

UML建模基础

——UML建模相关

东软IT人才实训中心

UML的重要内容：九种图形

UML静态图	用例图（Use Case Diagram）： 模型化系统与外界的交互
	类图（Class Diagram）： 模型化系统的结构
	对象图（Object Diagram）： 对象及对象间的相互关系
	组件图（Component Diagram）： 模型化组件的组织 and 依赖
	部署图（Deployment Diagram）： 模型化系统的硬件分布
UML动态图	时序图（Sequence Diagram）： 模型化系统的行为
	协作图（Collaboration Diagram）： 模型化系统的行为
	状态图（Statechart Diagram）： 模型化状态相关的方面
	活动图（Activity Diagram）： 模型化系统内的事件流

九种图形-1、用例图

用例图：是从用户角度描述描述**系统的边界及系统功能**，并指出各功能的**操作者**。

建立于系统需求分析阶段，是开发者和用户对系统需求达成的共识。

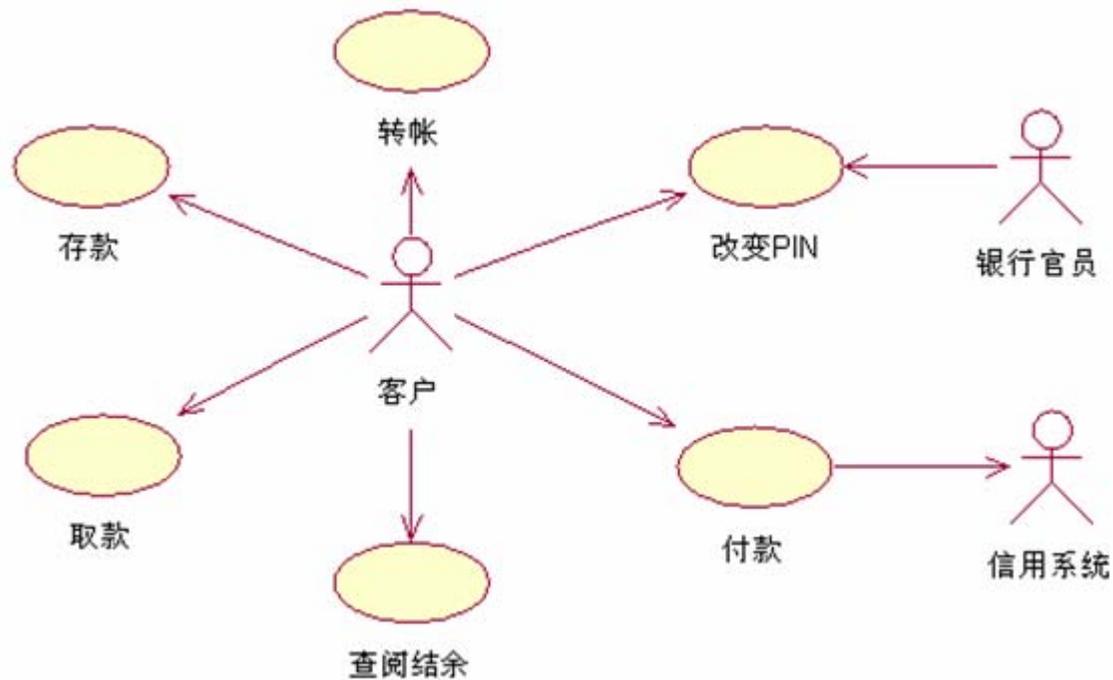
例：ATM取款

用例：描述一个系统做什么

参与者：表示用例的使用者在与这些用例交互时所扮演的角色—可以是人、硬件设备或一个系统。

九种图形-用例图示例

用例图示例

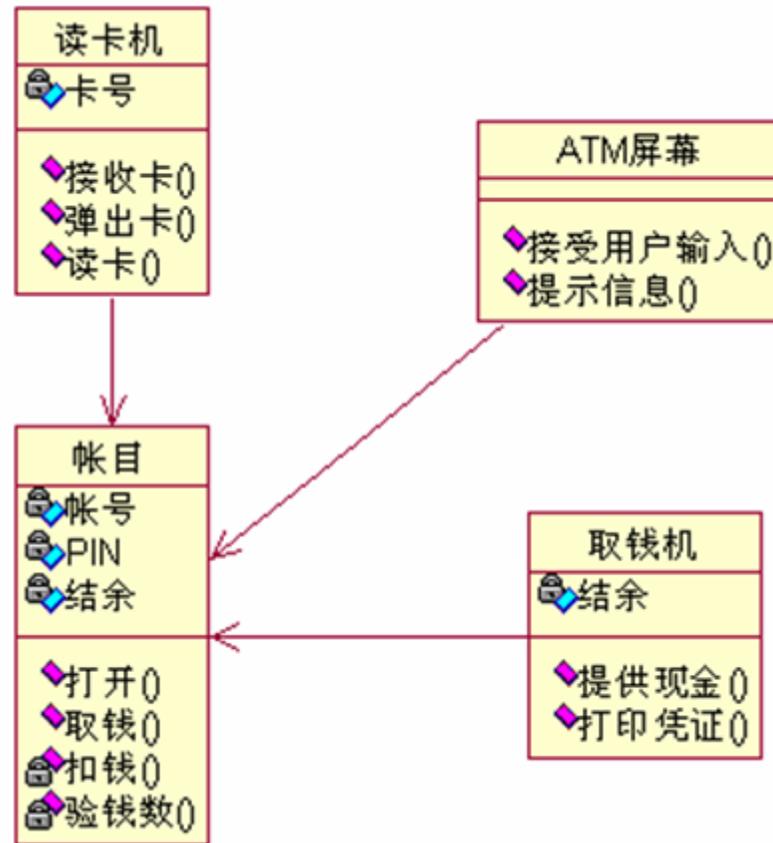


九种图形-2、类图

类图：描述系统中类的静态结构。
 不仅定义系统中的**类**，表示类之间的**联系**如关联、依赖、聚合等，也包括类的**内部结构**（类的属性和操作）。

类图描述的是一种静态关系，在系统的整个生命周期都是有效的。

例：ATM系统



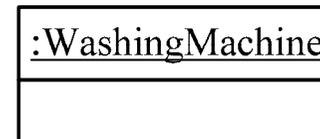
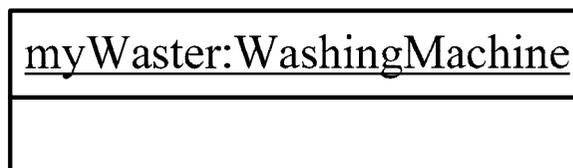
九种图形-3、对象图

对象图：是**类图的实例**，表示在**某一时刻系统对象的状态**、对象之间的**联系状态**，几乎使用与类图完全相同的标识。他们的不同点在于对象图显示类的多个对象实例，而不是实际的类。由于对象存在生命周期，因此对象图只能在系统某一时间段存在。

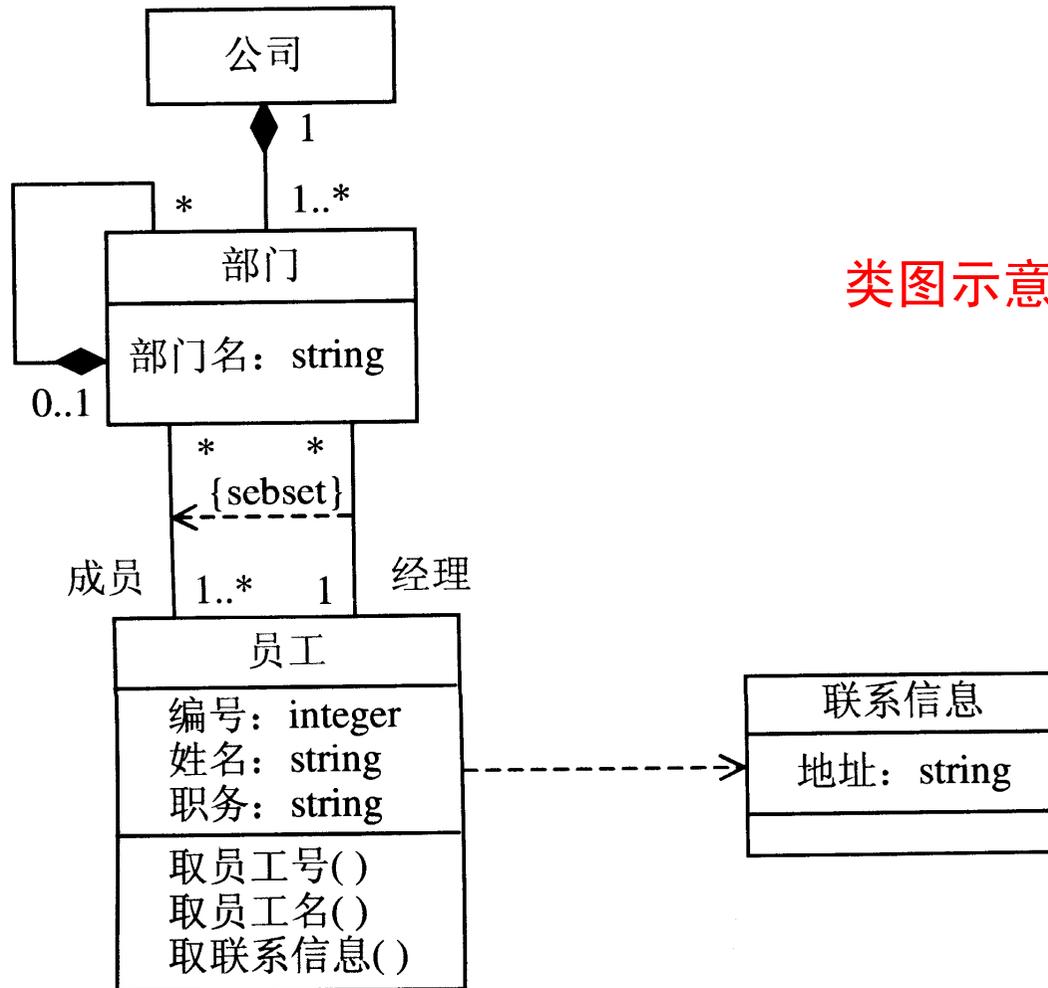
如下面：

左图指明对象和它对应的类名；

右图尽管指明了对象所属类，但并未提供一个具体的对象名，这是匿名对象。



九种图形-3、对象图



类图示意

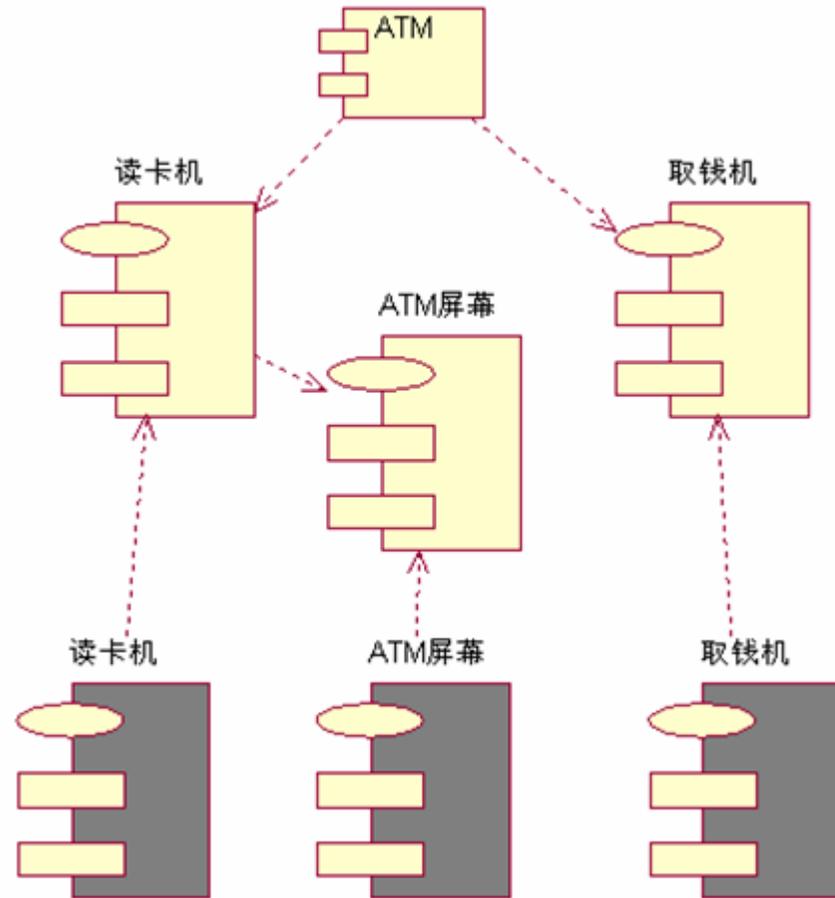
九种图形-4、组件图

组件图：也叫构件图，描述代码部件的**物理结构**及各部件之间的**依赖关系**。

一个构件可能是一个资源代码构件、一个二进制构件或一个可执行构件。

它包含逻辑类或实现类的有关信息。构件图有助于分析和理解构件之间的相互影响程度。

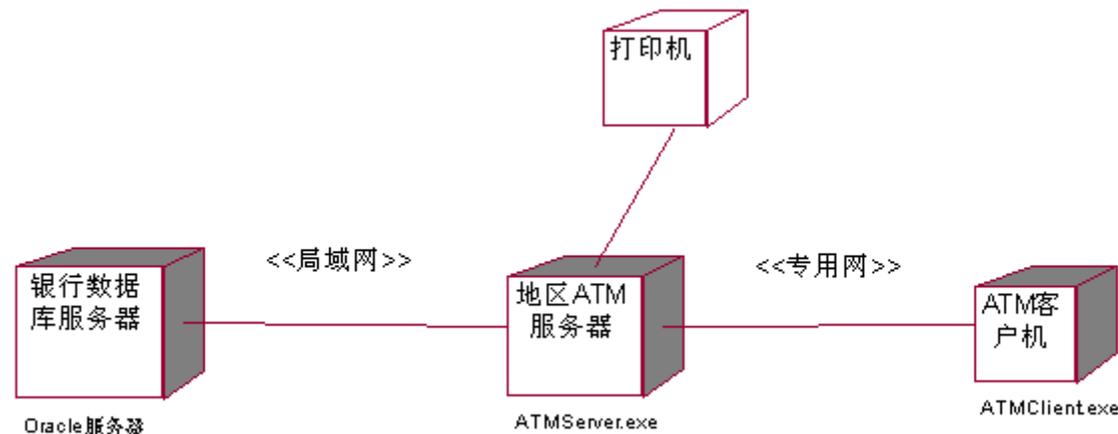
例：ATM系统



九种图形-5、部署图

部署图：也叫配置图，定义系统中软硬件的**物理体系结构**。显示实际的计算机和设备(用节点表示)以及它们之间的连结关系，也可显示连结的类型及部件之间的依赖性。
 在节点内部，放置可执行构件和对象以显示节点跟可执行软件单元的对应关系。

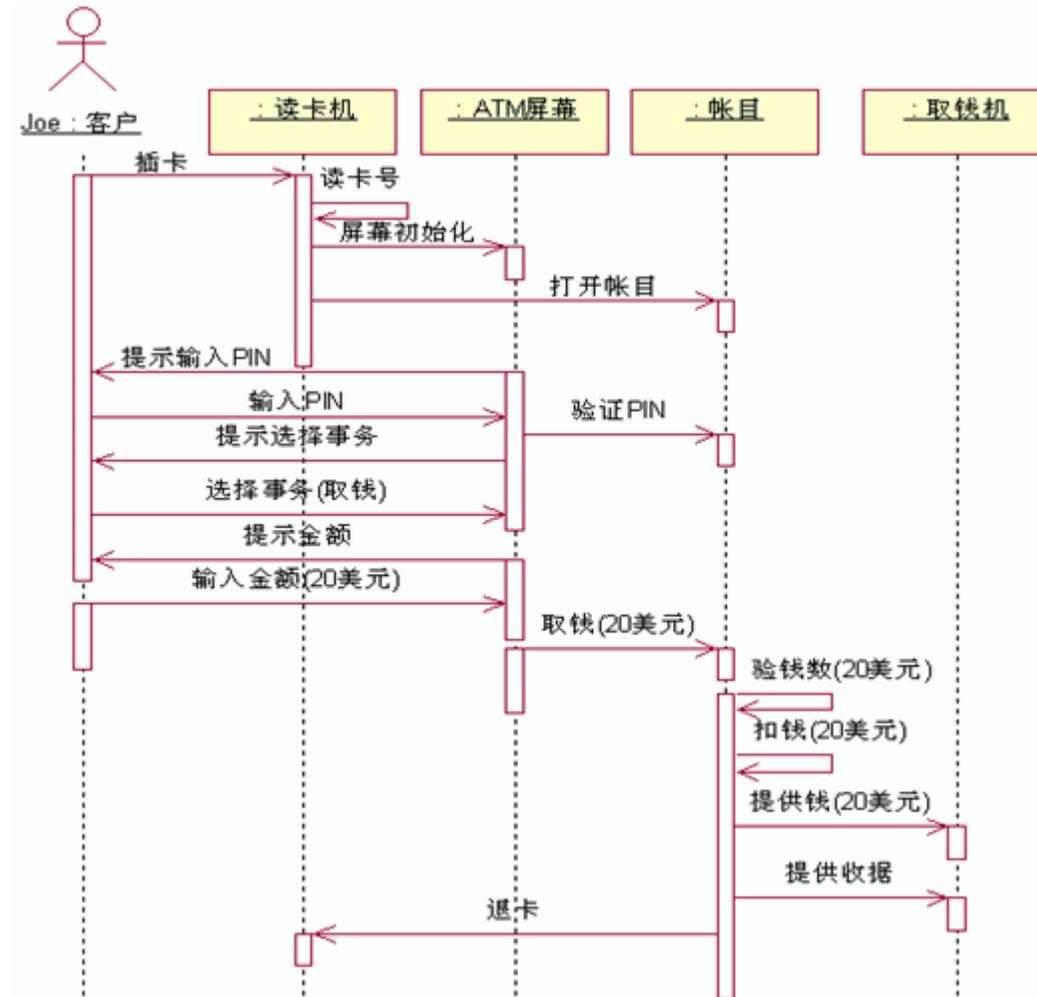
例：ATM部署图



九种图形-6、时序图

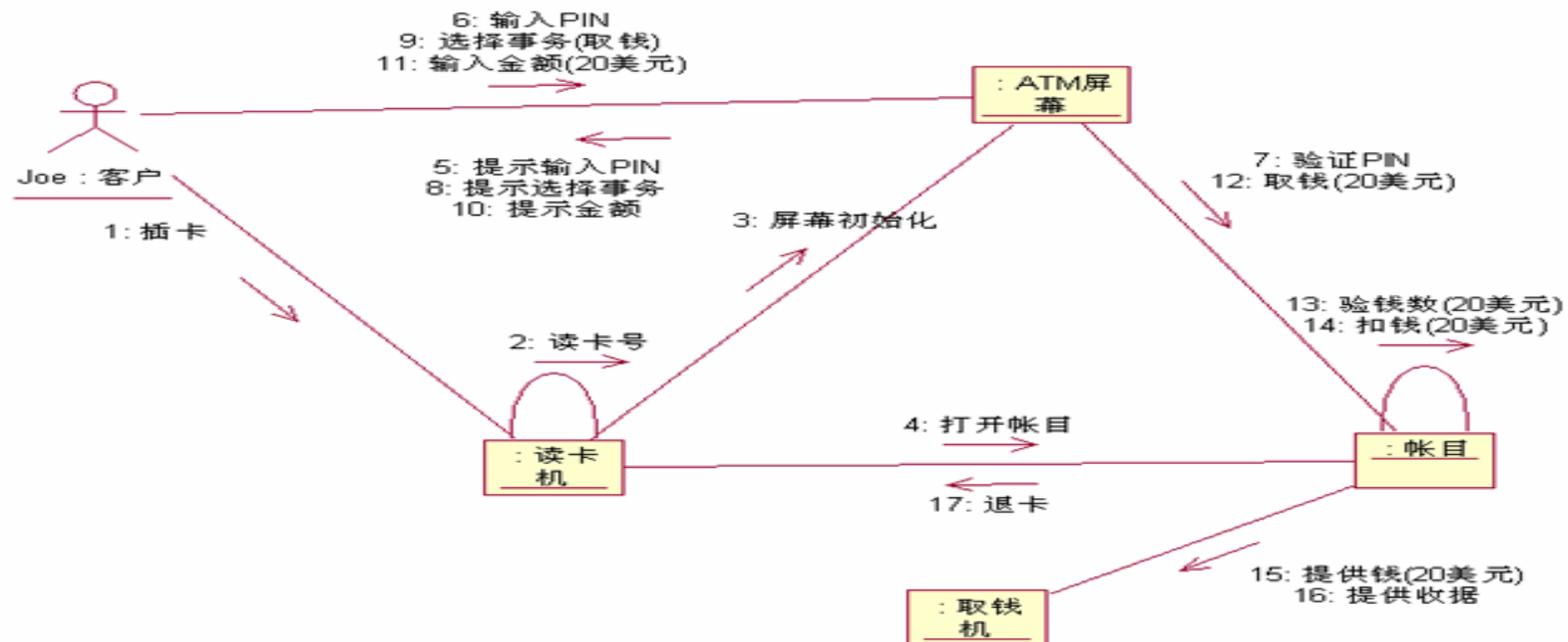
时序图：显示对象之间的动态合作关系，它强调对象之间**消息发送的顺序**，同时显示对象之间的交互。

例：ATM系统



九种图形-7、协作图

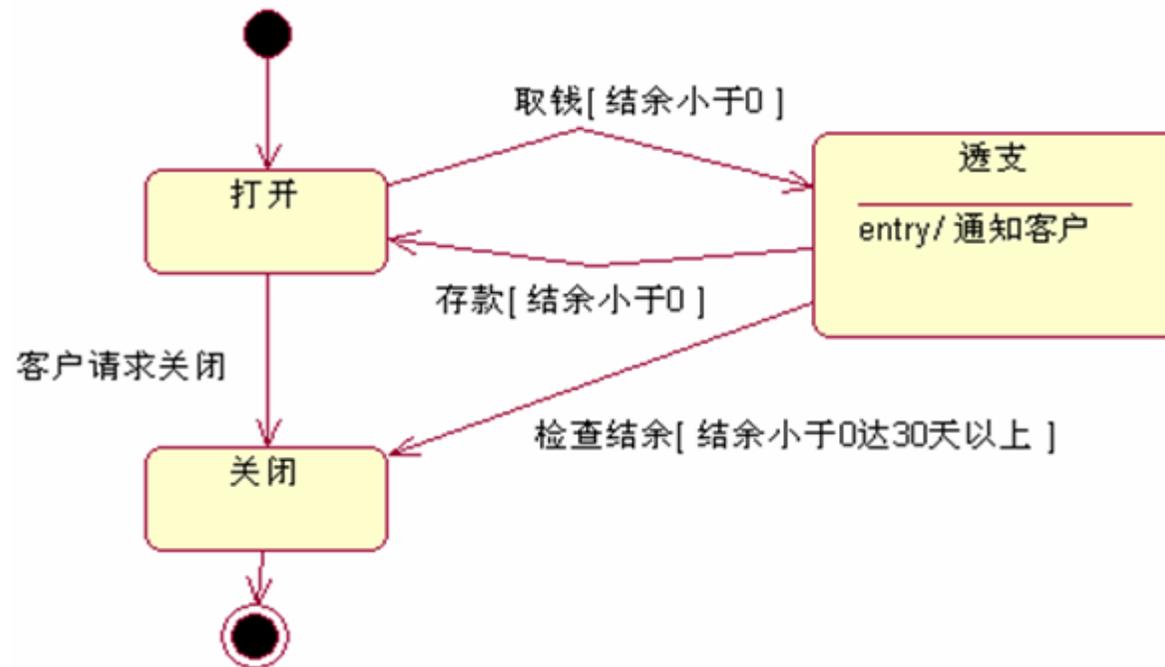
协作图：描述对象间的协作关系。协作图跟时序图相似，显示对象间的动态合作关系。除显示信息交换外，协作图还显示对象以及它们之间的关系。



九种图形-8、状态图

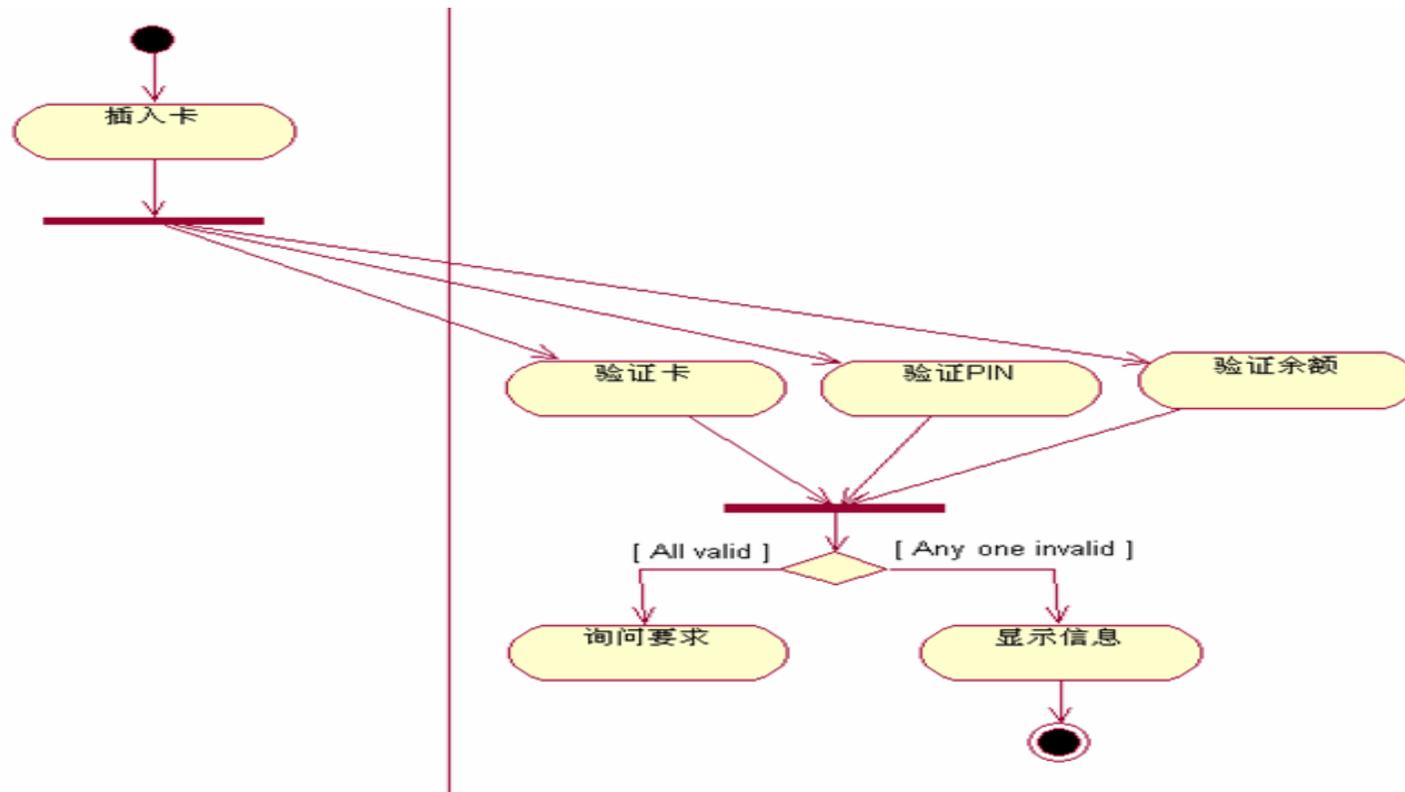
状态图：描述状态转换，是**对类图的补充**，描述了特定对象的**所有可能状态**。

在使用上并不需要为所有的类画状态图，仅为那些有多个状态并且其行为受外界环境的影响并且发生改变的类画状态图。



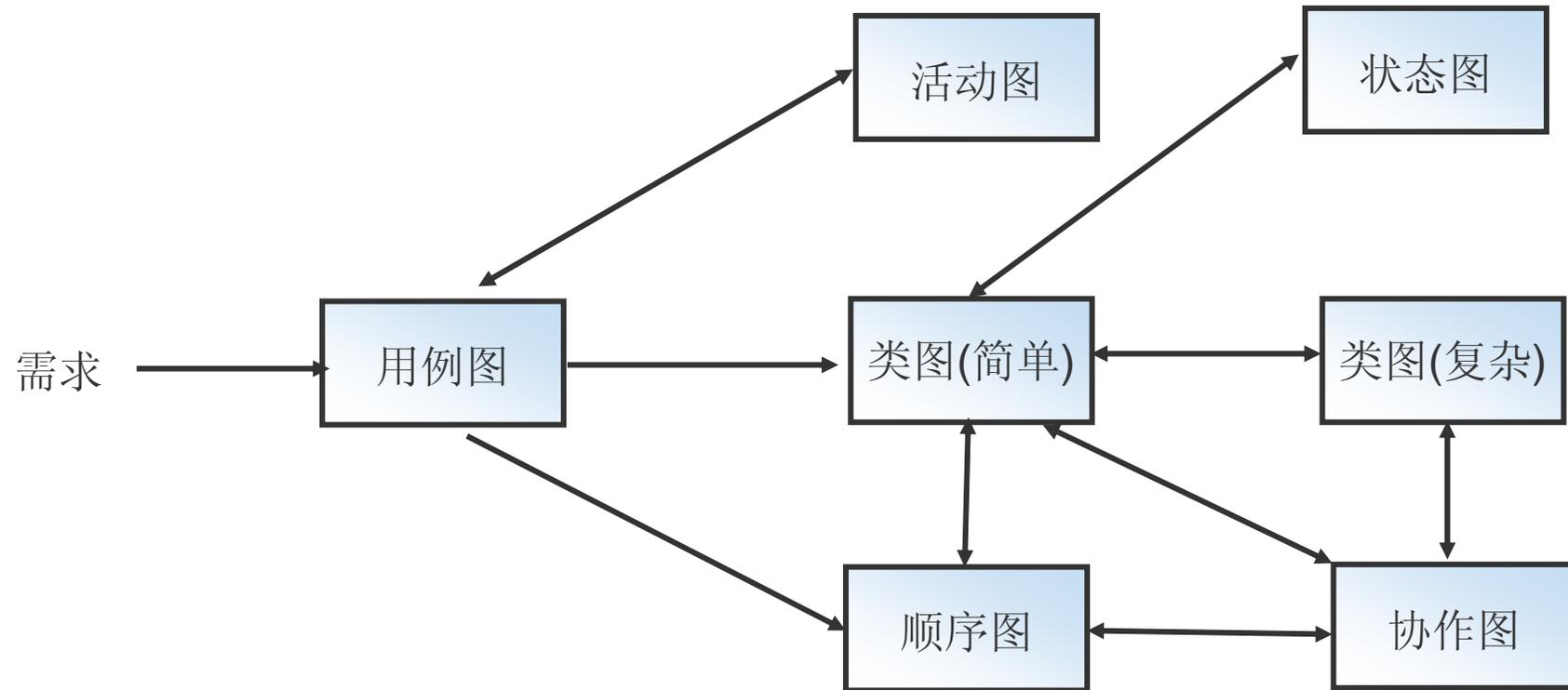
九种图形-9、活动图

活动图：描述满足用例要求所要进行的**活动**以及活动间的约束关系，有利于识别并行的活动。



图形关系

几种图形的关系



图形应用

使用UML进行项目的分析和设计时，一般遵循的步骤是

- 1、描述需求，产生用例图。
- 2、根据需求建立系统的静态模型，构造系统的结构。这个步骤产生：类图、对象图、组件图和部署图
- 3、描述系统的行为，产生状态图，活动图，顺序图和协作图。

实际开发中UML遵循的流程

- 1.识别Actor: Actor是和系统打交道最直接的人和模块。
- 2.识别use case: 行为
- 3.可以使用activity图分析特定的use case
- 4.根据use case 和document抽象出类
- 5.根据类和use case的特定场景制作时序图
- 6.根据Sequence完善类图的方法
- 7.使用State chart说明关键类的状态
- 8.使用Component图说明程序的逻辑组织
- 9.使用Deploy说明系统的部署

Uml运用于开发过程（总结）

- 1.完成use case diagram&document
- 2.对于比较复杂的use case用activity图辅助说明
- 3.根据use case document和activity diagram，分析业务领域的概念，抽象出概念模型
- 4.分析概念模型，抽象出类
- 5.分析类的职责和关系，作出类图
- 6.根据类图制作sequence， collaboration， state chart等各种图示
- 7.重复迭代5-6，直至“完美”
- 8.Coding
- 9.测试并修改，有必要的话从前面的步骤开始改起，或从1或从5
- 10.部署并和用户一起进行测试
- 11.从实施和测试的反馈驱动下一次的1-11

如何应用UML

如何应用uml建模
如何根据uml编程

简单图例：三国时期-借东风



刘备

孔明

关羽

张飞

求战

请拟策略

请防守荆州

请联络孙权

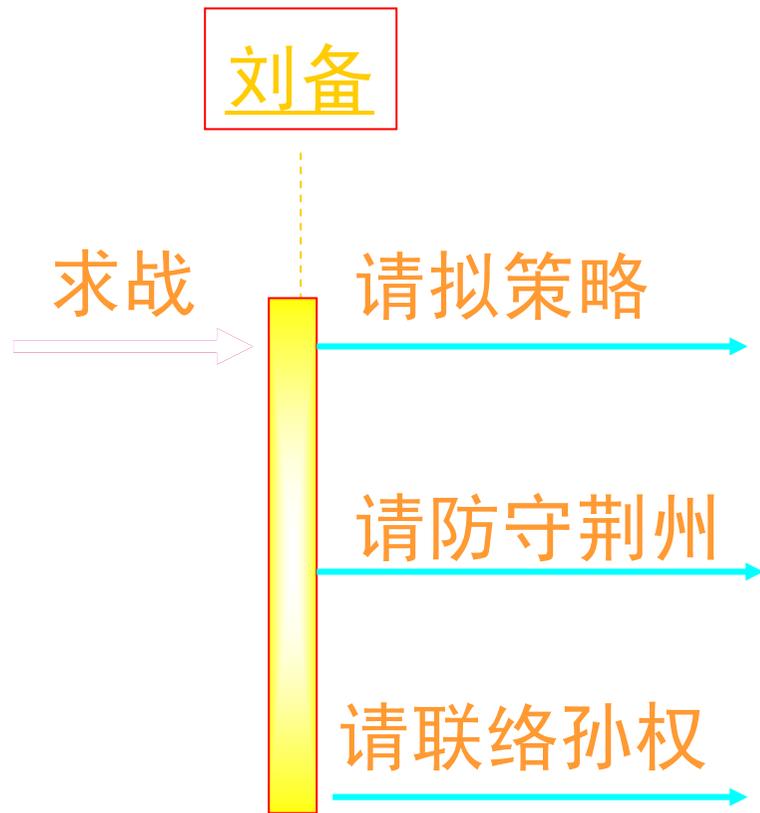
火攻曹军

借东风火攻

请防守
荆州前线

请孙权领兵相助

刘备



孔明

请拟策略



请联络孙权



请孙权领兵相助



火攻曹軍

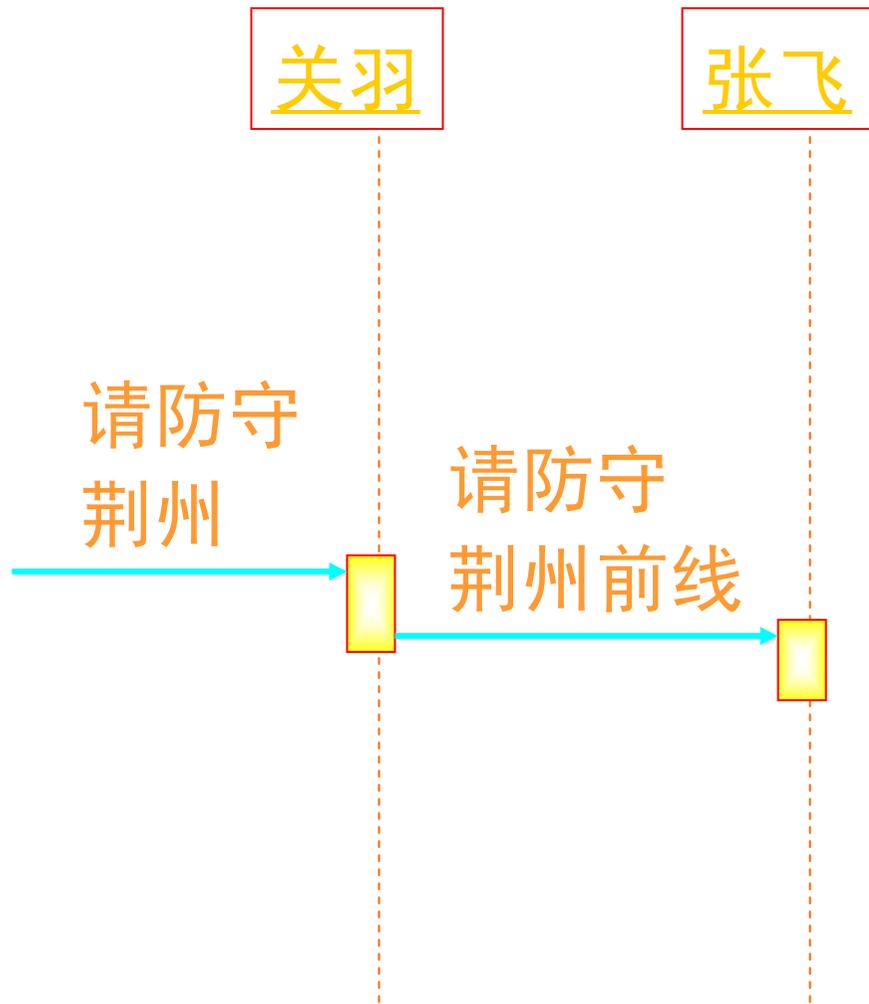


借东风火攻



孔明

- 拟定策略
- 联合孙权
- 借东风火攻



关羽

- 防守荆州

张飞

- 防守荆州前线

```
public class 刘备{  
    public 迎战曹操()  
    {  
        .....  
    }  
}
```

```
public class 关羽{  
    public 防守荊州()  
    {  
        .....  
    }  
}
```

```
public class 张飞{  
    public 防守前线()  
    {  
        .....  
    }  
}
```

```
public class 孔明{  
    public 拟定策略()  
    {  
        .....  
    }  
    public 联合孙权()  
    {  
        .....  
    }  
    private 借东风火攻()  
    {  
        .....  
    }  
}
```

```
public class 刘备{  
    孔明 k = new 孔明  
    关羽 g = new 关羽  
    public 迎战曹操()  
    {  
        k. 拟定策略  
        g. 防守荆州  
        k. 联合孙权  
    }  
}
```

```
class 孔明{  
    孙权 s = new 孙权  
    public 拟定策略()  
    {  
        .....  
    }  
    public 联合孙权()  
    {  
        s. 请领兵相助  
        借东风火攻()  
    }  
    private 借东风火攻()  
    {  
        .....  
    }  
}
```

谢谢！

Neusoft

Beyond Technology