当病人的生理信号超出医生规定的安全范围时向值班护士发出警告信息。请画出患者监护系统的状态图。

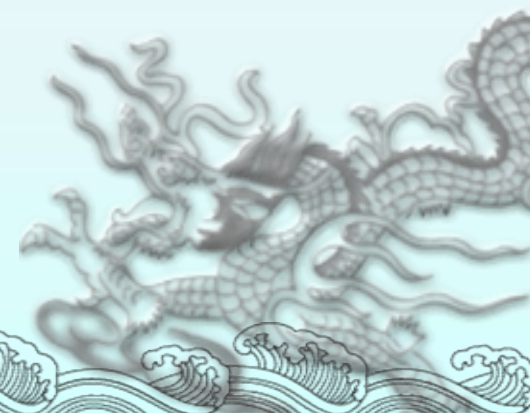


练习： 某公司拟研制一款高空监视无人直升机，该无人机采用遥控—自主复合型控制实现垂直升降。该直升机飞行控制系统由机上部分和地面部分组成，机上部分主要包括无线电传输设备、飞控计算机、导航设备等，地面部分包括遥控操纵设备、无线电传输设备以及地面综合控制计算机等。其主要工作原理是地面综合控制计算机负责发送相应指令，飞控计算机按照预定程序实现相应功能。经过需求分析，对该无人直升机控制系统纵向控制基本功能整理如下：

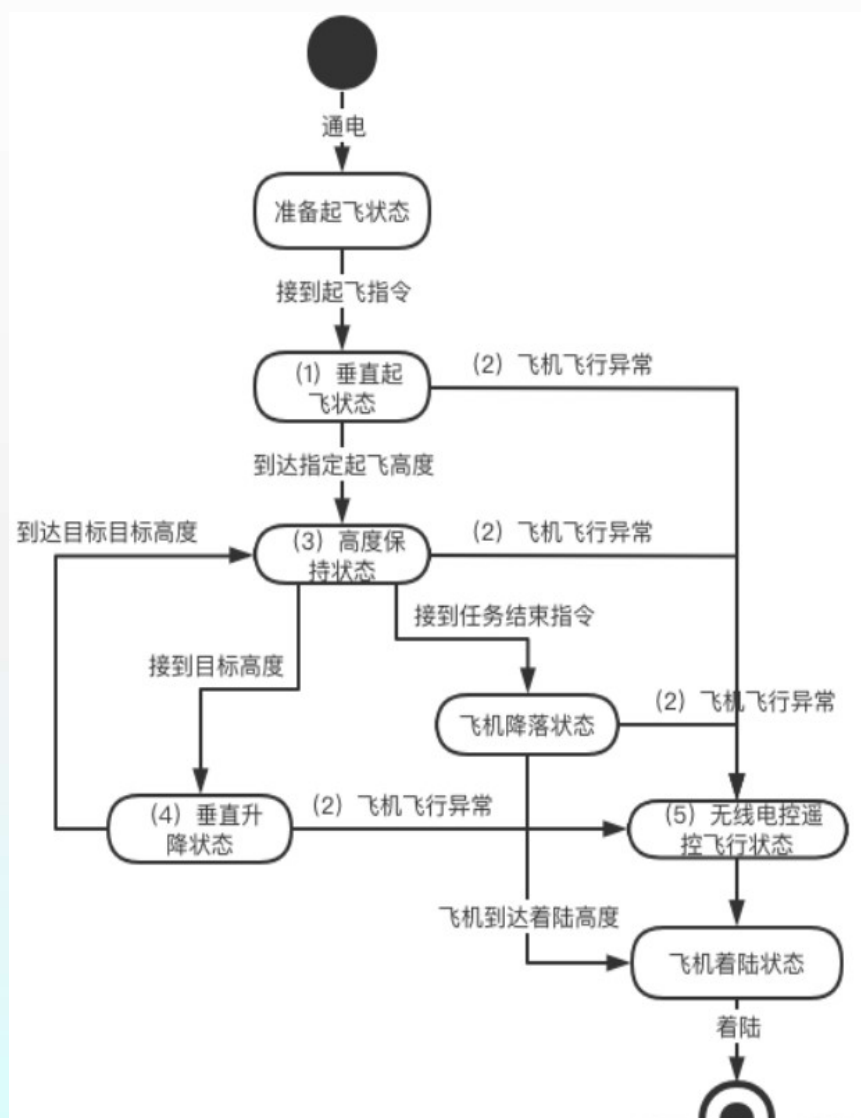
- (a) 飞控计算机加电后，应完成系统初始化，飞机进入准备起飞状态；
- (b) 在准备起飞状态中等待地面综合控制计算机发送起飞指令，飞控计算机接收到起飞指令后，进入垂直起飞状态；
- (c) 垂直起飞过程中如果飞控计算机发现飞机飞行异常，飞行控制系统应转入无线电遥控飞行状态，地面综合控制计算机发送遥控指令；
- (d) 垂直起飞达到预定起飞高度后，飞机应进入高度保持状态；
- (e) 飞控计算机在收到地面综合控制计算机发送的目标高度后，飞机应进入垂直升降状态，接近目标高度；垂直升降过程中出现飞机飞行异常，控制系统应转入无线电遥控飞行；
- (f) 飞机到达目标高度后，应进入高度保持状态，完成相应的任务；
- (g) 飞机在接到地面综合控制计算机发送的任务执行结束指令后，进入飞机降落状态；
- (h) 飞机降落过程中如果出现飞机飞行异常，控制系统应转入无线电遥控飞行；
- (i) 飞机降落到指定着陆高度后，进入飞机着陆状态，应按照预定着陆算法，进行着陆；
- (j) 无线电遥控飞行中，地面综合控制计算机发送着陆指令，飞机进入着陆状态，应按照预定着陆算法，进行着陆。

练习：

- (1) 垂直起飞状态
- (2) 飞机飞行异常
- (3) 高度保持状态
- (4) 垂直升降状态
- (5) 无线电控遥控飞行状态

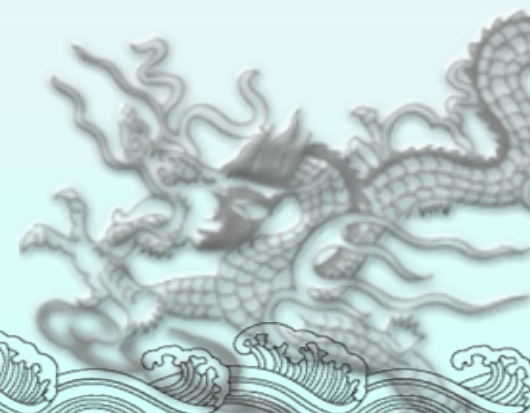


练习：



内容提纲

- 状态图
 - 相关概念
 - 实例讲解
- 活动图
 - 相关概念
 - 实例讲解

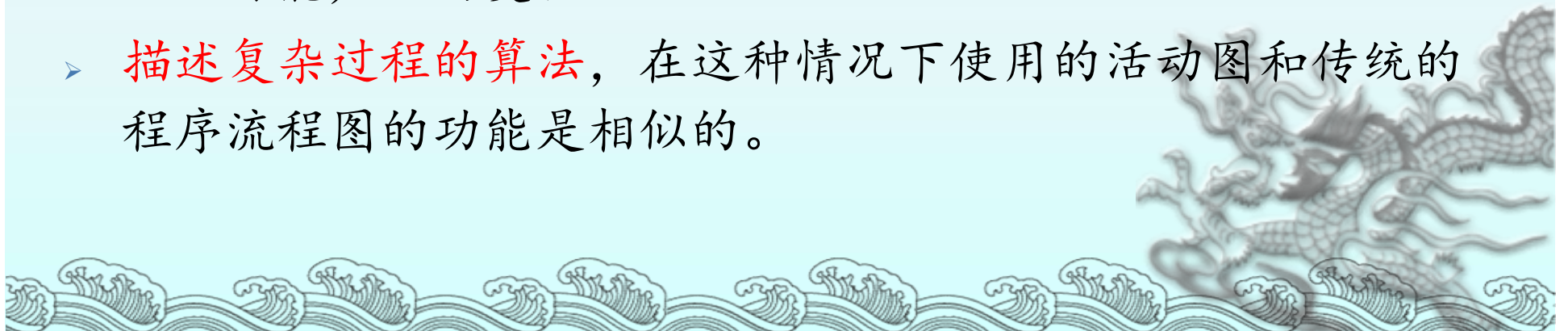


活动图

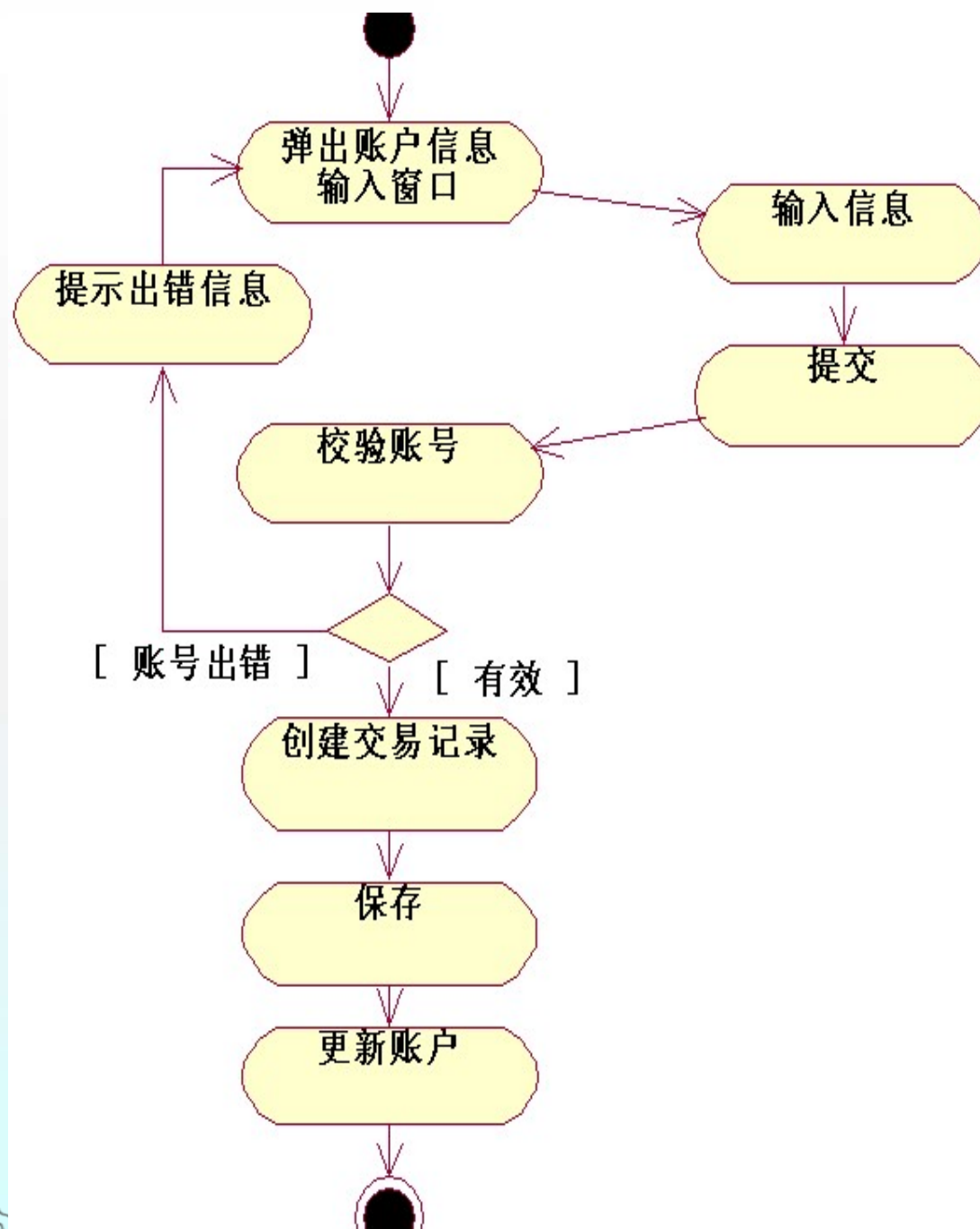
- **活动图** (Activity Diagram) 主要用于描述系统中对象的一个**活动**到另一个活动的**控制流**、一个事务的**活动序列**、**工作的流程**和**并发**的处理能力。
- **需求分析**中的活动图主要用来配合描述某个重要的用例，**用图形的方式对用例的具体实现的过程进行描述**，这样比文字的描述更直观、更准确地说明用例的功能。
- 在**系统分析与设计过程**中，经常用活动图进一步深入描述用例图的具体实现流程。
- **系统设计**中的活动图对系统的**业务 workflow**、**操作建模**，它描述业务流中从一个活动到另一个活动的控制流。它描述活动的序列，用来**分析和验证用例**，理解工作流程，**描述复杂过程的算法**，并且支持带条件的行为和**并发行为的多线程建模**。

活动图的作用:

- 描述一个操作执行过程中所完成的工作，说明角色、工作流、组织和对象是如何工作的。
- 活动图对用例描述尤其有用，它可建模用例的工作流、显示用例内部和用例之间的路径。
- 显示如何执行一组相关的动作，以及这些动作如何影响它们周围的对象。
- 活动图可以画出工作流用以描述业务，有利于与领域专家进行交流。通过活动图可以明确业务处理操作是如何进行的，以及可能产生的变化。
- 描述复杂过程的算法，在这种情况下使用的活动图和传统的程序流程图的功能是相似的。

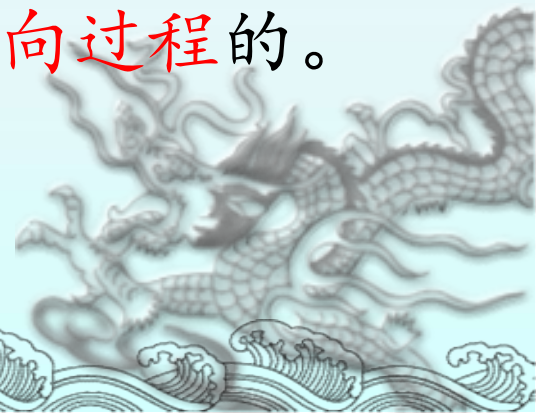


◆ 存款



活动图与流程图的区别

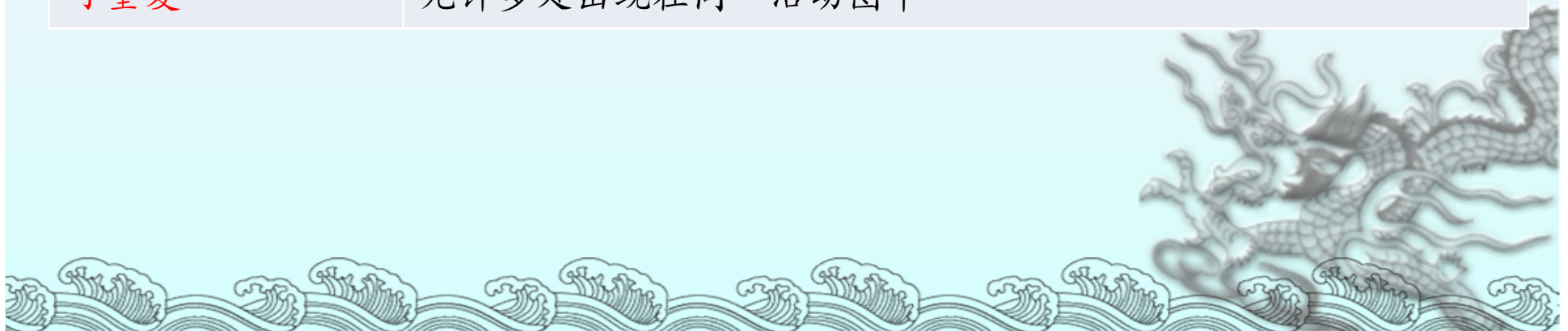
1. **流程图着重描述处理过程**，它的主要控制结构是顺序、分支和循环，各个处理过程之间有严格的顺序和时间关系；而**活动图描述的是对象活动的顺序关系所遵循的规则**，它着重表现的是系统的行为，而非系统的处理过程。
2. **活动图能够表示并发活动**的情形，而**流程图不能**。
3. **活动图是面向对象的**，而**流程图是面向过程的**。



活动图中的基本概念

1. 动作状态：指执行原子的、不可中断的动作，并在此动作完成后转换到另一个状态。

原子的	构造活动图的最小单位，无法分解为更小的部分
不可中断	一旦运行就不能中断，一直运行到结束
是瞬时的行为	所占用的处理时间极短，有时甚至可以忽略
有入转换	入转换可以是动作流，也可以是对象流
与状态图中的状态不同	不能有入口动作和出口动作，也不能有内部转移
可重复	允许多处出现在同一活动图中



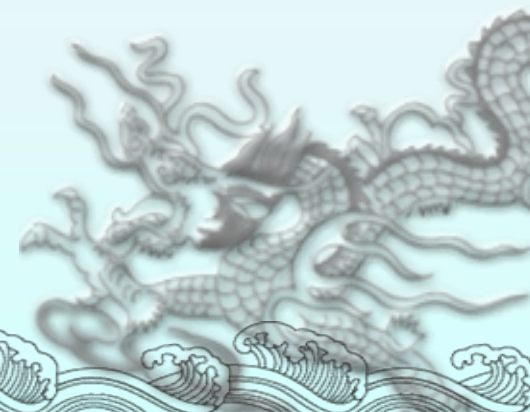
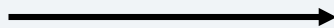
2. 活动状态：活动状态是非原子性的，用来表示一个具有子结构的纯粹计算的执行

- ▶ 活动状态可以分解成其他子活动或动作状态，由于它是一组不可中断的动作或操作的组合，所以可以被中断；
- ▶ 活动状态的内部活动可以用另一个活动图来表示；
- ▶ 活动状态可以有入口动作和出口动作，也可以有内部转移；
- ▶ 动作状态是活动状态的一个特例，如果某一个活动状态只包括一个动作，那么它就是一个动作状态。



3.转换

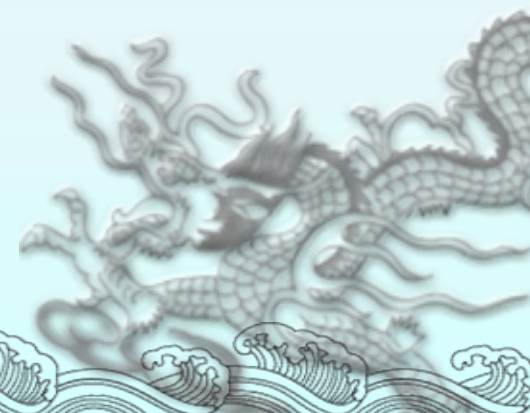
当一个活动结束时，活动控制流就会马上传递给下一个活动节点，在活动图中称之为“转换”，用一条带箭头的直线来表示转换。

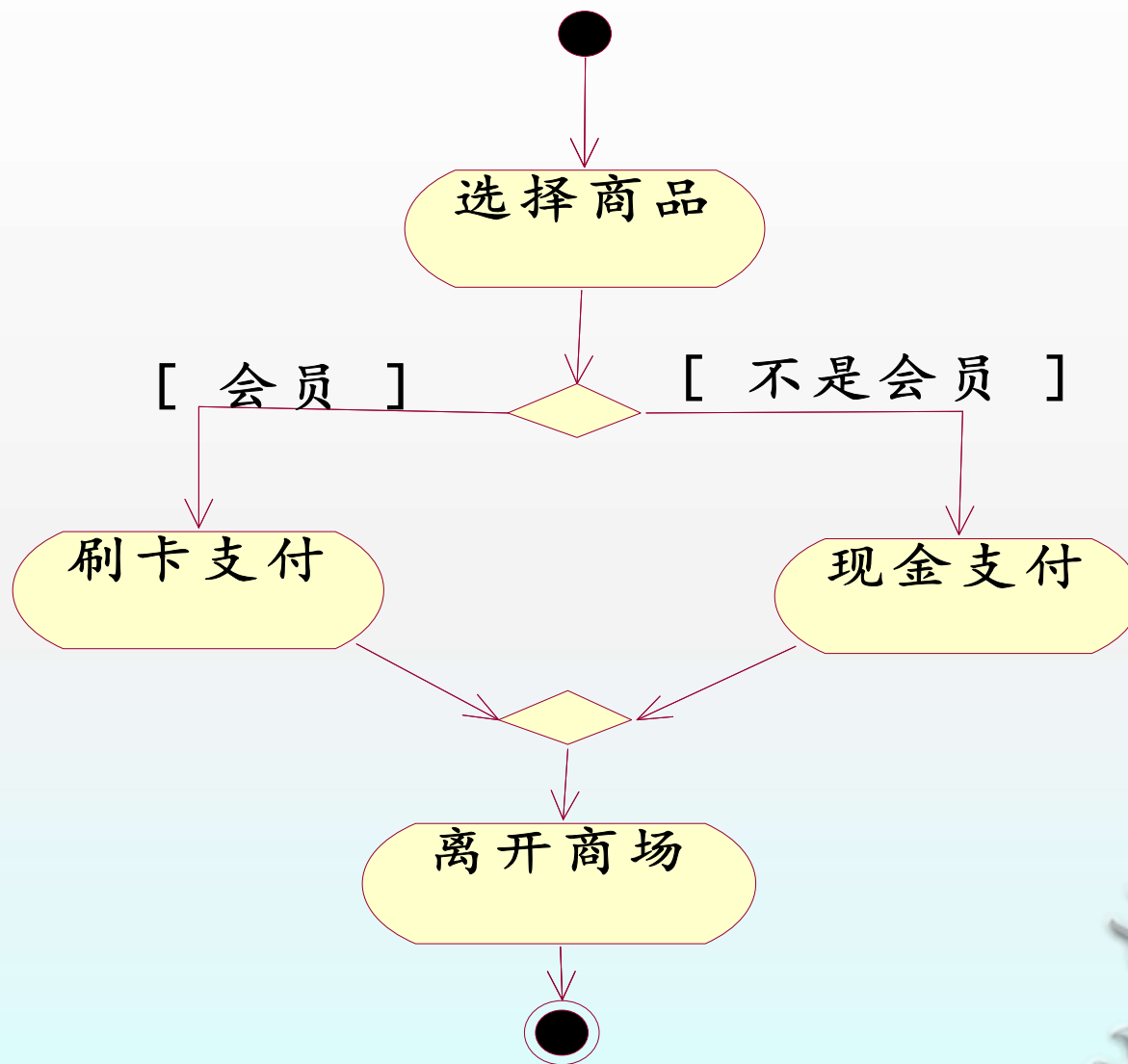


4. 分支

分支(Branch)

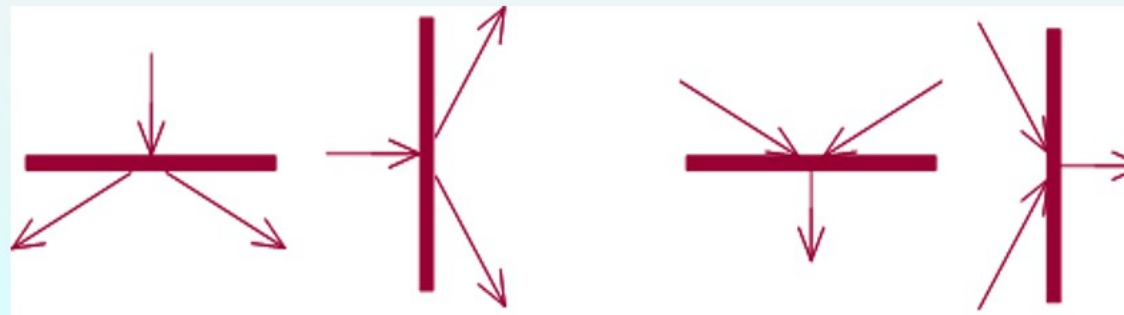
- ◆ 分支用于描述基于某个条件的可选择路径。
- ◆ 一个分支可以有一个进入转换和两个或多个输出转换。
- ◆ 在每条输出转换上都有监护条件表达式保护，当且仅当监护条件表达式为真时，该输出路径才有效。
- ◆ 在所有输出转换中，其监护条件不能重叠，而且它们应该覆盖所有的可能性。
- ◆ 分支在图形表示上用菱形表示。



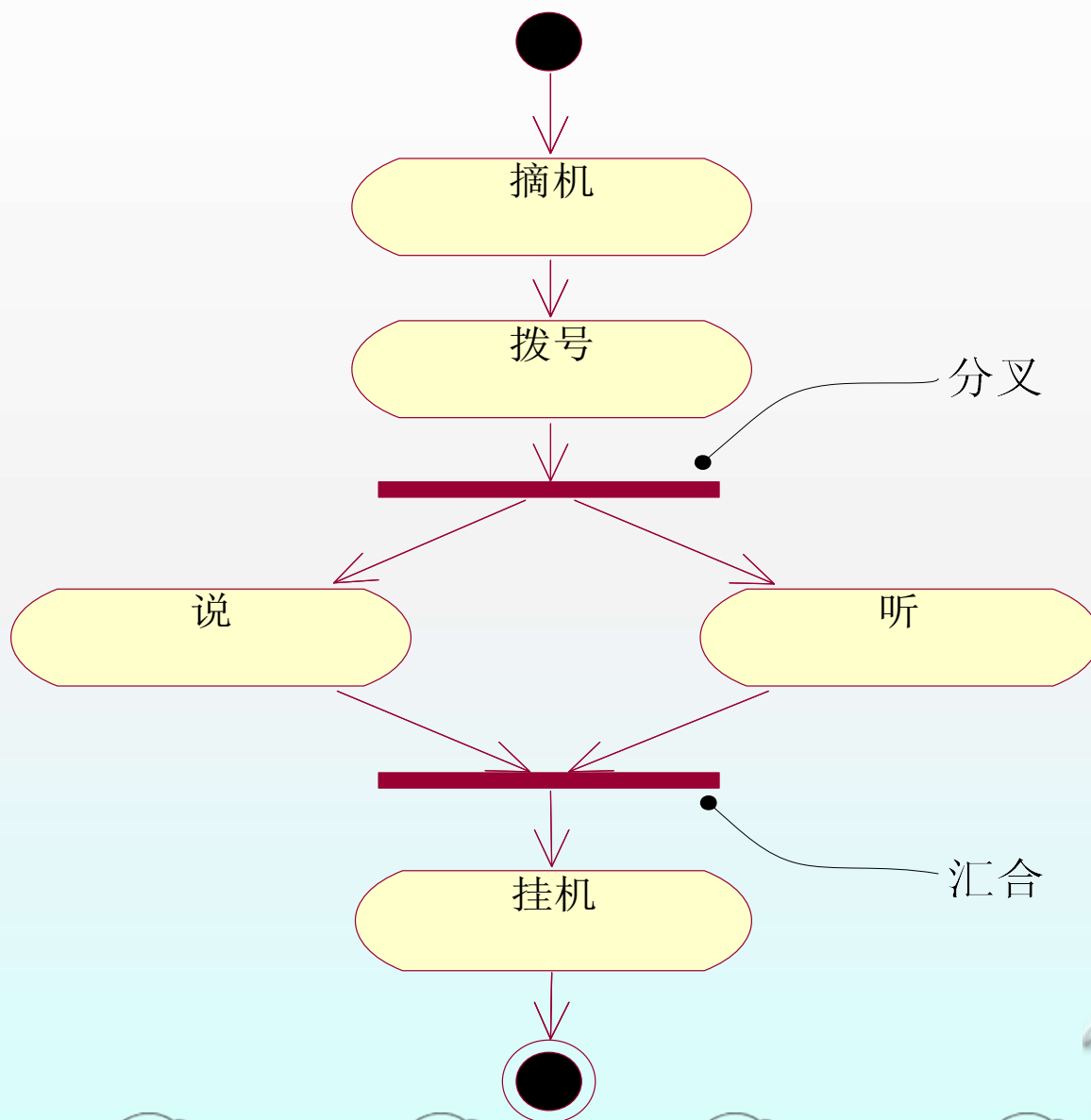


5.分叉 (fork) 和汇合 (join)

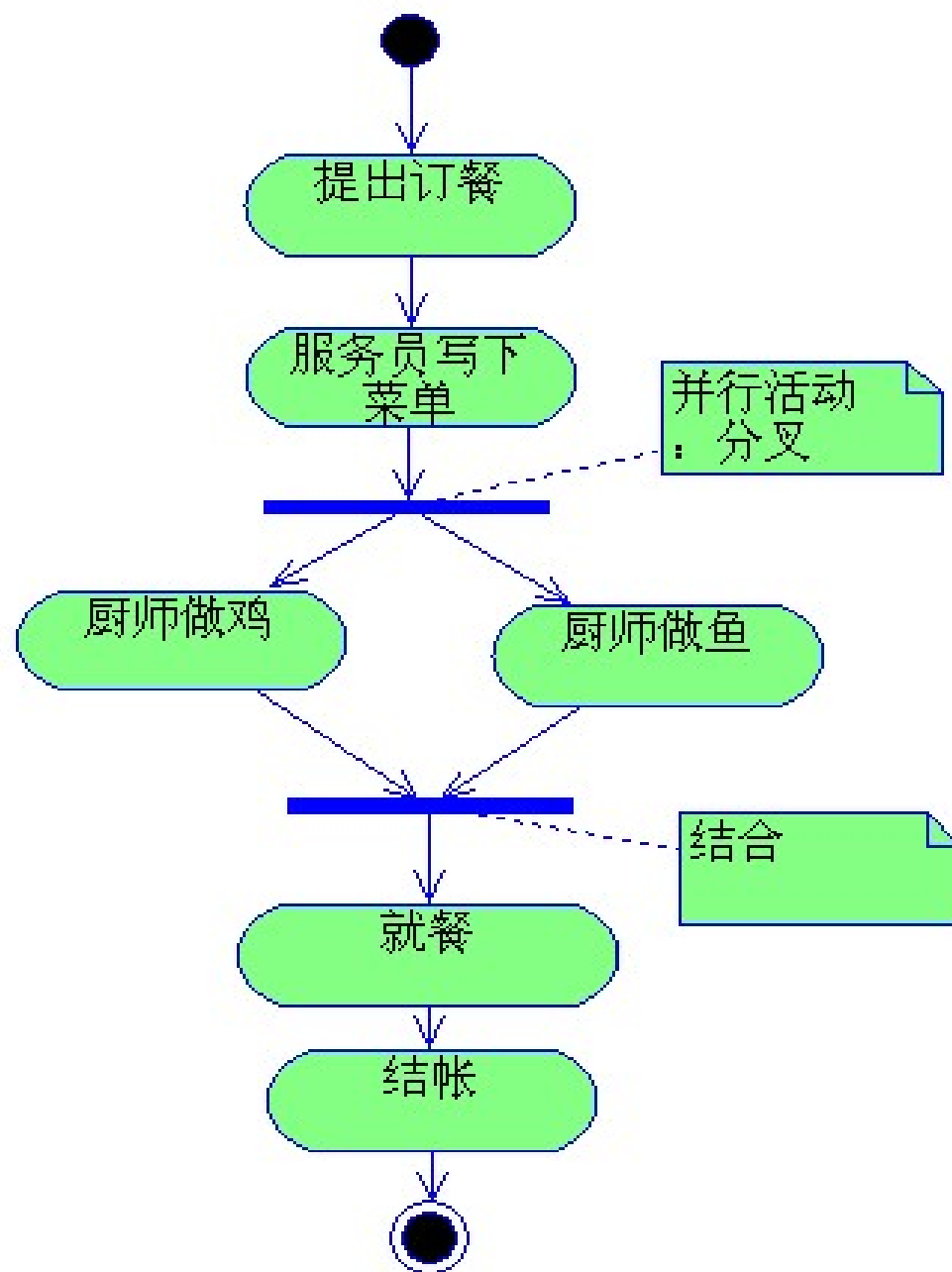
- ◆ 在UML中使用分叉和汇合表示**并行**发生的事件流
- ◆ 分叉表示把**一个单独的控制流**分成**两个或多个并发的控制流**。一个分叉可以有一个进入转移和两个或多个输出转移，每一个转移表示一个独立的控制流。
- ◆ 汇合表示两个或多个并发控制流的同步发生，一个汇合可以有**两个或多个进入转移**和一个输出转移。
- ◆ 分叉和汇合在图形上都使用**同步条**来表示，同步条通常用一条粗的水平线表示



描述打电话活动中的并发事件

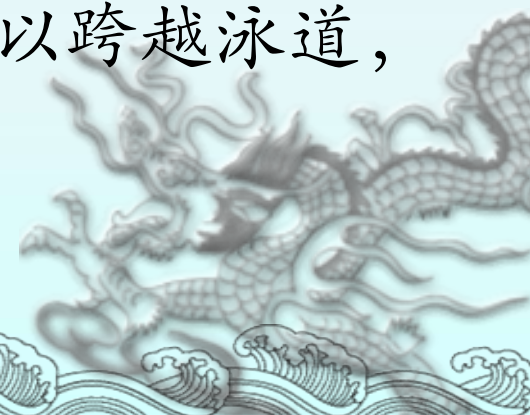


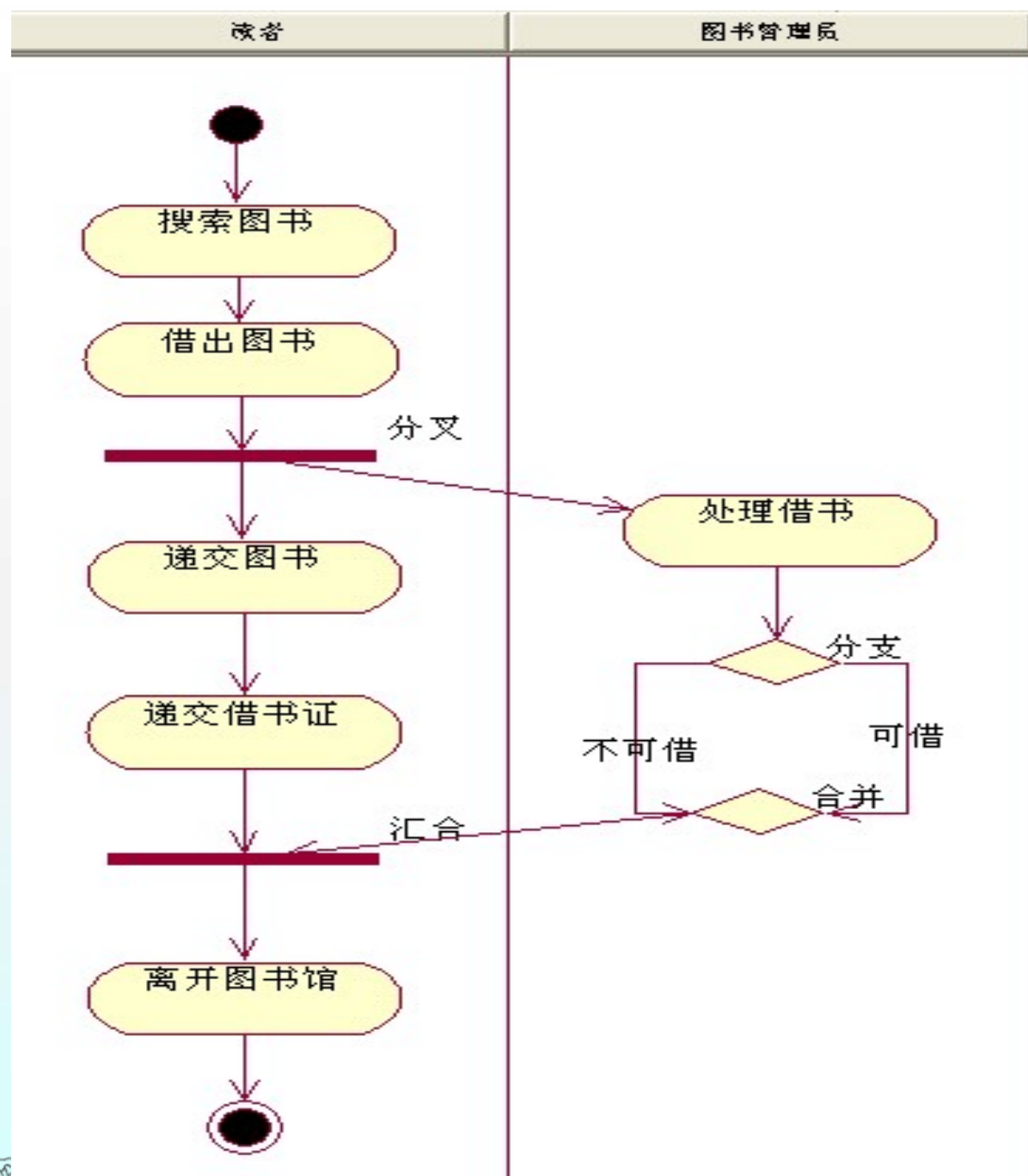
餐馆就餐活动的并发事件



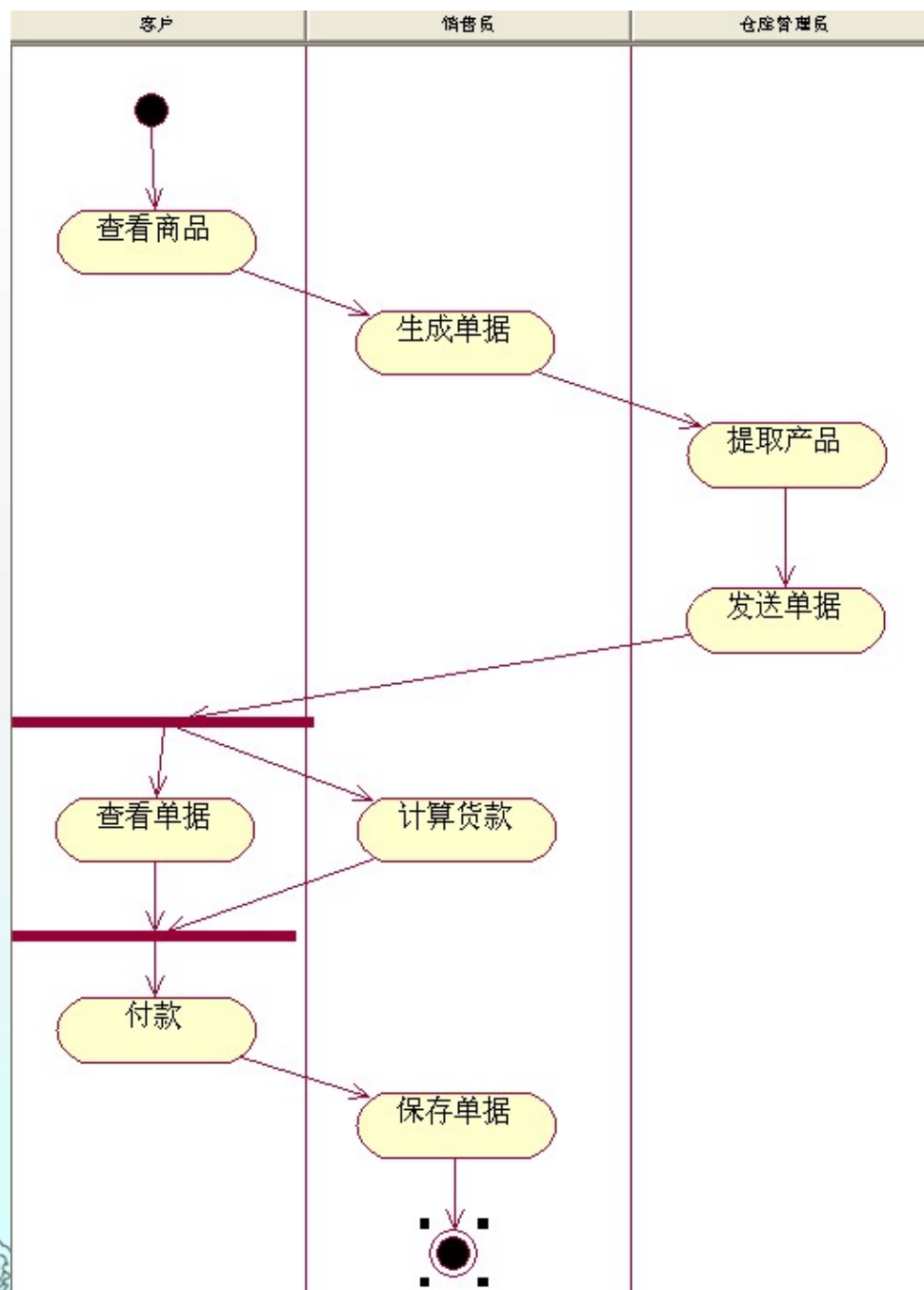
6.泳道 (swimlane)

- ◆ “泳道”技术，是将一个活动图中的活动状态进行**分组**，每一组表示一个特定的类、人或部门，他们负责完成组内的活动
- ◆ “泳道”技术来描述每个活动是由哪个对象负责完成
- ◆ UML中，每个组被称为一个泳道，用一条垂直的实线与邻居分开
- ◆ 每个活动都明确属于一个泳道，不可以跨越泳道，而转移则可以跨越泳道

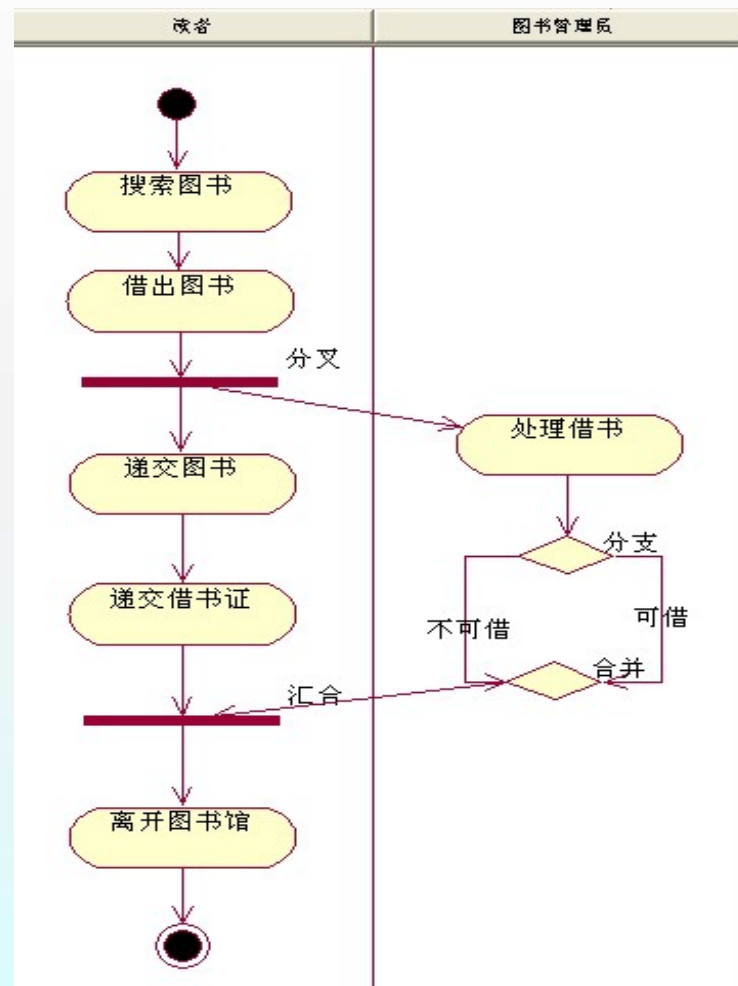
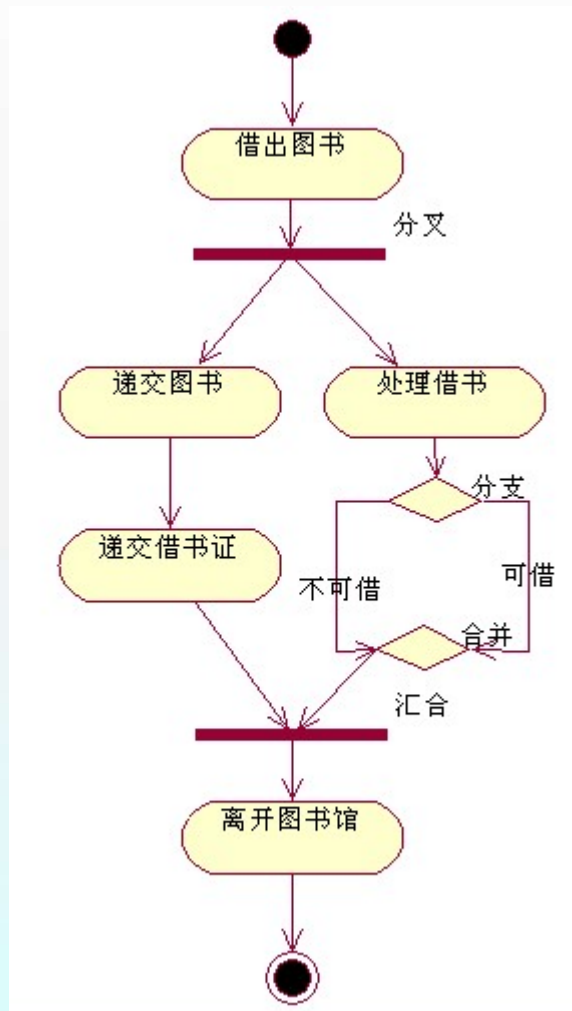




用活动图描述客户在商店中购买物品的过程。

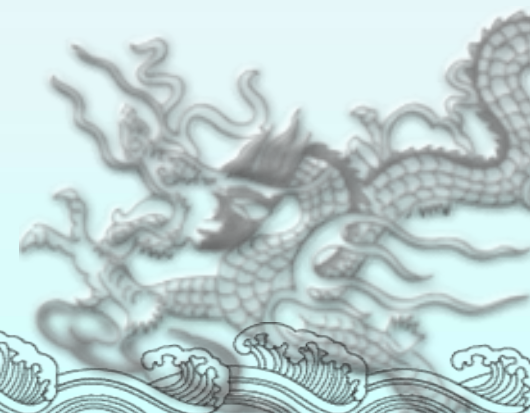


带泳道活动图的特点：泳道显示了对象对系统所做的贡献。



7. 对象流

- 对象流是动作状态或者活动状态与对象之间的依赖关系，表示动作使用对象或动作对对象的影响。
- 用活动图描述某个对象时，可以把涉及到的对象放置在活动图中并用一个**依赖**将其连接到进行**创建**、**修改**和**撤销**的动作状态或者活动状态上，对象的这种使用方法就构成了**对象流**。

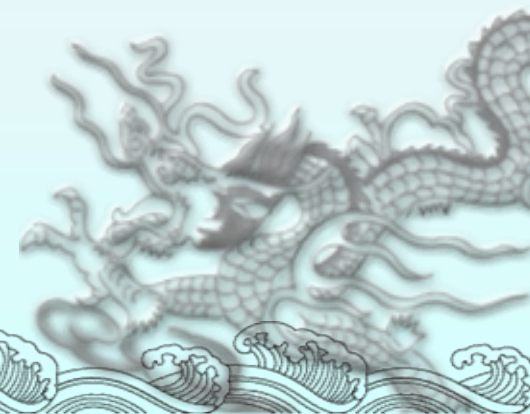


对象流中的对象有以下特点：

(1)一个对象可以由多个动作操作。

(2)一个动作输出的对象可以作为另一个动作输入的对象。

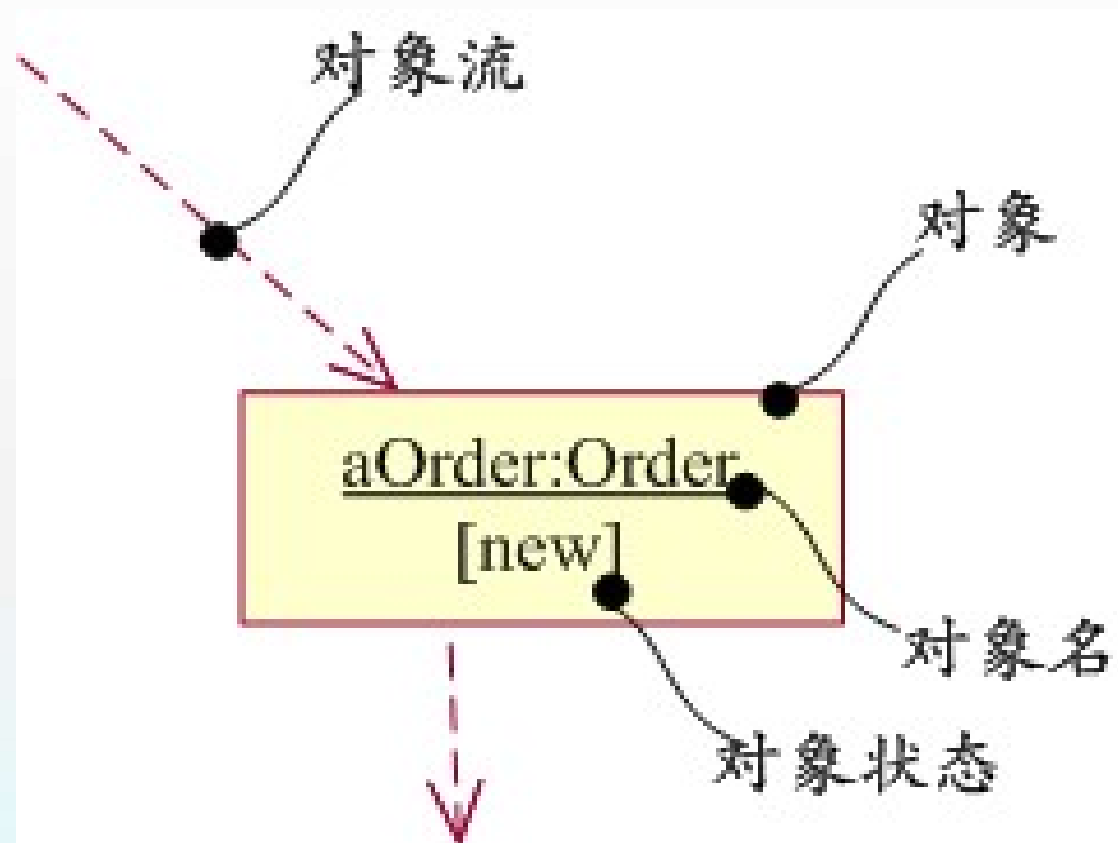
(3)在活动图中，同一个对象可以多次出现，它的每一次出现表示该对象正处于对象生存期的不同时间点。

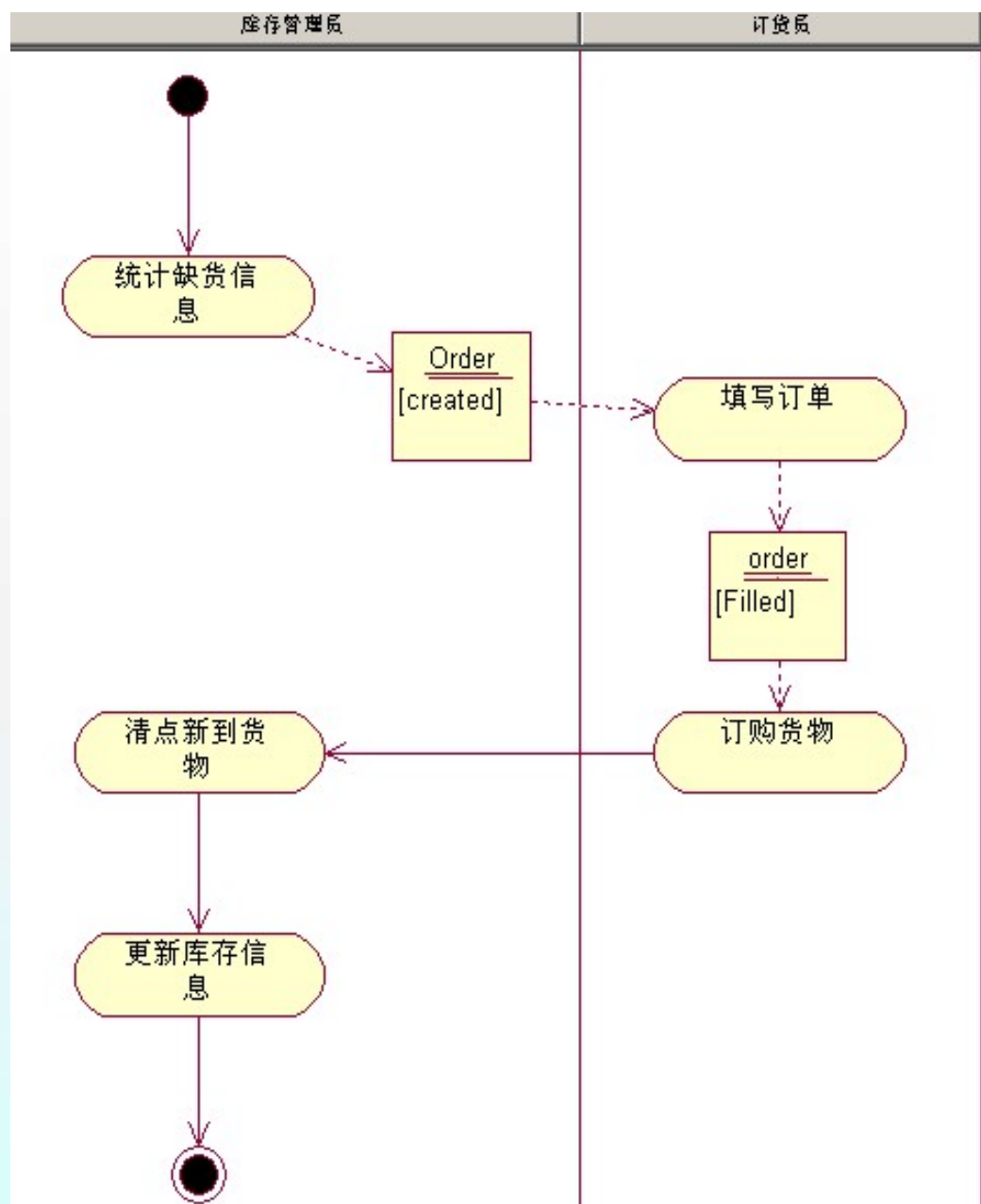


对象流用带有箭头的虚线表示。如果箭头是从动作状态出发指向对象，则表示动作对对象施加了一定的影响。施加的影响包括创建、修改和撤销等。如果箭头从对象指向动作状态，则表示该动作使用对象流所指向的对象。

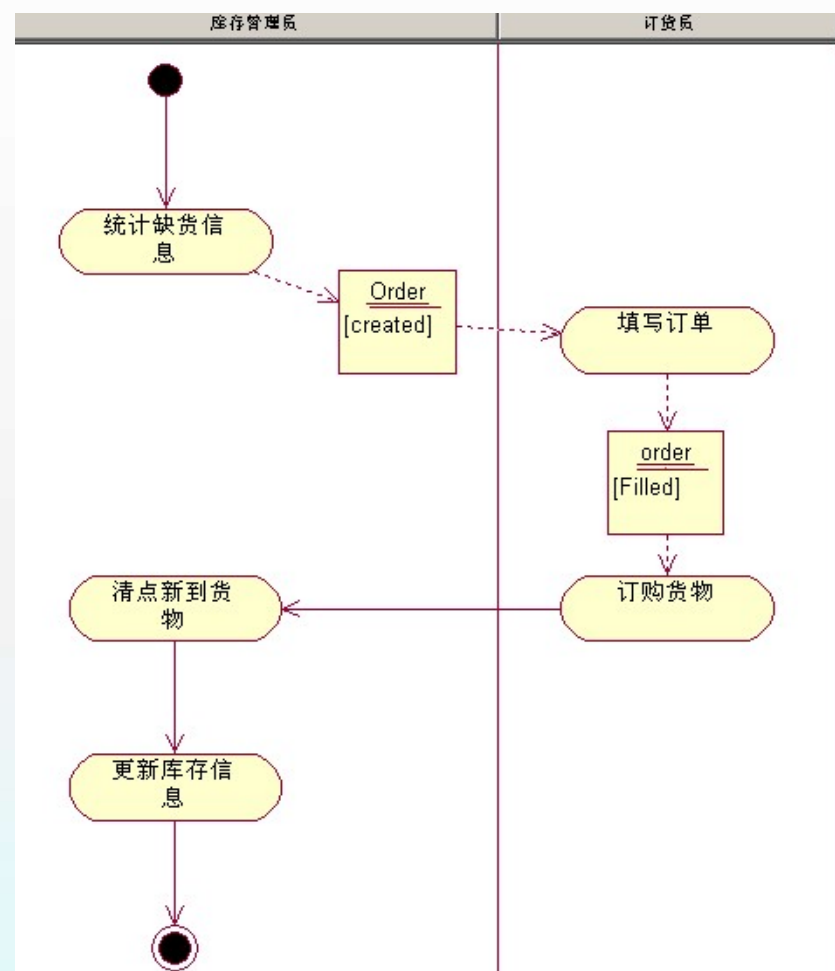
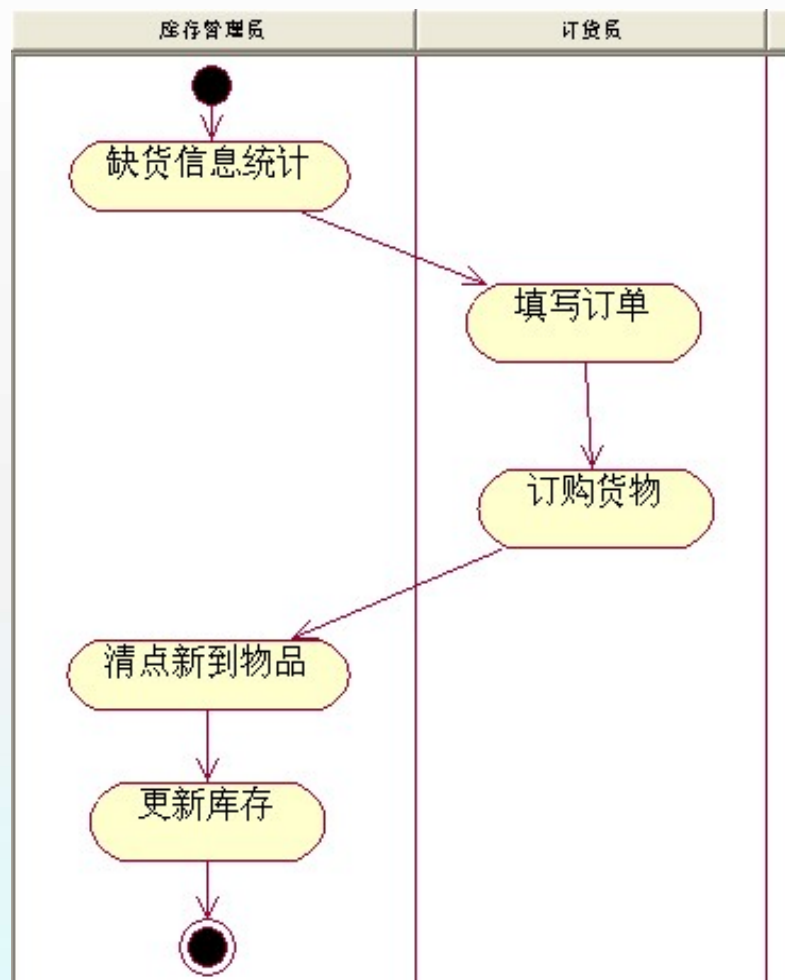
状态图中的对象用矩形表示，矩形内是该对象的名称，名称下的方括号表明对象此时的状态。







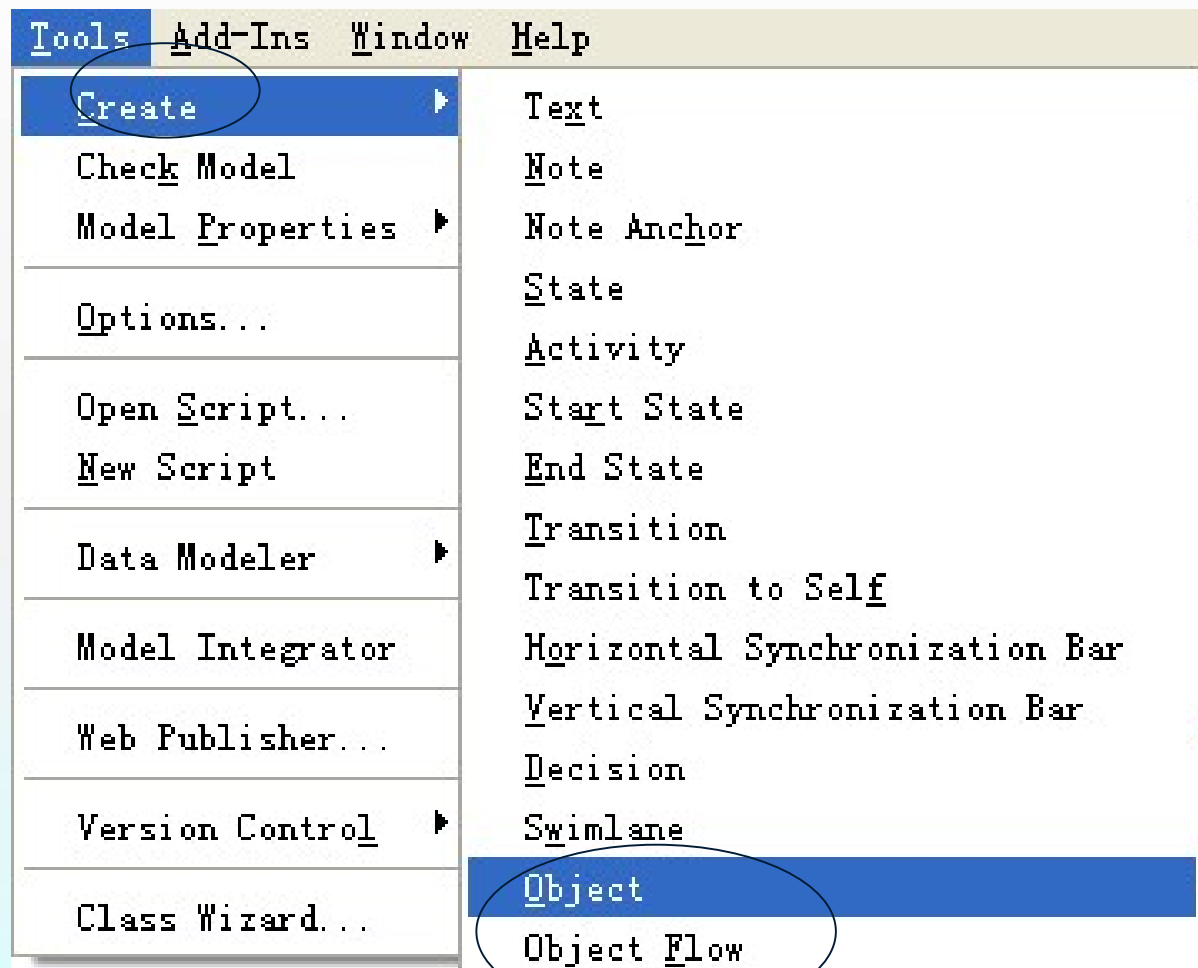
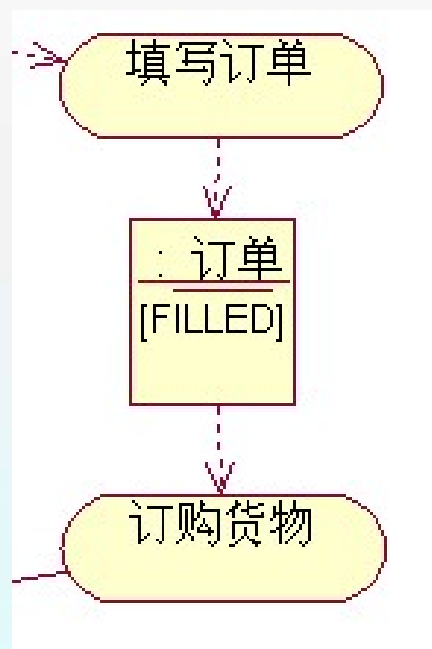
活动图中加入对象流更能显示系统的交互：



在活动图中添加对象及对象流：

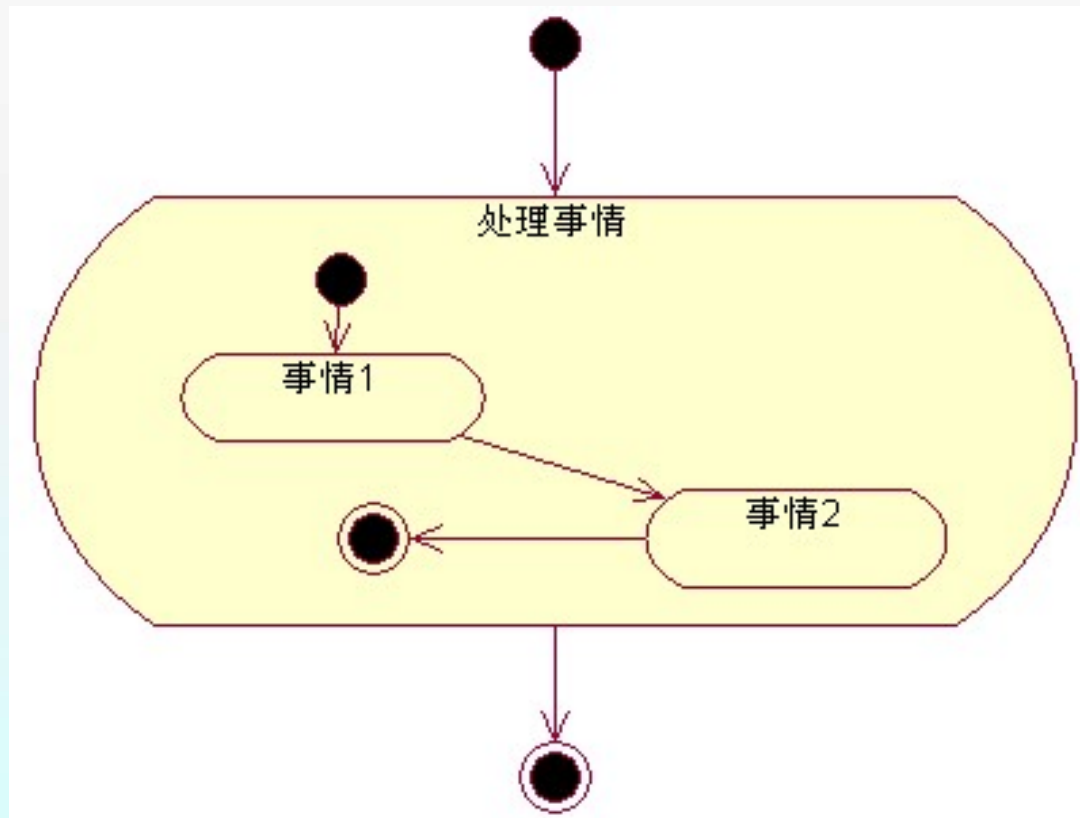
步骤：

工具——CREATE
——OBJECT



8. 组合活动

在UML的活动图中，一个大的活动可以分为若干个动作或子活动，这些动作或子活动本身又可以组成一个活动图。



活动图的用途

活动图的建模技术

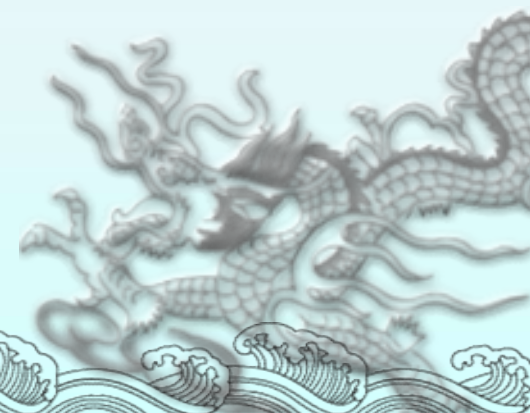
活动图用于对系统的动态行为建模，在对一个系统建模时，通常有两种使用活动图的方式：

- ◆ 一是在业务分析阶段，对工作流程进行建模；
- ◆ 二是在系统分析和设计阶段，对操作流程进行建模。



内容提纲

- 状态图
 - 相关概念
 - 实例讲解
- 活动图
 - 相关概念
 - 实例讲解



活动图创建案例

活动图建模步骤：

(1) 确定对象：找出负责实现工作流的主体对象。

对象主要参考与业务工作流程相关的用例中的参与者。

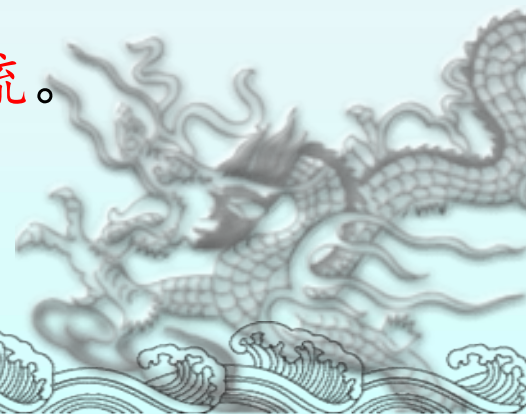
为重要的对象分配一条泳道。

(2) 确定范围：明确起始状态和终止状态。

(3) 确定主要活动序列：建模主要用例路径。

(4) 处理迁移：分支状态迁移，并发与同步（分叉与结合）迁移。

(5) 确定工作流中涉及的重要对象：添加对象流。

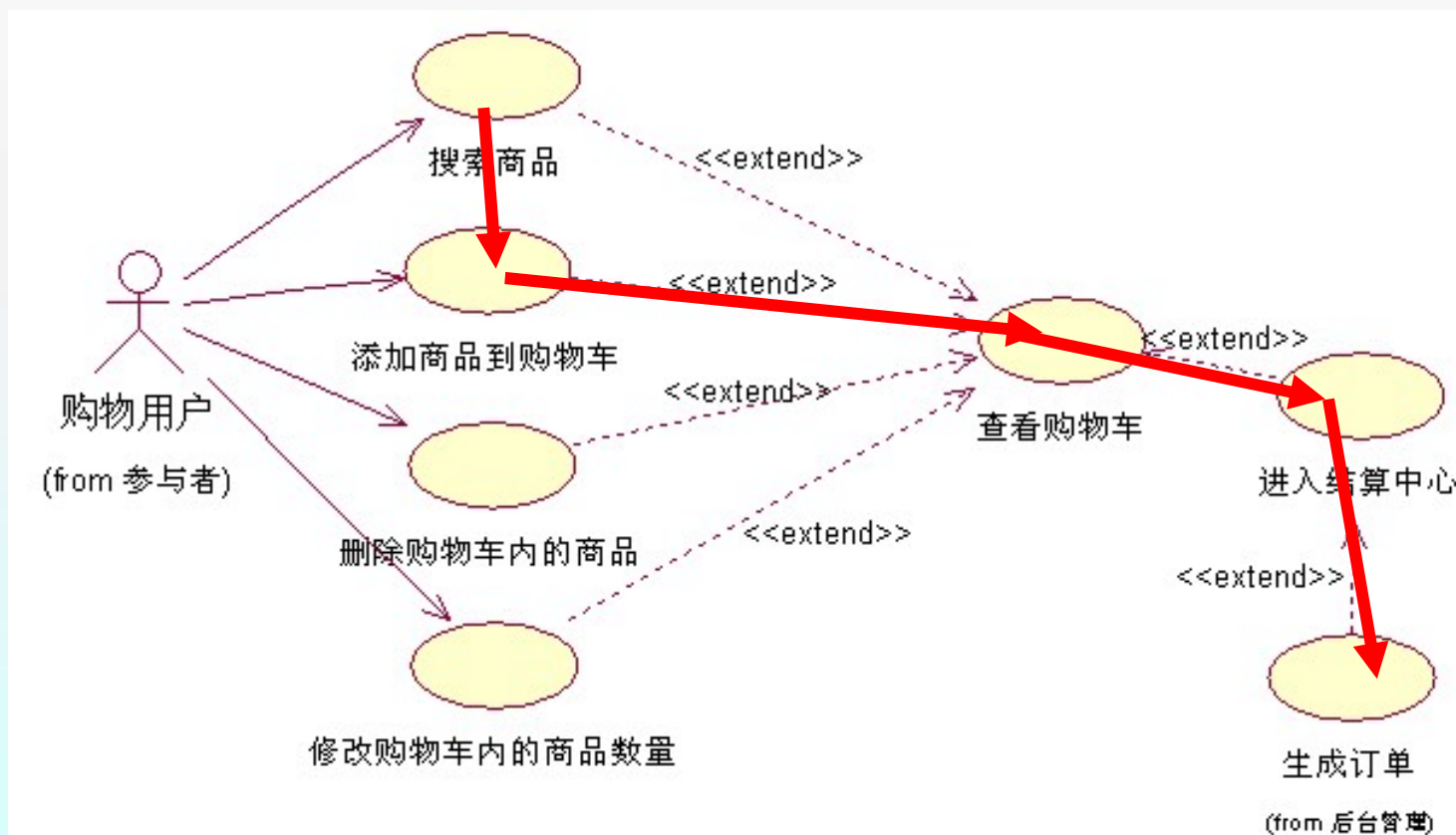


活动图创建案例

——电子商城用户购物活动图

1. 标识活动图的用例

用户购物活动中要涉及多个用例，完成工作流的主要对象是购物用户。

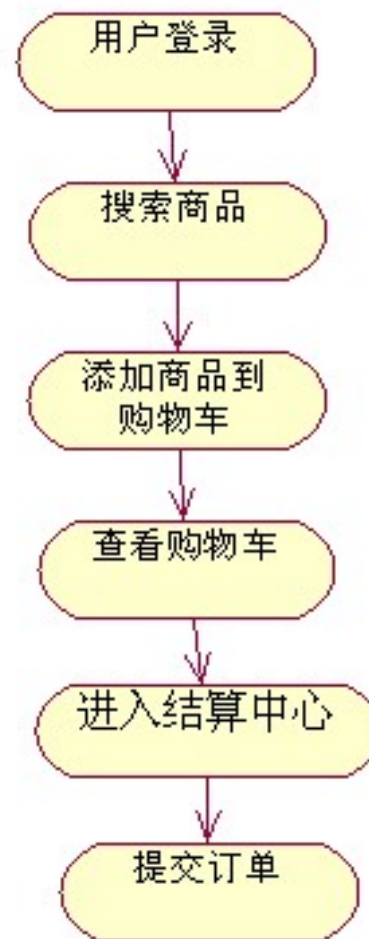


活动图创建案例

——电子商城用户购物活动图

2. 建模用例的路径

用户登录，首先选择要查看商品信息，...,最后退出系统。



活动图创建案例

——电子商城用户购物活动图

3. 创建活动图

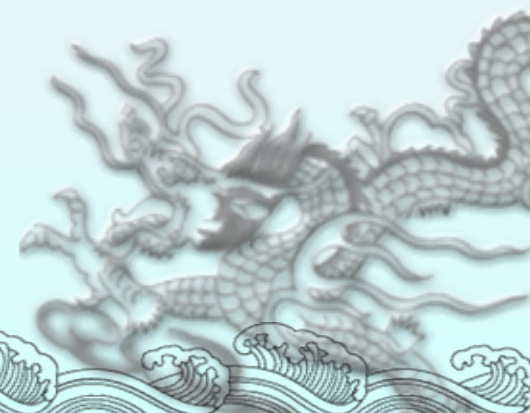
在创建活动图的过程中，需要注意如下问题：

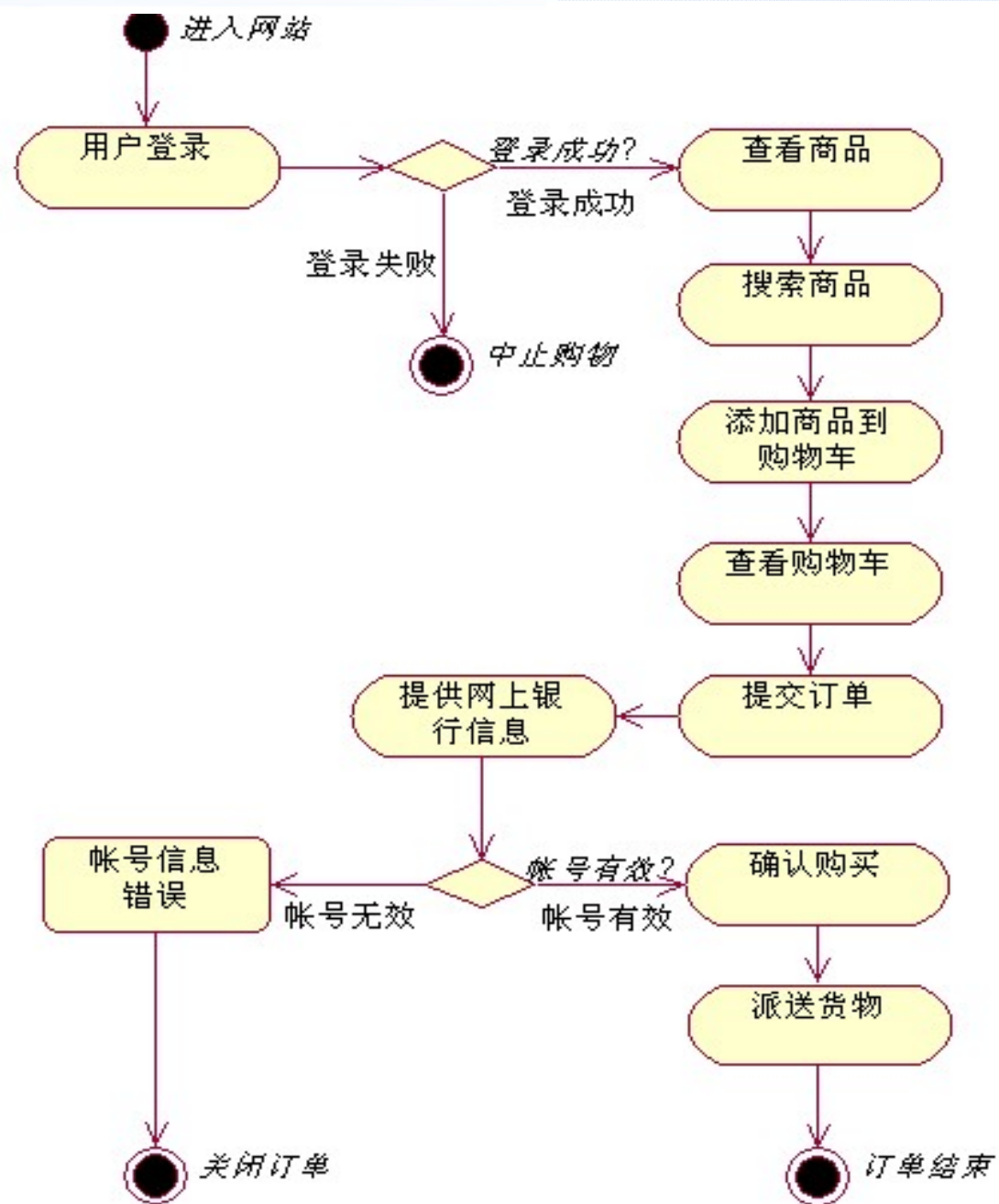
考虑用例其他可能的工作流情况，如执行过程中**可能出现的错误或是可能执行其他活动。**

使用泳道细化活动图。

按照时间顺序自上而下的**排列泳道内的动作或者状态。**

使用并发时，不要漏掉任何的**分支**，尤其是当分支比较多的时候。





活动图创建案例

——教师管理学生信息活动图

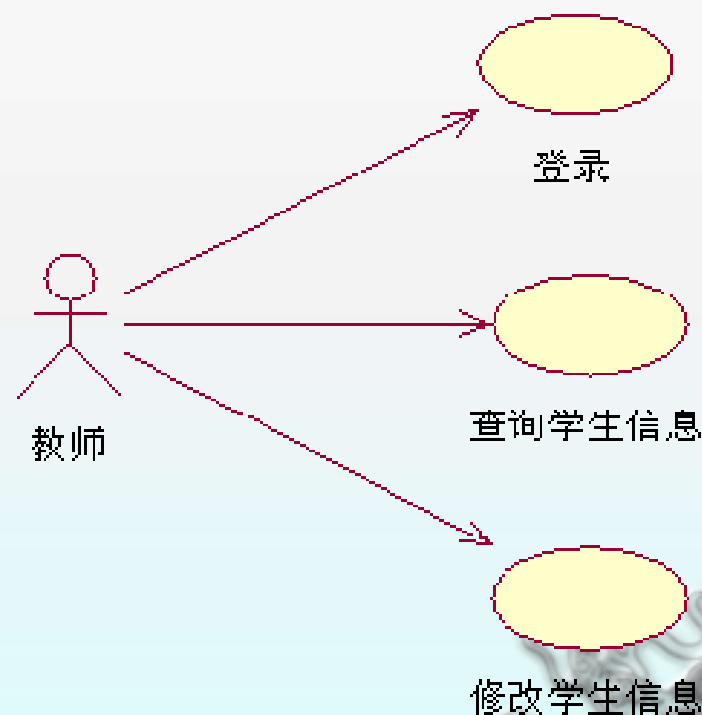
1. 标识活动图的用例

对于“教师查看、修改学生信息”来说，它的参与者是教师，教师在查看、修改学生信息的活动中有以下3个用例：

登录；

查询学生信息；

修改学生信息。

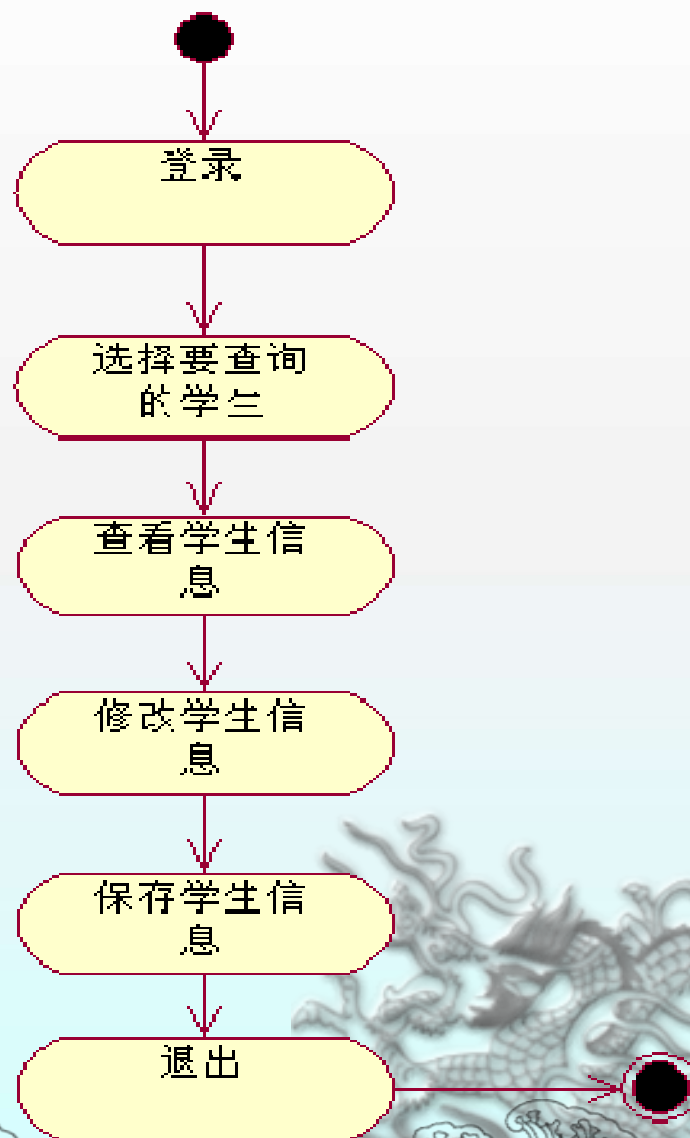


活动图创建案例

——教师管理学生信息活动图

2. 建模用例的路径

教师登录后，首先选择要查看哪位同学的信息，查看之后修改该学生的信息，修改完成后保存修改过的信息，最后退出系统。



活动图创建案例

——教师管理学生信息活动图

3. 创建活动图

在创建活动图的过程中，需要注意如下问题：

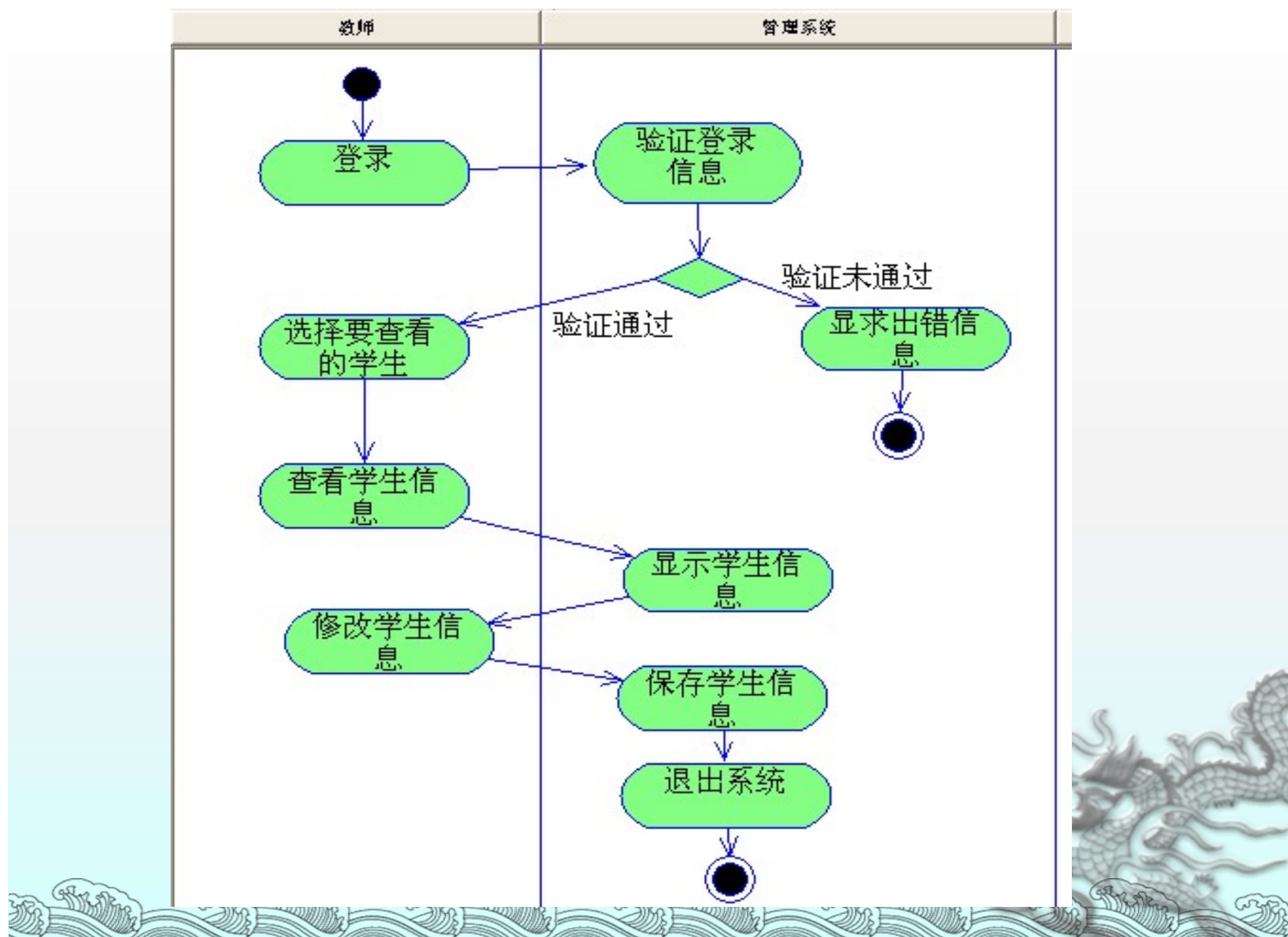
考虑用例其他可能的工作流情况，如执行过程中**可能出现的错误或是可能执行其他活动**。

使用泳道细化活动图。

按照时间顺序自上而下的**排列泳道内的动作或者状态**。

使用并发时，不要漏掉任何的**分支**，尤其是当分支比较多的时候。

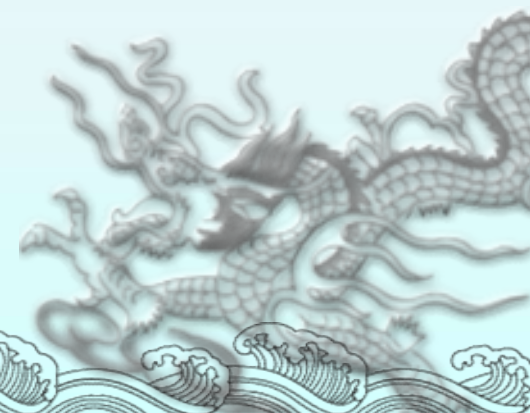




活动图与状态图的比较

状态图描述了一个特定对象的**所有可能状态**，以及由于各种事件的发生而引起的**状态之间的转移**；用来描述一个对象在其生命周期的行为，主要**强调外部动作**的影响。

活动图是一种描述工作流的方式，它用来描述采取何种动作、做什么、何时发生以及在何处发生。活动图是由状态图扩展而来的，是状态图的另外一种表现形式，它们之间有联系有区别。

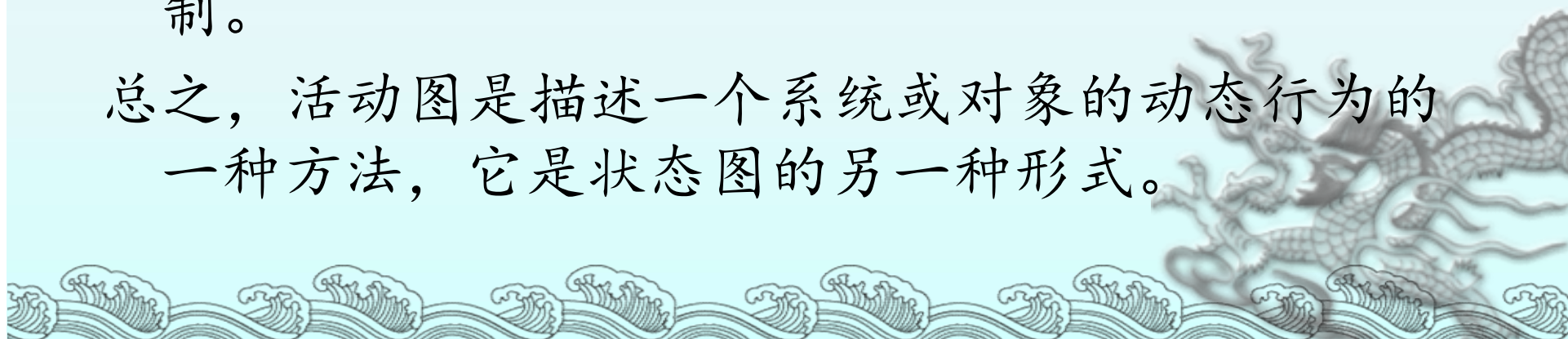


活动图与状态图的比较

状态图和活动图的相同点：

1. 描述图符基本一样；
2. 都可以描述一个系统和对象在生存期间的状态和行为；
3. 都可以描述系统或对象在多线程操作中的同步和并发行为；
4. 都可以采用分支流描述一个系统或对象的行为控制。

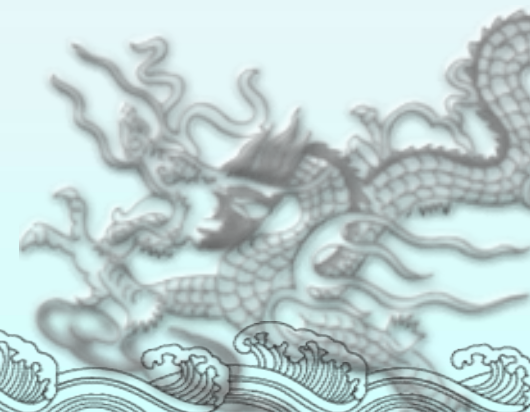
总之，活动图是描述一个系统或对象的动态行为的一种方法，它是状态图的另一种形式。



活动图与状态图的比较

状态图和活动图的主要区别在于：

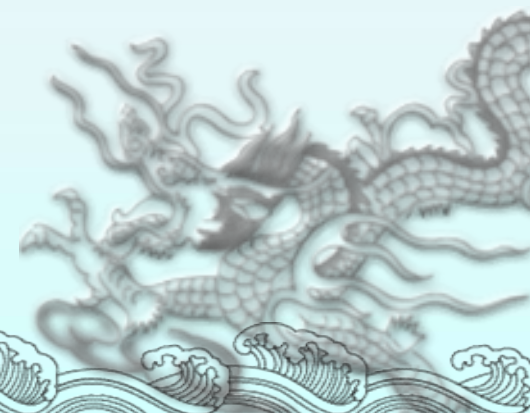
- 1、两者描述的重点不同。状态图描述的是对象的状态及状态之间的转移，而活动图描述的是从活动到活动的控制流。
- 2、两者使用的场合不同。如果是为了显示一个对象在其生命周期内的行为，则使用状态图较好；如果目的是为了分析用例，或理解涉及多个用例的工作流程，或处理多线程应用等，则使用活动图较好。



练习1：画出ATM自动取款机“取款”用例的活动图

事件路径：

1. 用户选择取款
2. 系统要求输入取款金额
3. 用户输入取款金额
4. 系统验证取款金额
 - 4a 余额不足
 - 4a1 系统显示余额不足
 - 4a2 用例终止
5. 系统出款
6. 系统询问是否打印凭条
7. 用户选择不打印凭条
8. 系统显示取款成功



◇例子：买票

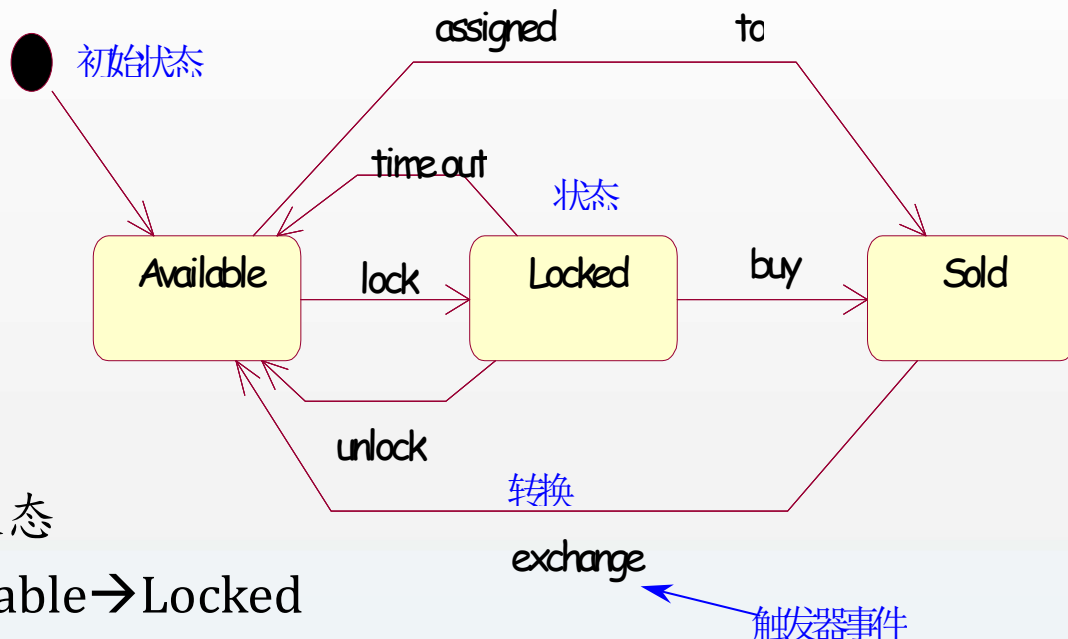
(1) 对象的状态图

图中包含以下状态

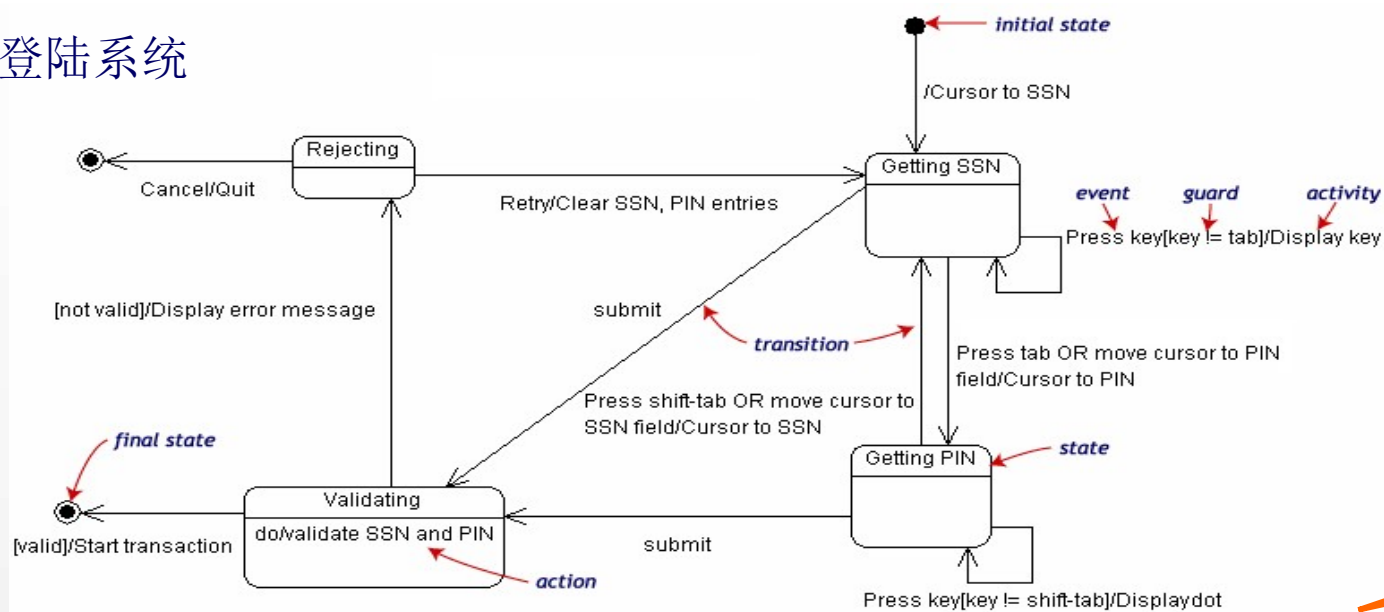
- ◆初始状态
- ◆Available状态
- ◆Locked状态
- ◆Sold状态

◇ 状态间的转移

- ◆初始状态→Available状态
- ◆票被预订(lock): Available→Locked
- ◆预定后付款(buy): Locked→Sold
- ◆预定解除(unlock): Locked→Available
- ◆预定过期(time out): Locked→Available
- ◆直接购买(assigned to): Available→Sold
- ◆换其它票(exchang), 该票重有效:
Sold→Available



网上银行登陆系统



登陆要求提交个人社会保险号(SSN)和密码(PIN)经验证有效后登陆成功。

登陆过程包括以下状态:

- ※初态(Initial state)
- ※获取社会保险号状态(Getting SSN)
- ※获取密码状态(Getting PIN)
- ※验证状态(Validating)
- ※拒绝状态(Rejecting)
- ※终态 (Final state)

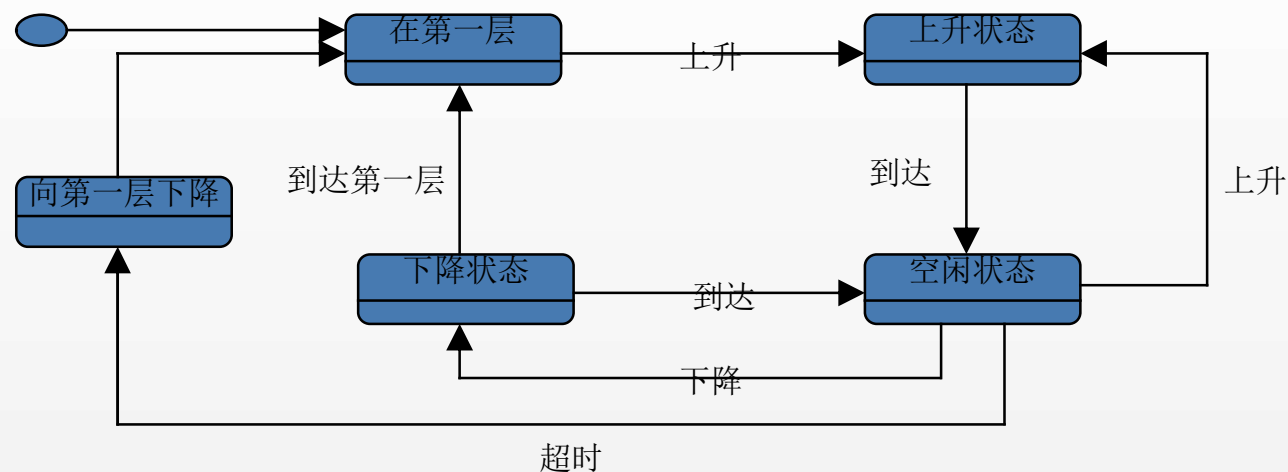
出发状态	动作	到达状态
Initial state	移动鼠标到 SSN	Getting SSN
Getting SSN	键入非tab键, 显示键入内容	Getting SSN
	键入tab键, 或移动鼠标到BIN	Getting PIN
	提交	Validating
Getting PIN	键入非shift-tab键, 显示 “ * ”	Getting PIN
	键入shift-tab键, 或移动鼠标到SSN	Getting SSN
	提交	Validating
Validating	验证提交信息有效, 状态转移	Final state
	验证提交信息无效, 显示错误信息	Rejecting
Rejecting	退出	Final state
	重试, 清除无效的SSN, PIN	Getting SSN

状态转移的过程

有两个不同的终态

◆ 练习

◆ 分析下面的状态图，回答问题



(1) 以下那些图形元素是对状态的描述？

(a) 超时

(b) 到达

(c) 在第一层

(2) 空闲状态 超时后转移到____状态

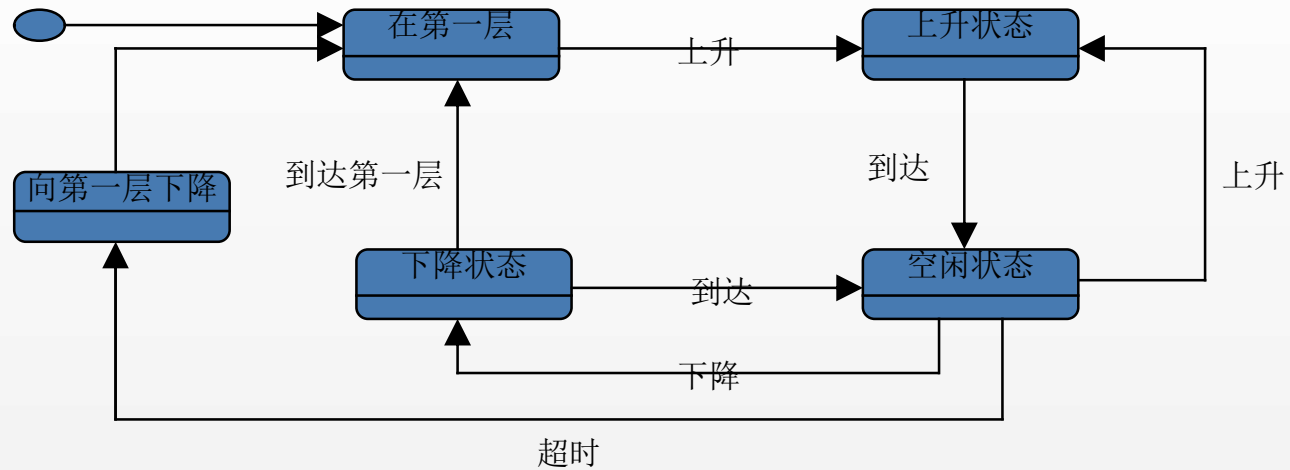
(a) 向第一层下降

(b) 上升状态

(c) 终态

◆ 练习

◆ 分析下面的状态图，回答问题



(1) 以下那些图形元素是对状态的描述? c

(a) 超时 (b) 到达 (c) 在第一层

(2) 空闲状态 超时后转移到____状态a

(a) 向第一层下降 (b) 上升状态 (c) 终态

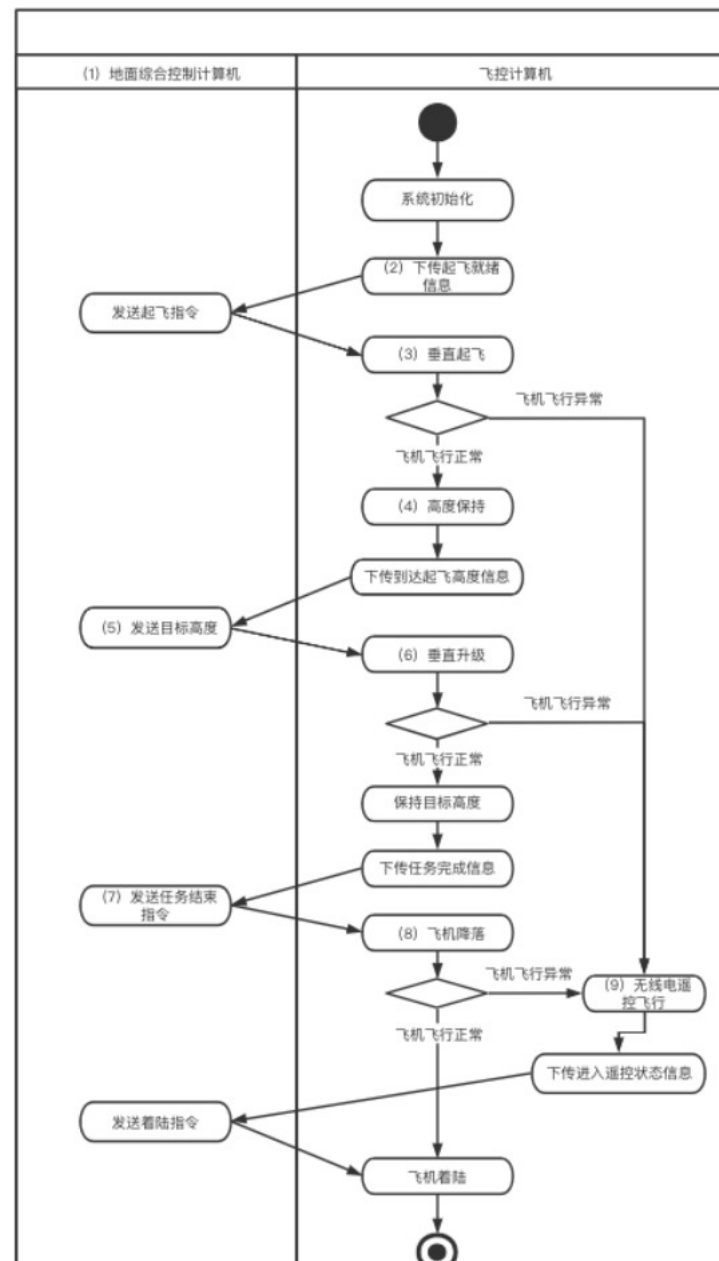
习题答案

(1) (c)
(2) (a)

练习： 某公司拟研制一款高空监视无人直升机，该无人机采用遥控—自主复合型控制实现垂直升降。该直升机飞行控制系统由机上部分和地面部分组成，机上部分主要包括无线电传输设备、飞控计算机、导航设备等，地面部分包括遥控操纵设备、无线电传输设备以及地面综合控制计算机等。其主要工作原理是地面综合控制计算机负责发送相应指令，飞控计算机按照预定程序实现相应功能。经过需求分析，对该无人直升机控制系统纵向控制基本功能整理如下：

- (a) 飞控计算机加电后，应完成系统初始化，飞机进入准备起飞状态；
- (b) 在准备起飞状态中等待地面综合控制计算机发送起飞指令，飞控计算机接收到起飞指令后，进入垂直起飞状态；
- (c) 垂直起飞过程中如果飞控计算机发现飞机飞行异常，飞行控制系统应转入无线电遥控飞行状态，地面综合控制计算机发送遥控指令；
- (d) 垂直起飞达到预定起飞高度后，飞机应进入高度保持状态；
- (e) 飞控计算机在收到地面综合控制计算机发送的目标高度后，飞机应进入垂直升降状态，接近目标高度；垂直升降过程中出现飞机飞行异常，控制系统应转入无线电遥控飞行；
- (f) 飞机到达目标高度后，应进入高度保持状态，完成相应的任务；
- (g) 飞机在接到地面综合控制计算机发送的任务执行结束指令后，进入飞机降落状态；
- (h) 飞机降落过程中如果出现飞机飞行异常，控制系统应转入无线电遥控飞行；
- (i) 飞机降落到指定着陆高度后，进入飞机着陆状态，应按照预定着陆算法，进行着陆；
- (j) 无线电遥控飞行中，地面综合控制计算机发送着陆指令，飞机进入着陆状态，应按照预定着陆算法，进行着陆。

练习：



作业：

- 1、什么是状态图？其包含的建模元素是什么？
- 2、UML中的事件分为哪4类？请分别阐述。
- 3、简述活动图的分支与分叉。
- 4、简述状态图与活动图的异同。

