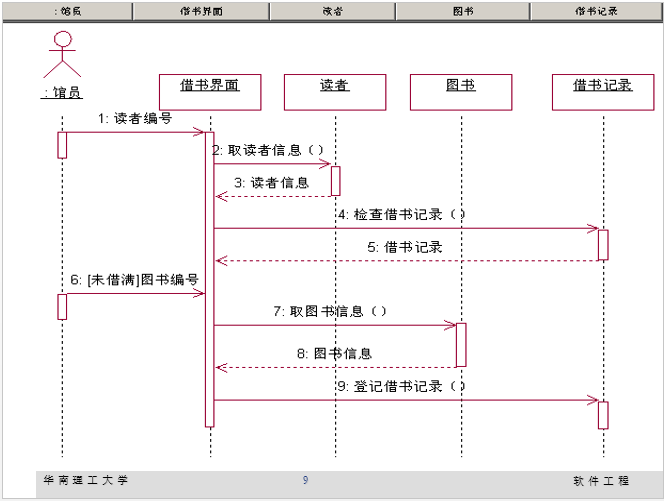
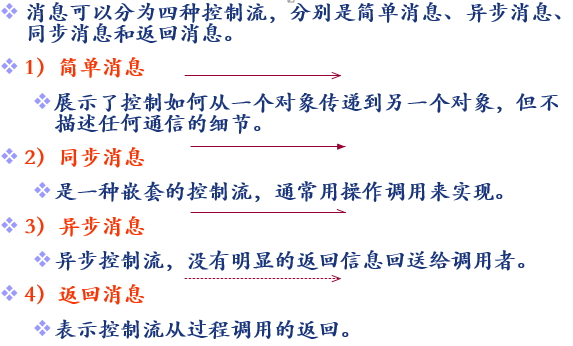
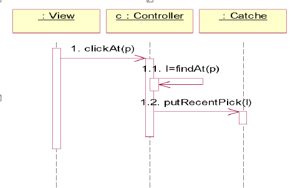
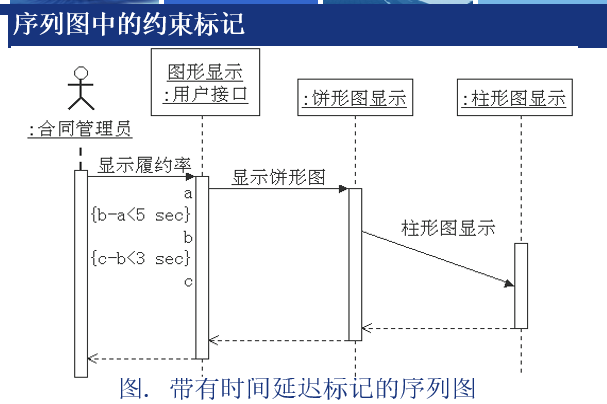
**序列图组成部分**

* **（1）对象：序列图中所包含的每个对象用一个对象框（短式）表示，对象名需带下划线。**
* **（2）生存线：对象框下画的一条垂直虚线，称为该对象的生存线，表示对象的生存时间。**
* **（3）激活期：对象生存线上的一个细长方形框，**
* **表示该对象的激活时间段，即活动期间。**
* **（4）消息：对象之间消息的发送和接收用两个 对象生存线（激活期）之间的消息箭头线。**

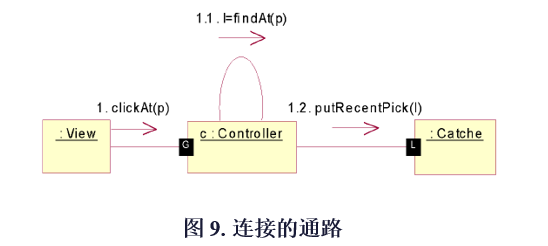






* **建立序列图**
* **① 从用例中识别交互过程;**
* **② 识别参与交互过程的对象;**
* **③ 为每一个对象设置生命线,并确定对象的存在期限;**
* **④ 从引发交互的初始消息开始,在对象生命线上依次画出交互的消息;**
* **⑤如果需要,可以给消息增加时间约束,以及前置条件和后置条件。**
* 

协作图



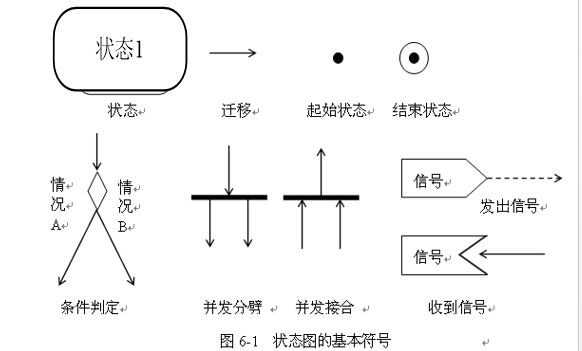
* **建立协作图**
* **① 从用例中识别交互过程;**
* **② 识别参与交互过程的对象;**
* **③ 确定对象之间的链，以及链上的消息;**
* **④ 从引发交互的初始消息开始,将随后每个消息附在相应的链上;**
* **⑤ 如果需要,可以给消息增加时间约束,以及前置条件和后置条件。**

**顺序图的建模分析步骤**

* **(1)完成用例图的分析；**
* **(2)对每个用例，识别出参与基本事件流的对象(包括接口、子系统、角色等)。**
* **(3)识别出这些对象是主动对象还是被动对象。**
* **(4)识别出这些对象发出的消息是同步消息还是异步消息。**
* **(5)从主动对象开始向接收对象发消息。**
* **(6)接收对象再调用自己的服务为主动对象返回结果。**
* **(7)如果接收对象需要再调用其他对象的服务，需要向其他对象再发消息。**
* **(8)如此反复，最后返回给主动对象有意义的结果。**
* **(9)用UML建模工具绘出顺序图。**
* **(10)给顺序图补充必要的说明文档。**

**状态被图形化表示为一个圆角矩形（图3）**

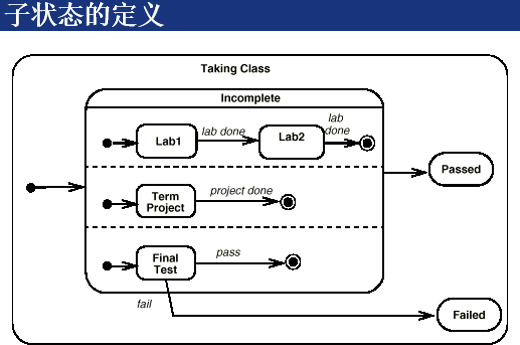


* **在UML里，变迁由五个部分构成，它们是：**
  + **起始状态**
  + **目标状态**
  + **触发事件(event trigger)**
  + **触发条件(guard condition)**
  + **变迁动作。**
* **在绘制变迁时，变迁的触发事件、触发条件和变迁动作被表达成一个字符串被放置在表示变迁的箭头上。此字符串又被称为变迁的文字标记（text label）。变迁的文字标记的格式如下：**
* **触发事件[触发条件]/变迁动作**
* **描述状态图的图符元素有:状态图符、迁移图符、起始状态、终止状态、条件判定、发出信号、接收信号和并发等。**
* 
* **一个完整的状态包括三个组成部分，它们是：**
  + **状态名**
  + **状态变量**
  + **活动**

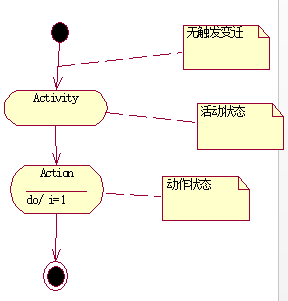
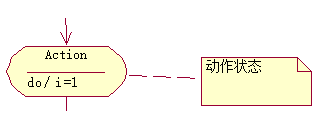
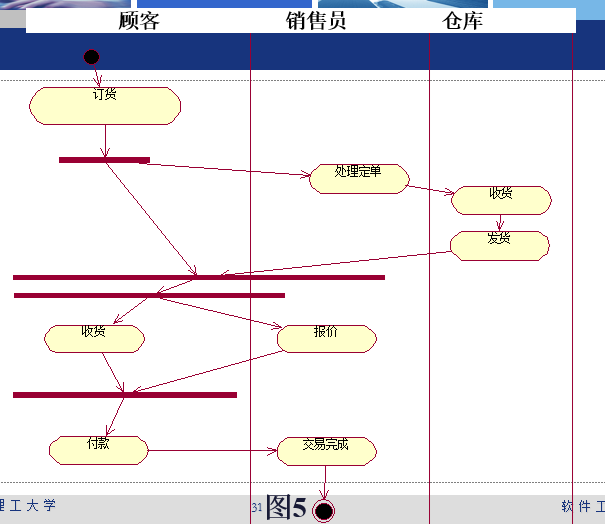
**状态名**

**状态变量**

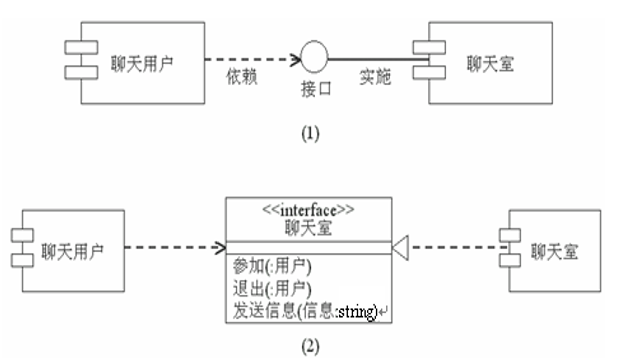
**活动**

* **在UML里，一个状态内部可以包含其它状态，它们在此状态的内部构成了另一个状态机。**
* 
* **动作：一组可执行的语句，具有迁移性、原子性和连续性。**
* **活动：一组可执行的动作，具有有限性和非原子性。**
  + **浅层历史状态只记忆被转出变迁打断时最外层的活跃状态。**
* **如果需要历史状态记忆最深层的内嵌活跃子状态（即不再包含子状态的活跃子状态），则应使用**
* **深层历史状态（deep history state）。**
  + **深层历史状态在绘制时用一个被圆环包围的带星号的字母H表示（H\*, 见图5）。**
* **自动转移: 一个 状态根据本状态的有关情况，自动触发进入目标状态，在转移上没有事件。**

**状态图的建模分析步骤**

* **（1）首先要确定进行系统控制的对象，可以从前面分析的顺序图中寻找。**
* **（2）确定对象的起始状态和结束状态。**
* **（3）在对象的整个生命周期寻找有意义的控制状态。**
* **（4）寻找状态之间的转换。**
* **（5）补充引起转换的事件。**
* **（6）UML建模工具画状态图。**
* **（7）补充必要的文档。**
* **当非反应型对象的动态行为被执行时，**
  + **动态行为的一系列的动作按照特定的控制逻辑（算法）顺序执行。**
* **活动图中的状态表示计算过程中所处的各种状态，而不是普通对象的状态**
  + **活动图着重表现活动的控制流，描述在对象之间传递的操作**
  + **交互图着重表现的是对象到对象的控制流，描述在对象之间传递的消息**
* 
  + **活动状态机的动作是自动执行的**
  + **状态机内不存在对外部事件的描述**
  + **控制在动作之间的转换不由事件触发，而是由完成变迁自动触发。**
* **活动置于责任区（泳道）中，责任区将活动按责任目标和组织归属的原则分类。**
* **活动图主要包括三个方面的内容：**
  + **动作状态**
  + **活动状态**
  + **无触发变迁**
* 
  + **分支中的输出变迁都是**
    - **带触发条件的完成变迁,**
    - **触发条件的书写格式可以是一个布尔表达式。**
* 

组件图



* + **组件是存在于物理的资源上的**
    - **存储 / 运行**
  + **需要为这样的物理资源建模**
    - **节点/部署(deployment)**
      * **节点(node)**
        + **节点是一个运行时刻的物理对象，它代表一类计算资源，该计算资源常至少具有存储功能，并且在大多数的情况下具有数据处理的能力。  
          - e.g.:计算机、外设,etc.**
      * **部署视图(deployment view)...**