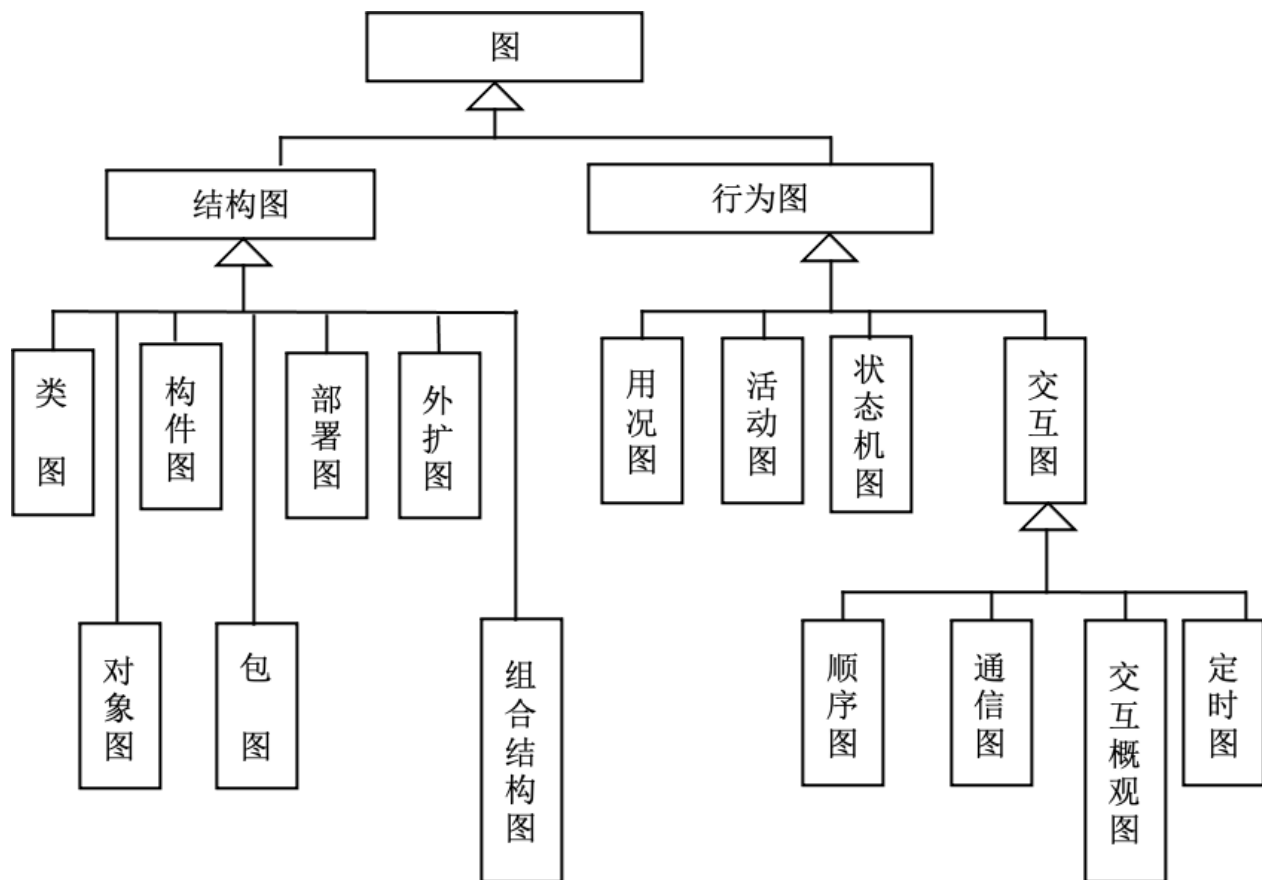


### 3 关于UML的图:表达格式-模型表达工具





### 3 关于UML的图:表达格式-模型表达工具

◆ UML为不同抽象层提供了7种可对**系统静态部分**建模的

图形工具:

- ① 类图; ② 对象图; ③ 构件图;
- ④ 包图; ⑤ 部署图; ⑥ 外扩图;
- ⑦ 组合结构图。

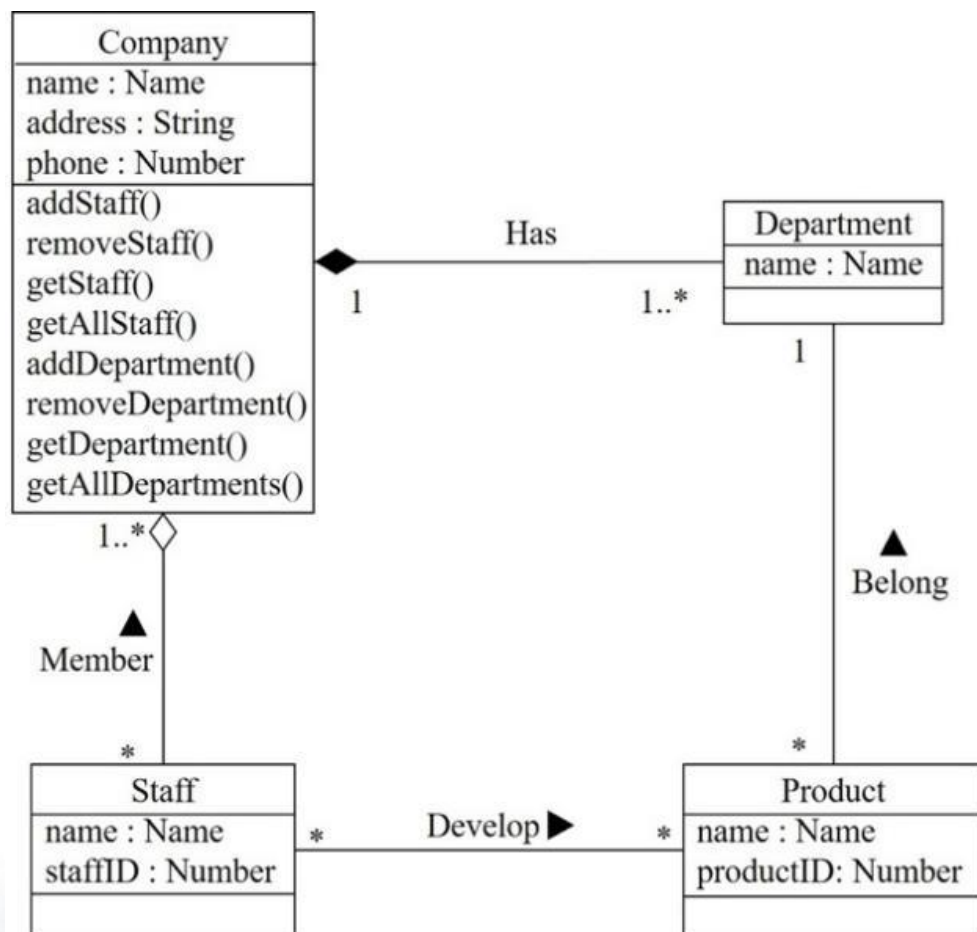
可将系统的静态方面看作是系统相对稳定的骨架的表示, 正如房屋的静态方面是由墙、门、窗、管子、电线等事物的布局组成一样



北京大学

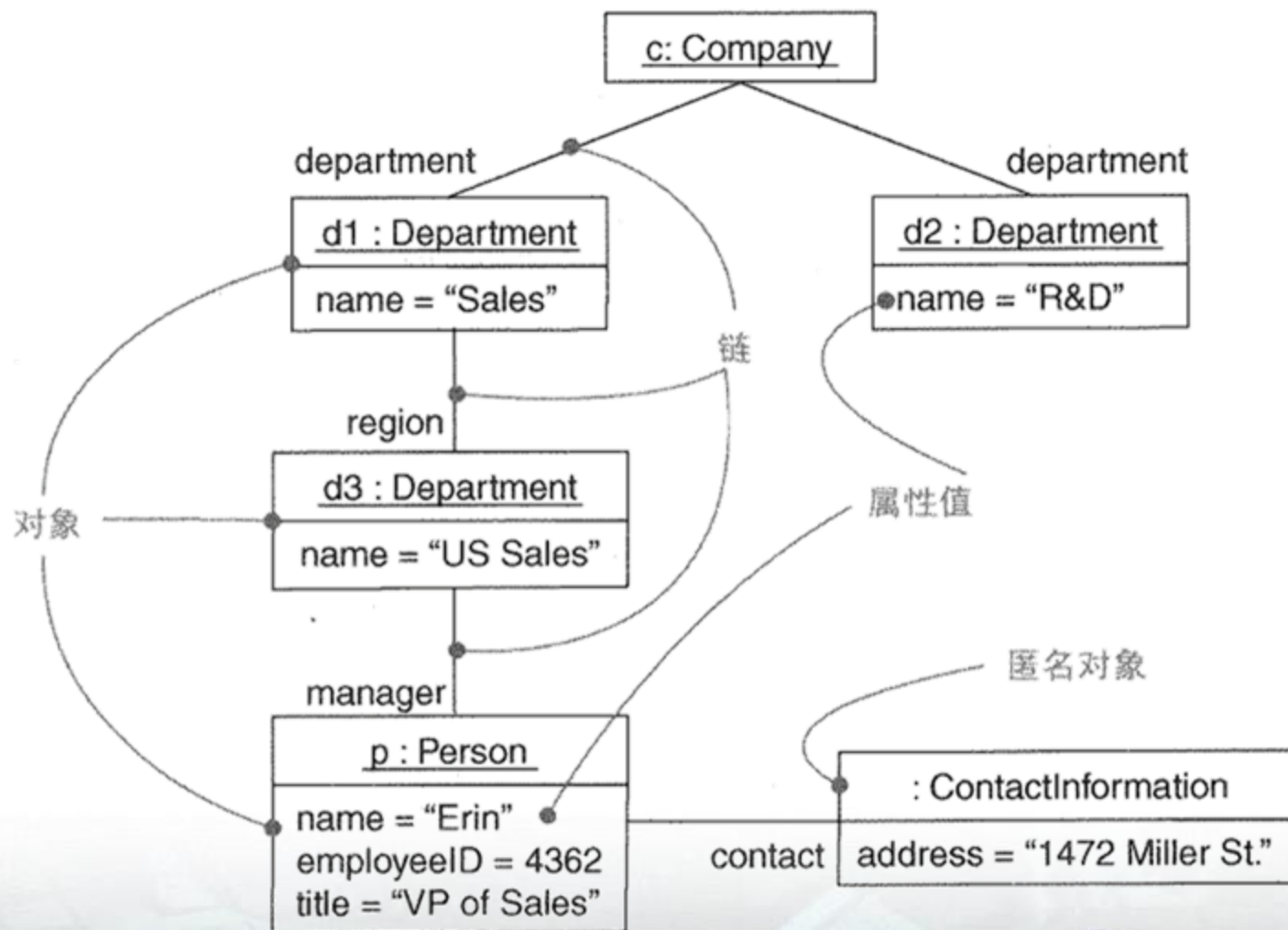


①类图：类图显示了类（及其接口）、类的内部结构以及与其他类的联系。  
是面向对象分析与设计所得到的最重要的模型。





②对象图：展示了一组对象以及它们之间的关系。用对象图说明在类图中所发现的事物的实例的数据结构和静态快照。



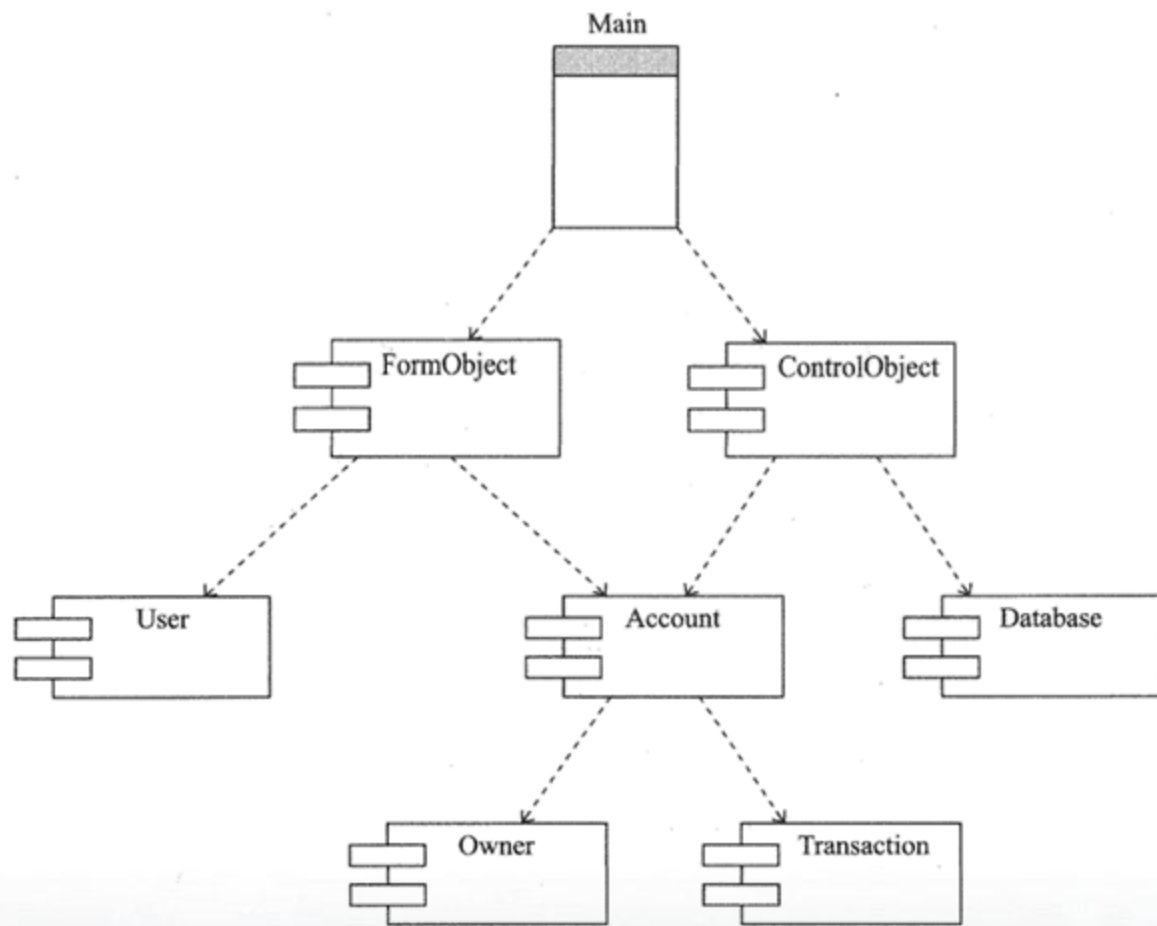
对象图示例



北京大学



③**构件图**：在转入实现阶段之前，可以用它表示如何组织构件。构件图描述了构件及构件之间的依赖关系。

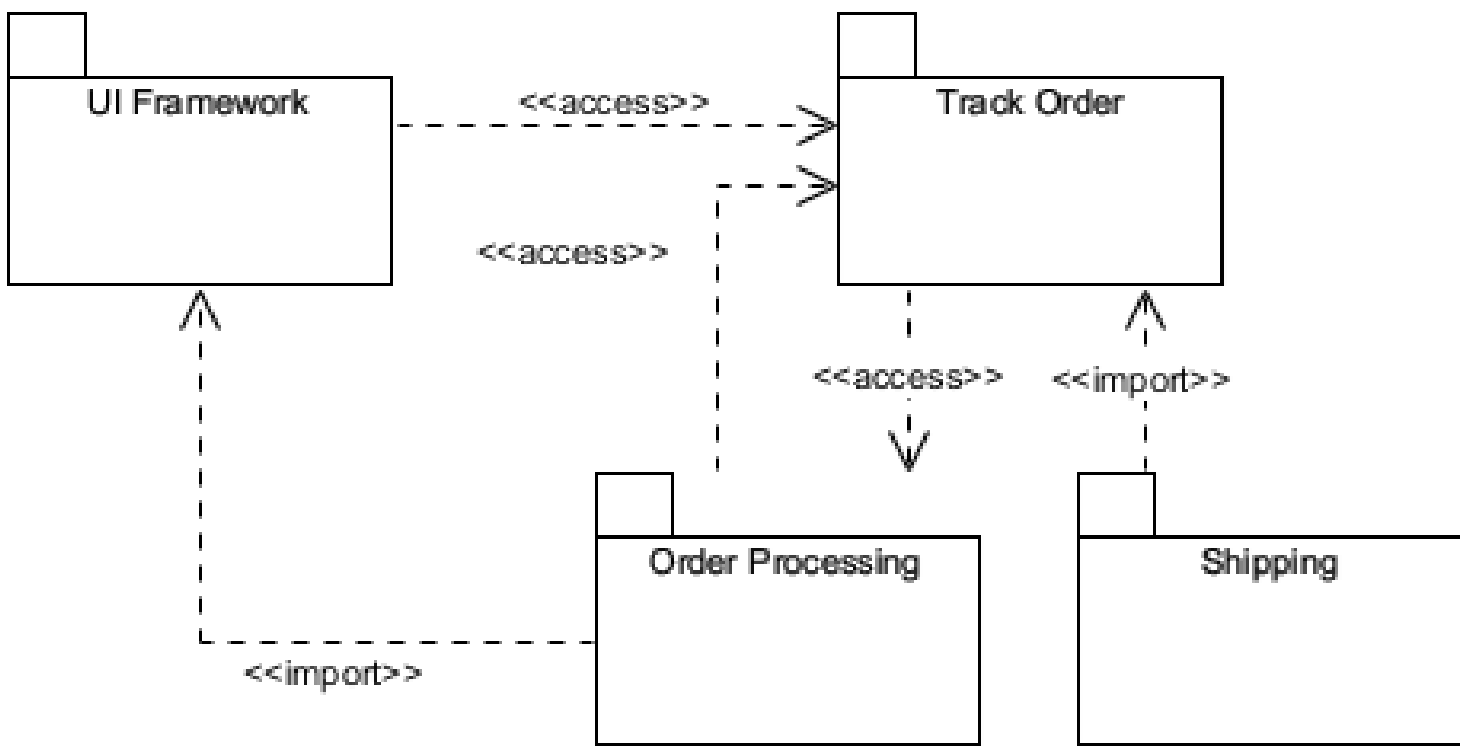


构件图示例：银行储蓄系统的构件图





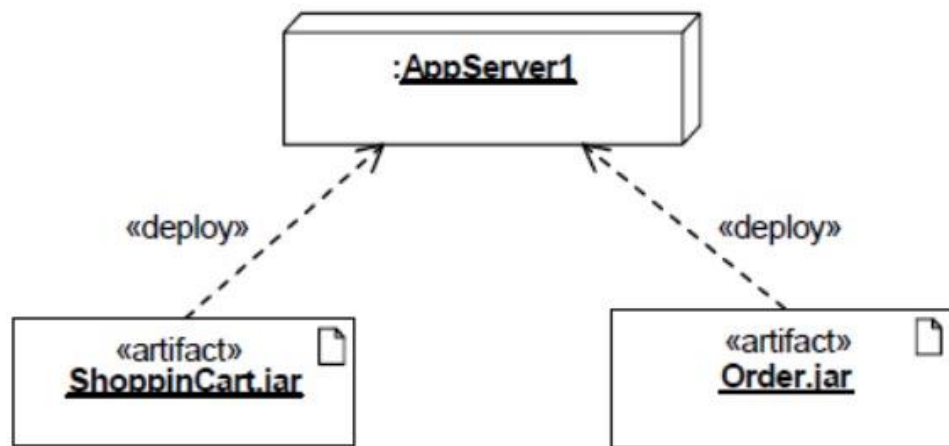
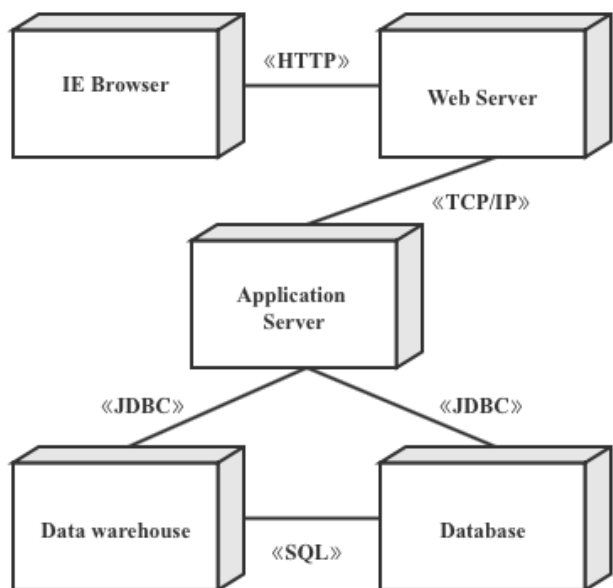
④包图：描述模型元素的分组（包），以及包之间的依赖关系的图。利用包图可以组织类目，控制信息组织和文档组织的复杂性。



包图示例



⑤部署图：部署图展示运行时进行处理的结点和在结点上生存的制品的配置。部署图用来对系统的静态部署视图建模。



部署图示例







# ⑥外扩图：外扩图是在UML2.5的基础上定义新建模元素的图，用以增加新的建模能力。

## UML Profile Diagrams

**Profile diagram** is **structure diagram** which describes **lightweight extension mechanism** to the UML by defining custom **stereotypes**, **tagged values**, and constraints. Profiles allow adaptation of the UML metamodel for different:

- *platforms*, such as Java Platform, Enterprise Edition (Java EE) or Microsoft .NET Framework, or
- *domains*, such business process modeling, service-oriented architecture, medical applications, etc.

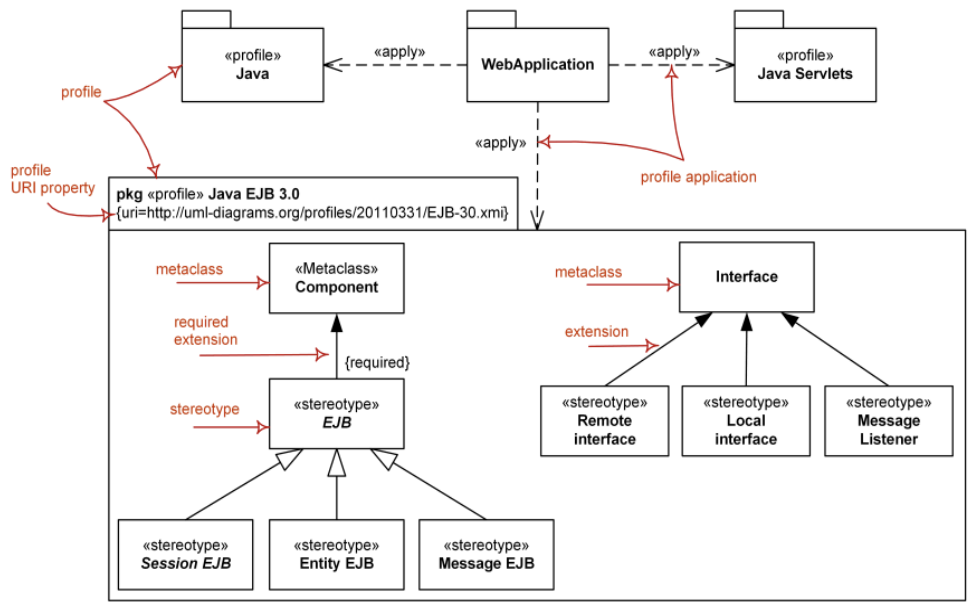
For example, semantics of standard UML metamodel elements could be specialized in a profile. In a model with the profile "Java model," generalization of classes should be able to be restricted to single inheritance without having to explicitly assign a stereotype «Java class» to each and every class instance.

The profiles mechanism is not a first-class extension mechanism. It does not allow to modify existing metamodels or to create a new metamodel as MOF does. Profile only allows adaptation or customization of an existing metamodel with constructs that are specific to a particular domain, platform, or method. It is not possible to take away any of the constraints that apply to a metamodel, but it is possible to **add new constraints** that are specific to the profile.

Metamodel customizations are defined in a profile, which is then applied to a package. **Stereotypes** are specific metaclasses, **tagged values** are standard metaattributes, and **profiles** are specific kinds of packages.

Profiles can be dynamically applied to or retracted from a model. They can also be dynamically combined so that several profiles will be applied at the same time on the same model.

Graphical nodes and edges used on profile diagrams are: **profile**, **metaclass**, **stereotype**, **extension**, **reference**, **profile application**.

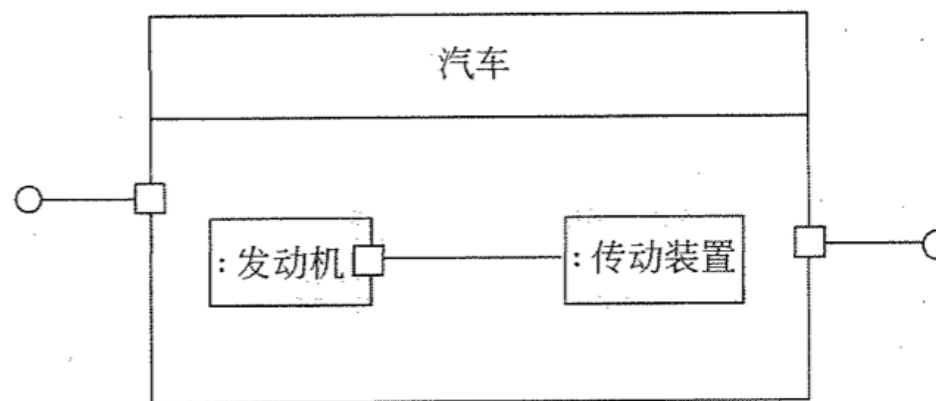


Major elements of UML profile diagram - **profile**, **stereotype**, **metaclass**, **extension**, **profile application**.

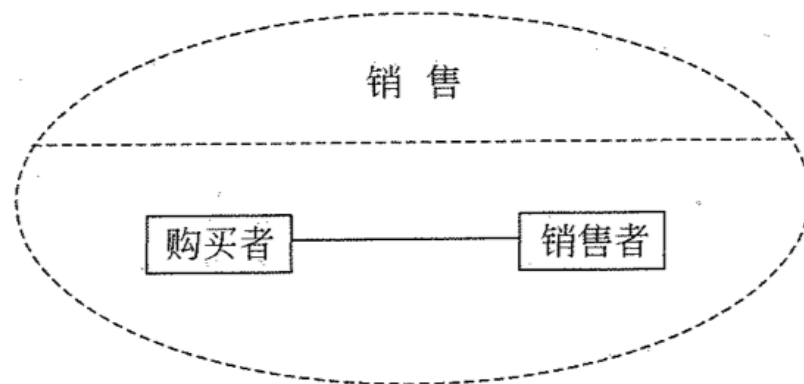




⑦组合结构图：是描述类或协作的内部结构的图。



(a) 一个类的组合结构图



(b) 一个协作的组合结构

组合结构图示例



北京大学



◆ UML为不同抽象层提供了7种可对**系统动态部分**建模的图形工具：

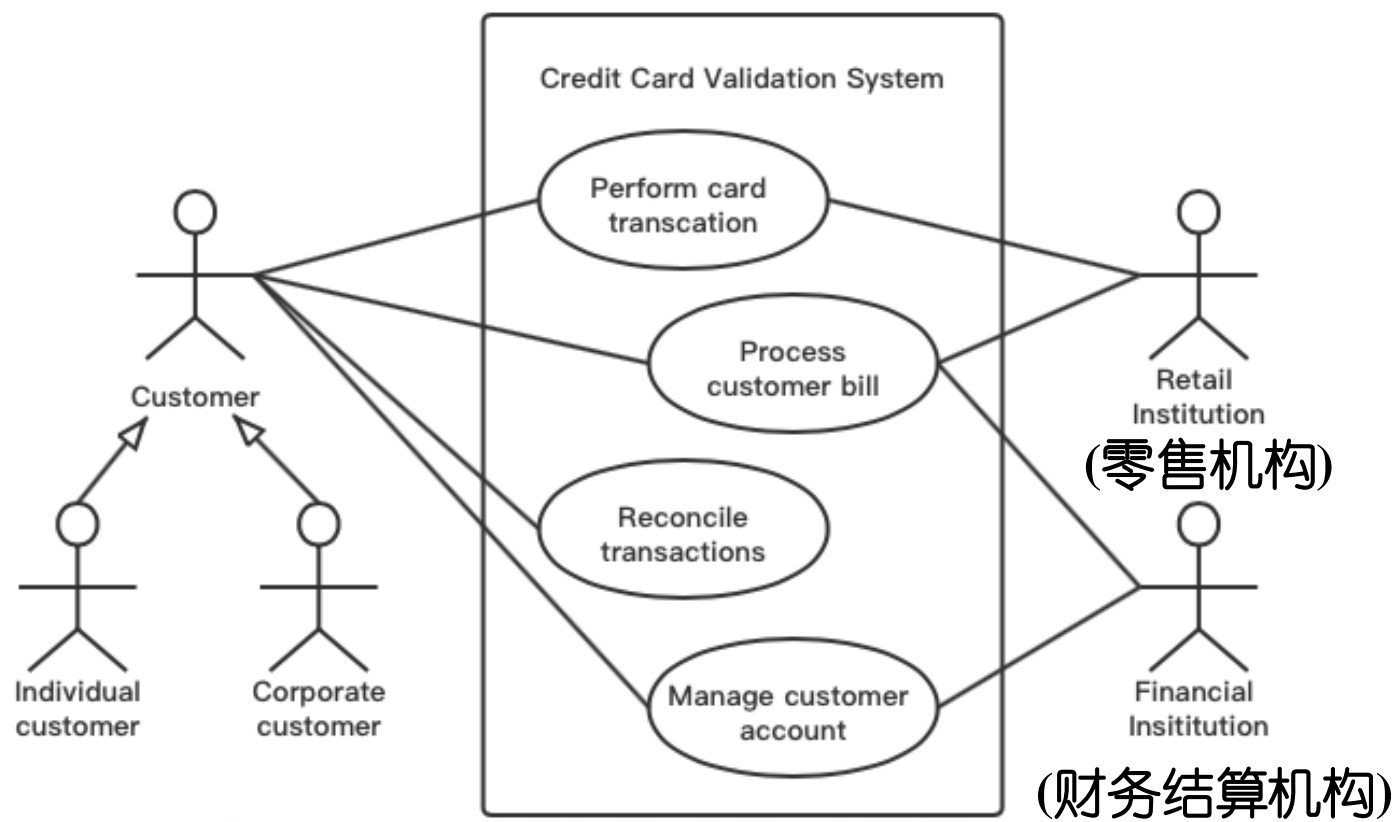
- ①用况图；      ②状态机图
- ③活动图；     ④顺序图
- ⑤通信图；     ⑥交互概观图
- ⑦定时图

可将系统的动态方面看作是对系统变化部分的表示。正如房屋的动态方面包含了气流和人在房间中的走动一样



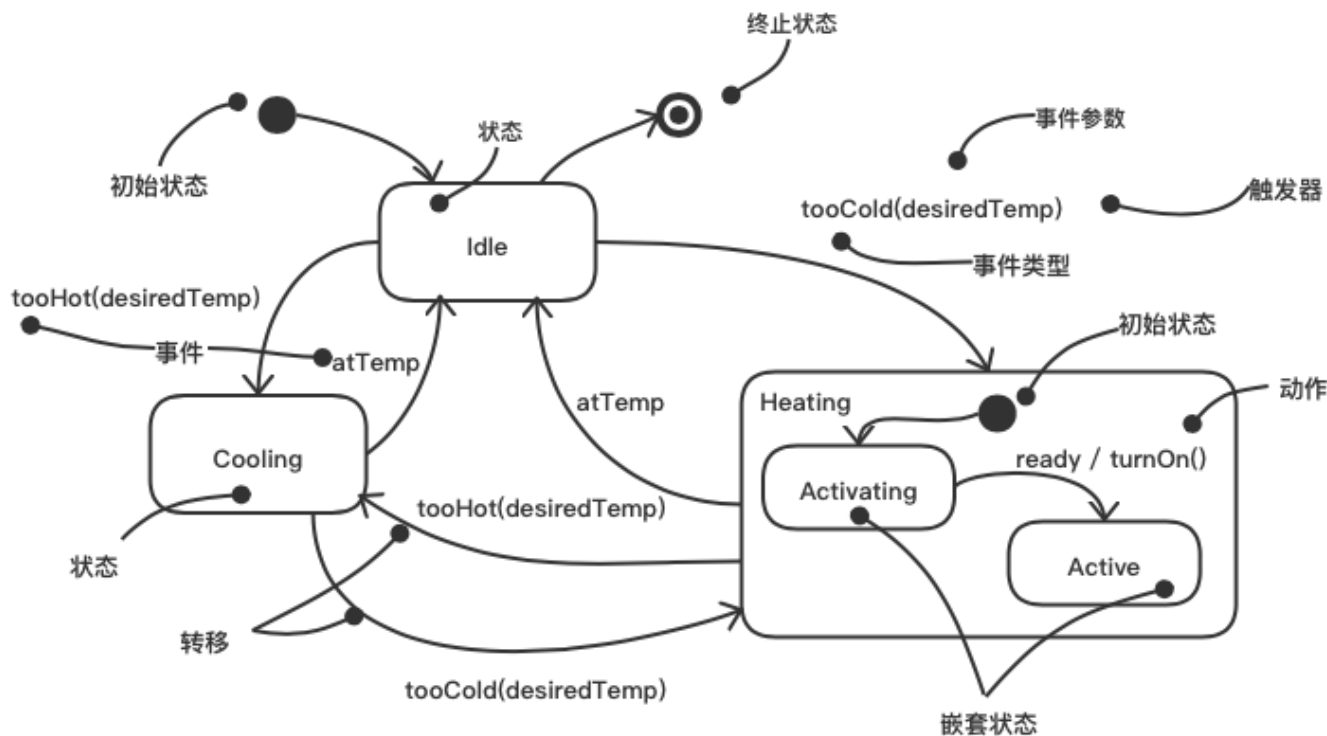


①用况图: 系统的需求模型, 用况图是表现一组用况、参与者以及它们之间关系的图。是描述需求模型的视图。



用况图示例

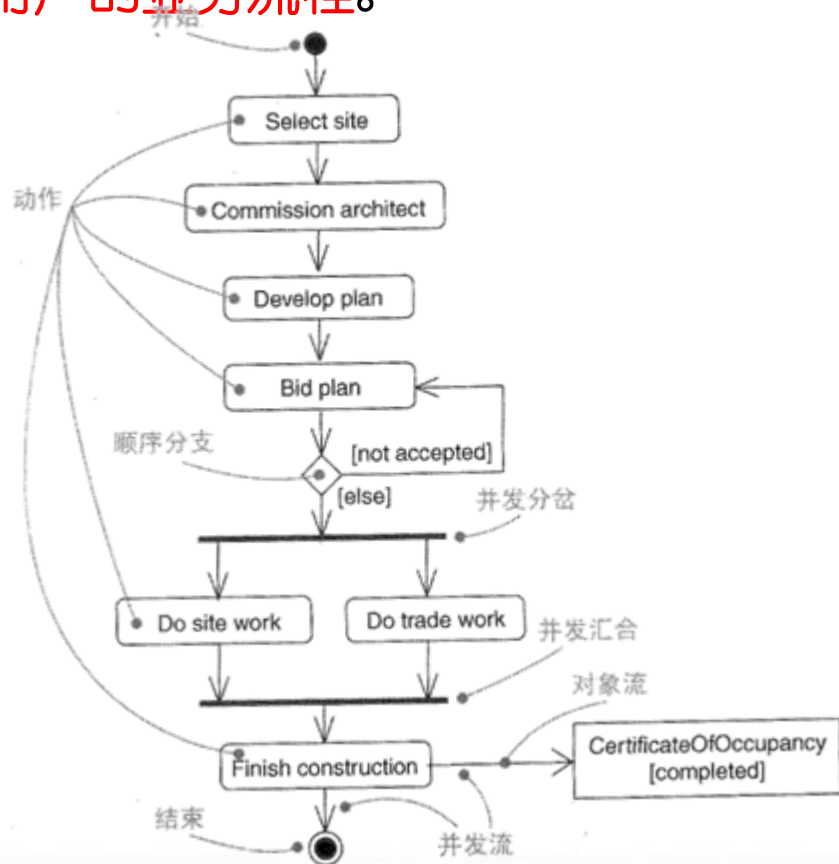
②**状态机图**：状态机图是描述一个对象或其他实体在其生命周期中所经历的各种状态以及状态变迁的图。当对象的行为比较复杂时，可用状态图作为辅助模型描述对象的状态及其状态转移，从而**更准确地定义对象的操作**。



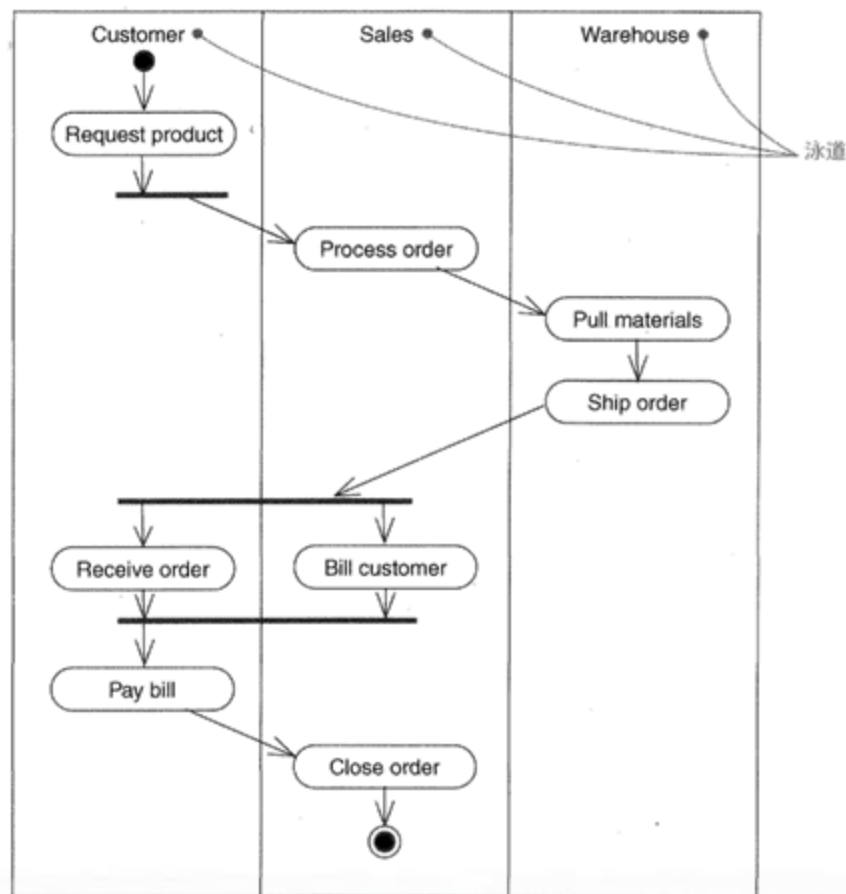
状态图示例



③活动图：描述活动、活动的执行顺序以及活动的输入和输出的图，  
可用来描述对象的操作流程，也可以描述一组对象之间的协作行为或  
用户的业务流程。



活动图示例1



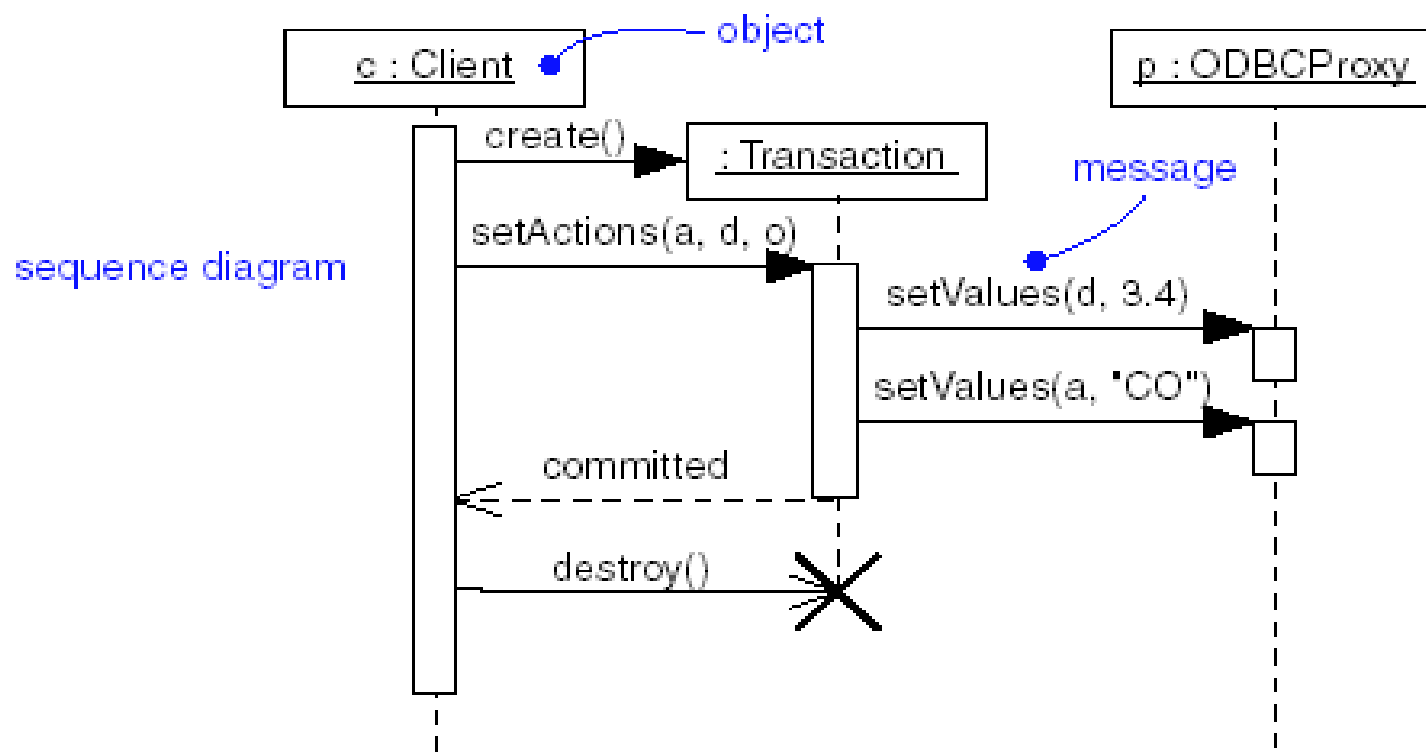
活动图示例2



北京大学

④顺序图：注重于消息的时间次序。可用来表示一组对象之间的交互情况

。



顺序图示例



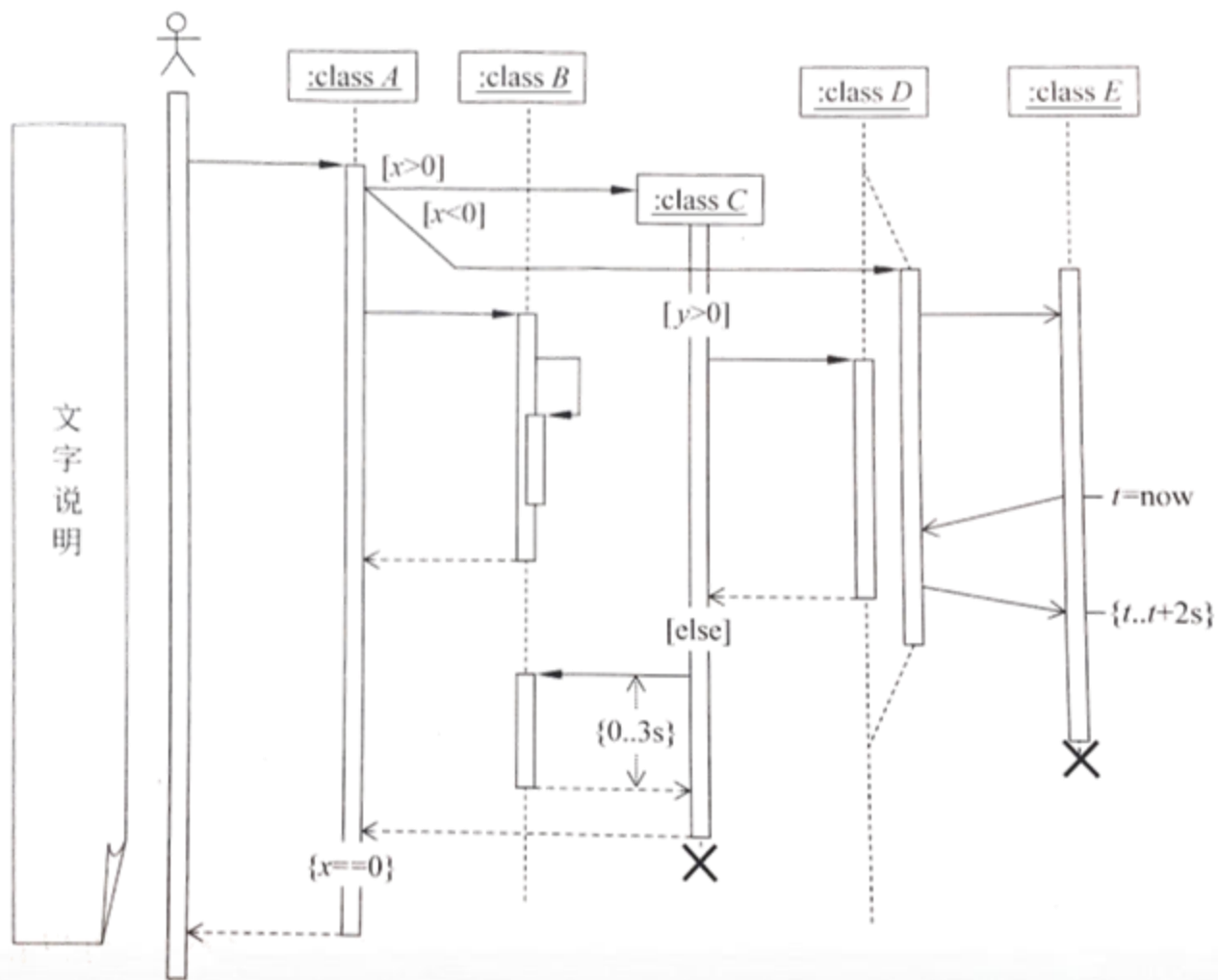




## 带参与者的顺序图画法说明：

早期的文献还支持在顺序图中表示系统边界和参与者。**Jacobson**方法在文字描述和图中的对象之间画一条系统边界，用内部涂为阴影的长条表示。这有利于看清系统内部的对象与外部的界限，而且便于表示系统与外部发生的交互。**Booch**方法也在顺序图的一侧画一个长条，并在它的上方画一个参与者符号。于是这个长条既表示了参与者的行为，也能显示系统内部的对象与外部事物之间的界限。**UML1**基本上采用了**Booch**的表示法，如下页图所示。**UML2**淡化了对这些内容的表示，在它给出的顺序图的例子中，系统边界和参与者都看不到了。原因是**UML2**把系统边界以外的参与者和系统内部的对象同等看待，都作为一条生命线，所以在表示法上就看不出它们之间有什么区别了。从实用的角度看，区分系统边界以外的参与者和系统内部的对象，对大部分软件系统开发是有意义的。因此我们建议：只要建模工具能够支持，则应尽可能采用顺序图的这种传统风格。

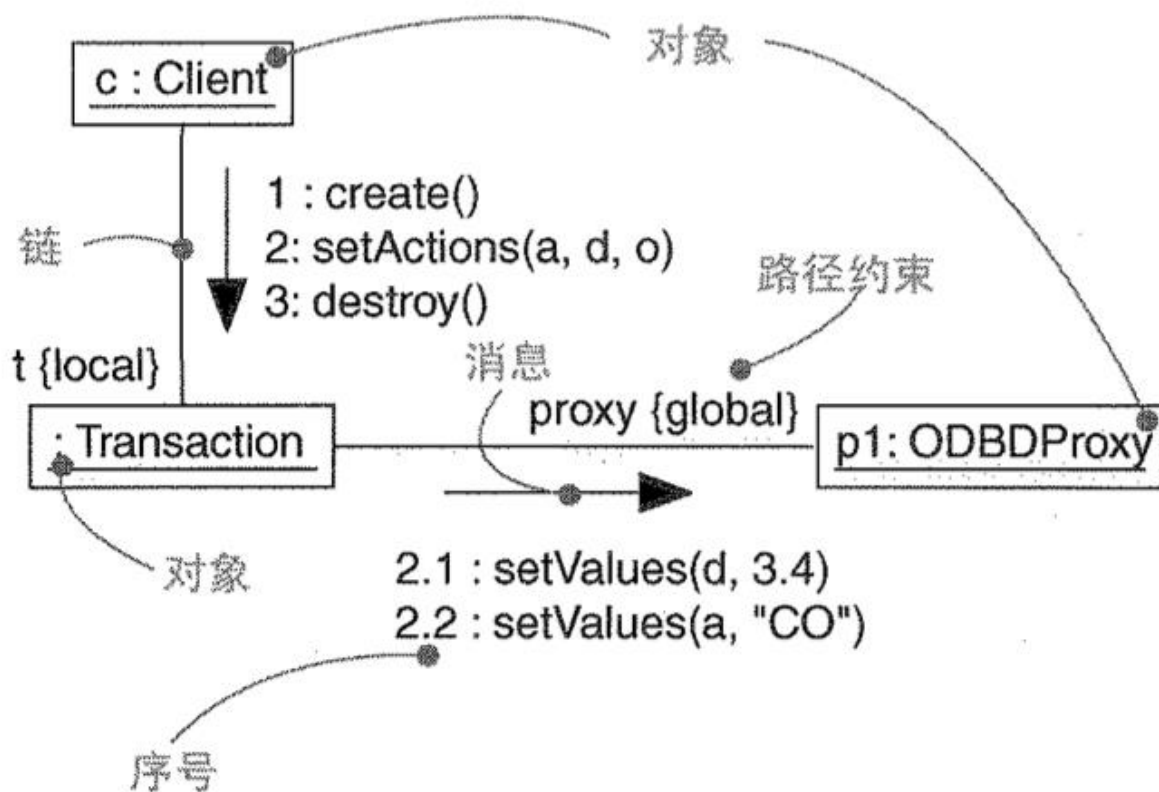




带参与者的顺序图例子



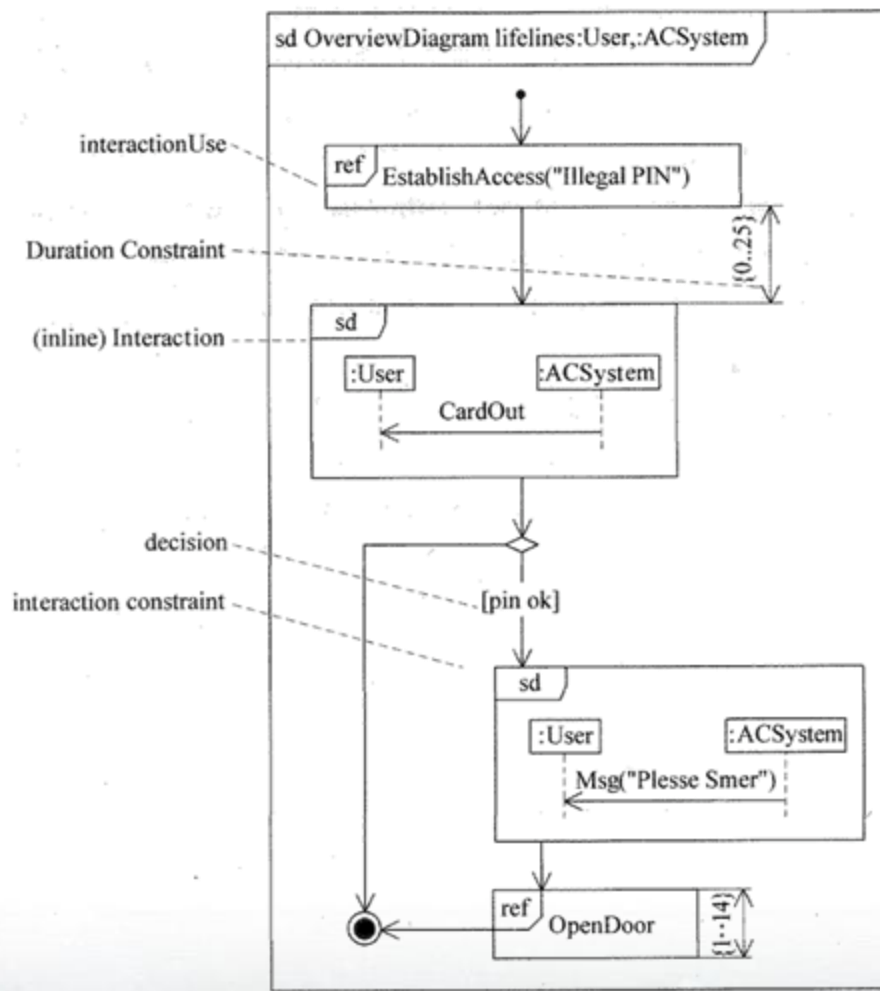
⑤通信图：注重于收发消息的对象的组织结构。可用来表示一组对象之间的交互情况。



通信图示例



⑥交互概观图：用于描述系统的宏观行为，是活动图和顺序图的混合物。

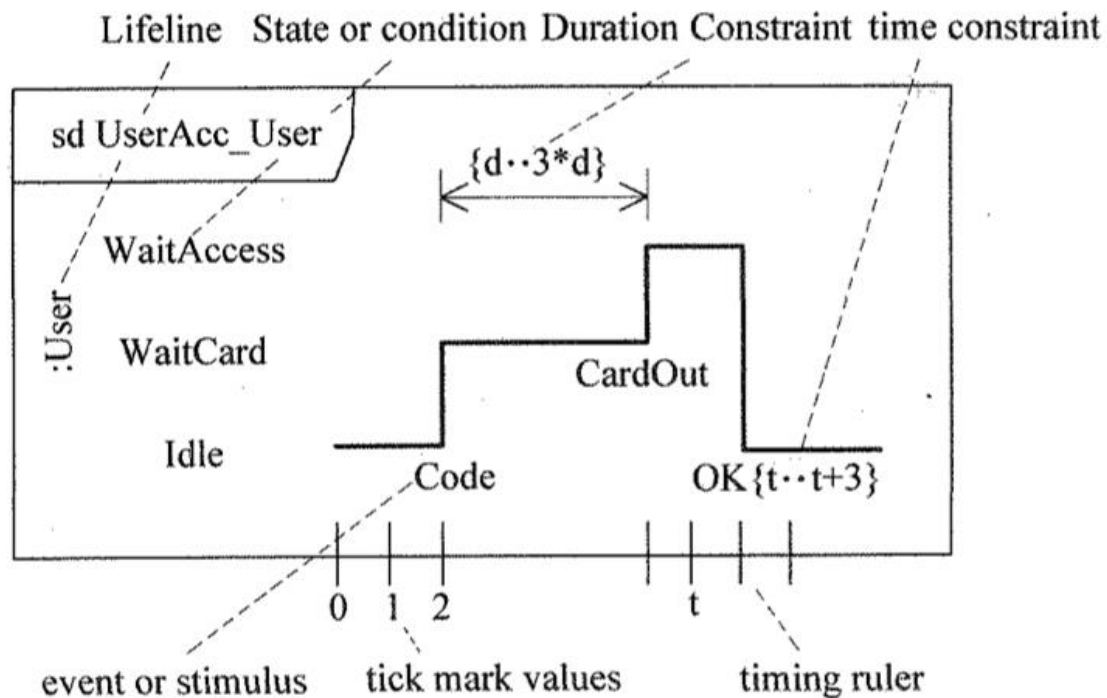


交互概览图示例





⑦**定时图**：用于表示交互，它展现了消息跨越不同对象或角色的实际时间，而不仅仅关心消息的相对顺序。



定时图示例







## 3.1 静态模型表达工具-类图

①定义：类图显示了类（及其接口）、类的内部结构以及与其他类的联系，是面向对象分析和设计所得到的最重要的模型。

作用：•可视化地表达系统的静态结构模型。

②类图的内容：

- 通常包含：•类；•接口；•依赖、泛化和关联关系等
- 还可以包含注解和约束，以及包或子系统，甚至，可包含一个实例，以便使其可视化。

注：这些成分，确定了所表达系统的各种形态。



北京大学





图1显示了公司有多个部门，每个部门有多个产品，公司有多位职员，例如：每位职员开发多个产品。

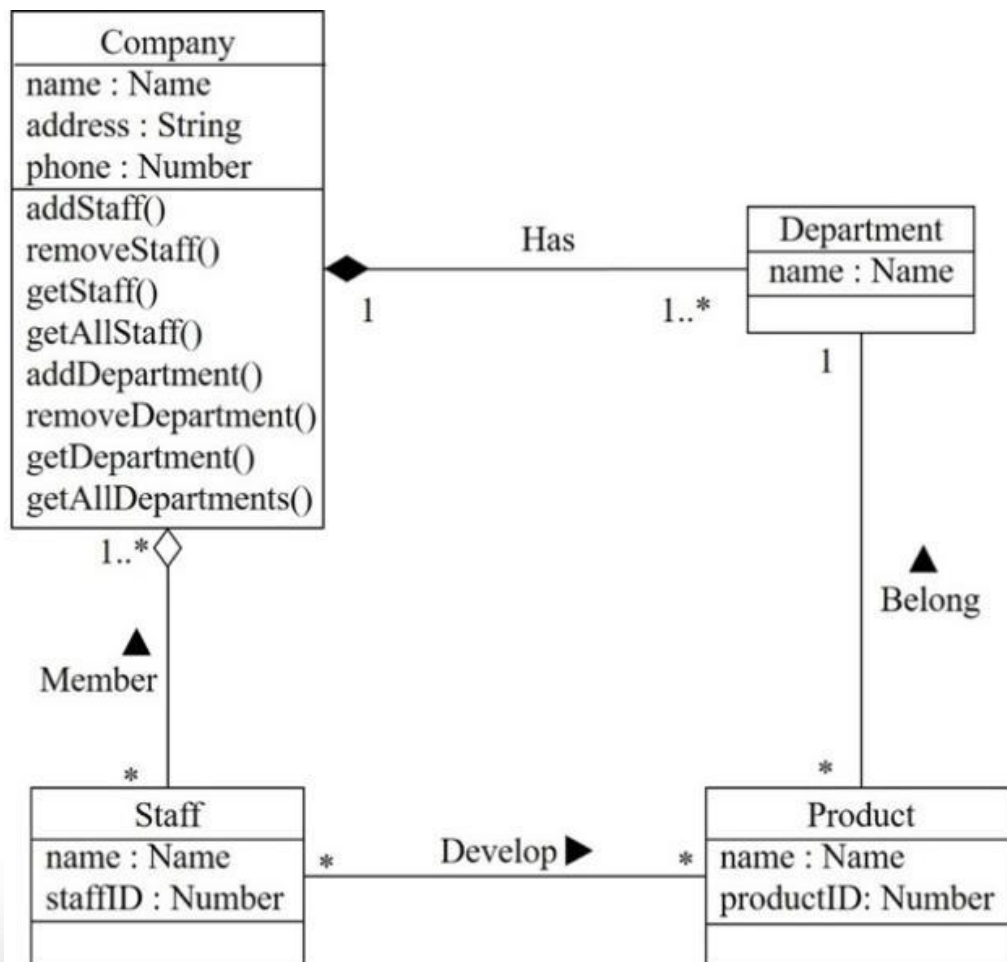


图1 类图示例





### ③类图的一般用法

类图主要用于对系统的静态视图进行建模（投影），支持表达系统的功能需求，即系统提供给最终用户的服务。

创建类图包括以下四方面工作：

①对系统中的概念（词汇）建模，形成类图中的基本元素

使用UML中的术语“类”，来抽象系统中各个组成部分，包括系统环境。然后，确定每一类的责任，最终形成类图中的模型元素。

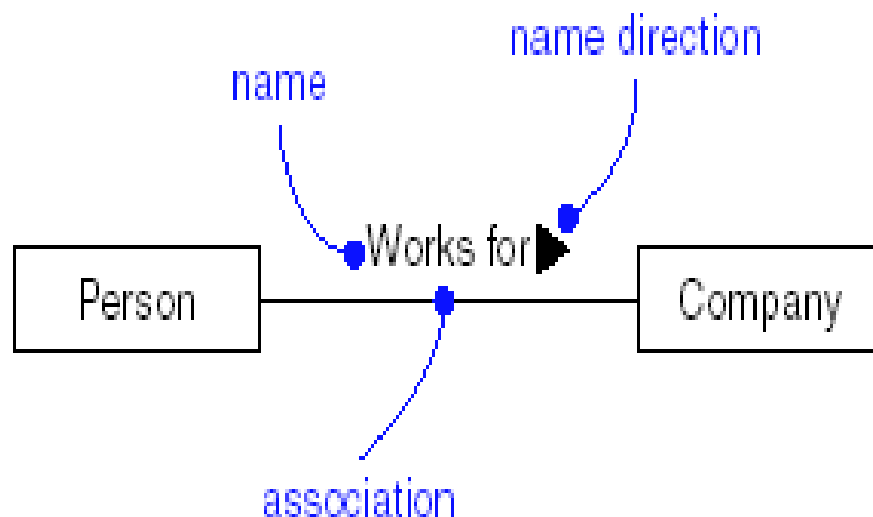


②对待建系统中的各种关系建模，形成该系统的初始类图

**使用UML中表达关系的术语，例如关联、泛化和依赖等来抽象系统中各成分之间的关系，形成该系统的初始类图。**

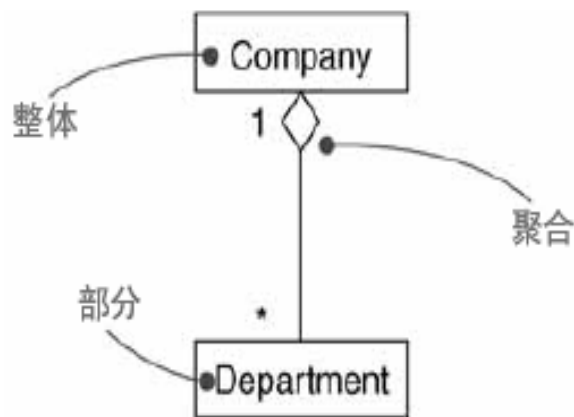
●当用关联关系建模时，是在对相互同等的两个类建模。给定两个类间的关联，则这两个类以某种方式相互依赖，并且常常从两边都可以导航。

A) 对于每一对类，如果需要一个类的对象到另一个类的对象导航，就要在这两个类之间建立一个关联；





- B) 对于每一对类，如果一个类的对象要与另一个类的对象相互交互，而后者不作为前者的过程局部变量或操作参数，就要在这两个类之间建立一个关联；
- C) 如果关联中的一个类与另一端的类相比，前者在结构或者组织上是一个整体，后者看来像它的部分，则在靠近整体的一端用一个菱形对关联修饰，从而将其标记为聚合。



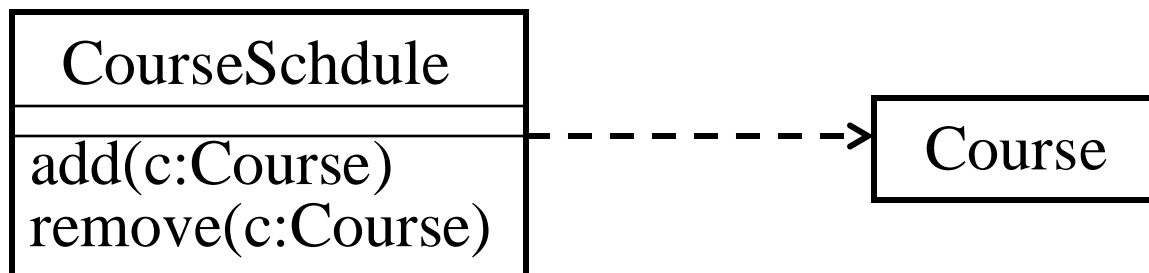
- D) 对于每一个关联，都要说明其多重性（特别当多重性不为\* 时，其中\*是默认的多重性）





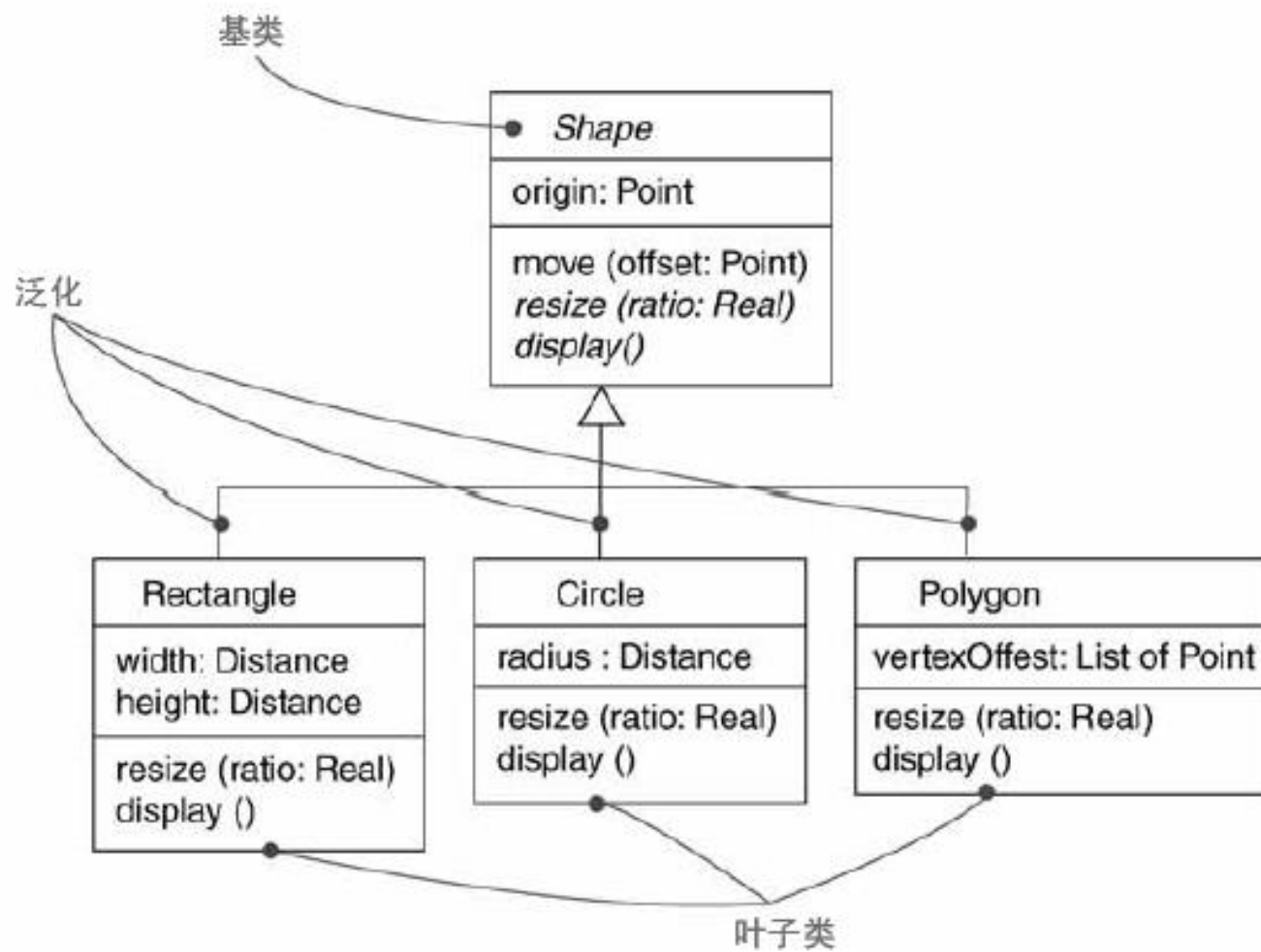
- 依赖关系是使用关系，常见的依赖关系是两个类之间的连接，其中一个类只是使用另一个类作为它的操作参数

➢ 创建一个依赖，从含有操作的类指向被该操作用作参数的类。



- 泛化关系是“is-a-kind-of”关系，在对系统的词汇建模中，经常遇到结构或行为上与其他类相似的类，可以提取所有共同的结构特征和行为特征，并把它们提升到较一般的类中，特殊类继承这些特征
- 给定一组类，寻找两个或以上的类的共同责任、属性和操作
- 把这些共同的责任、属性和操作提升为较一般的类
- 画出从每个特殊类到它的较一般的父类的泛化关系，用以表示较特殊的类继承较一般的类



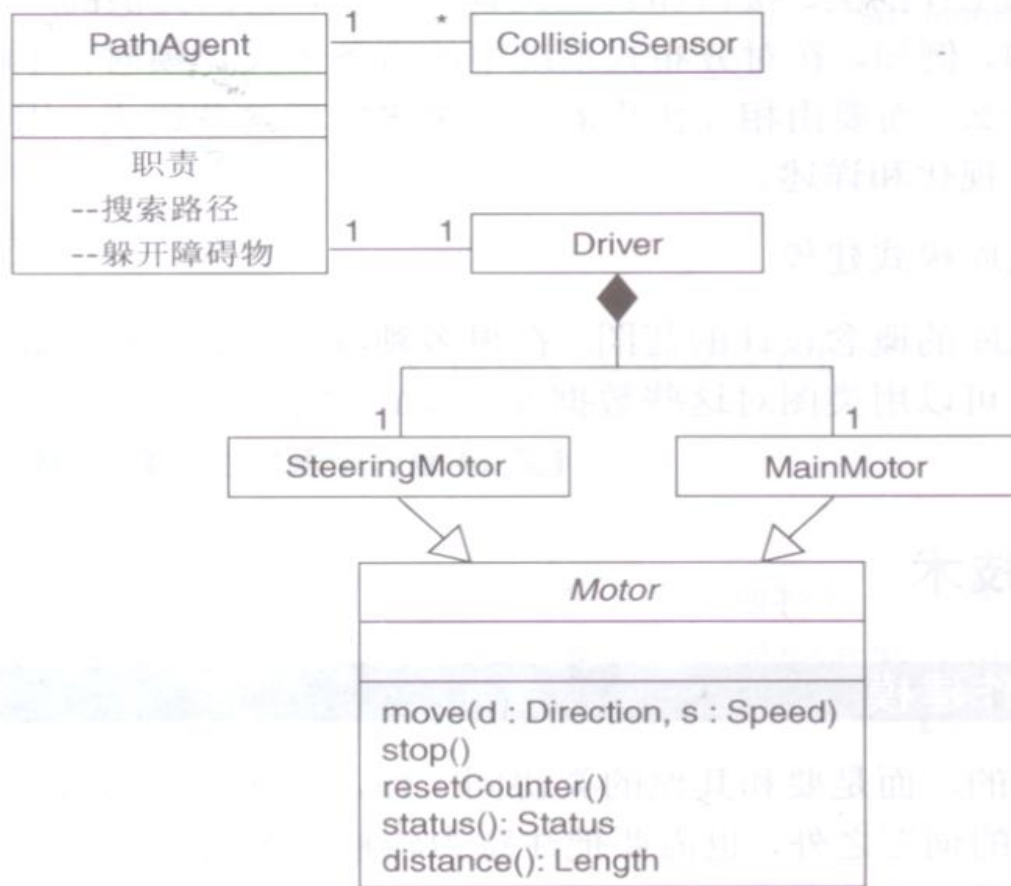




### ③ 模型化系统中的协作，给出该系统的最终类图

使用类和UML中表达关系的术语，模型化一些类之间的协作，用类图对这组类以及它们之间的关系建模。

例如下图显示了使机器人沿着一条路径移动所涉及的类



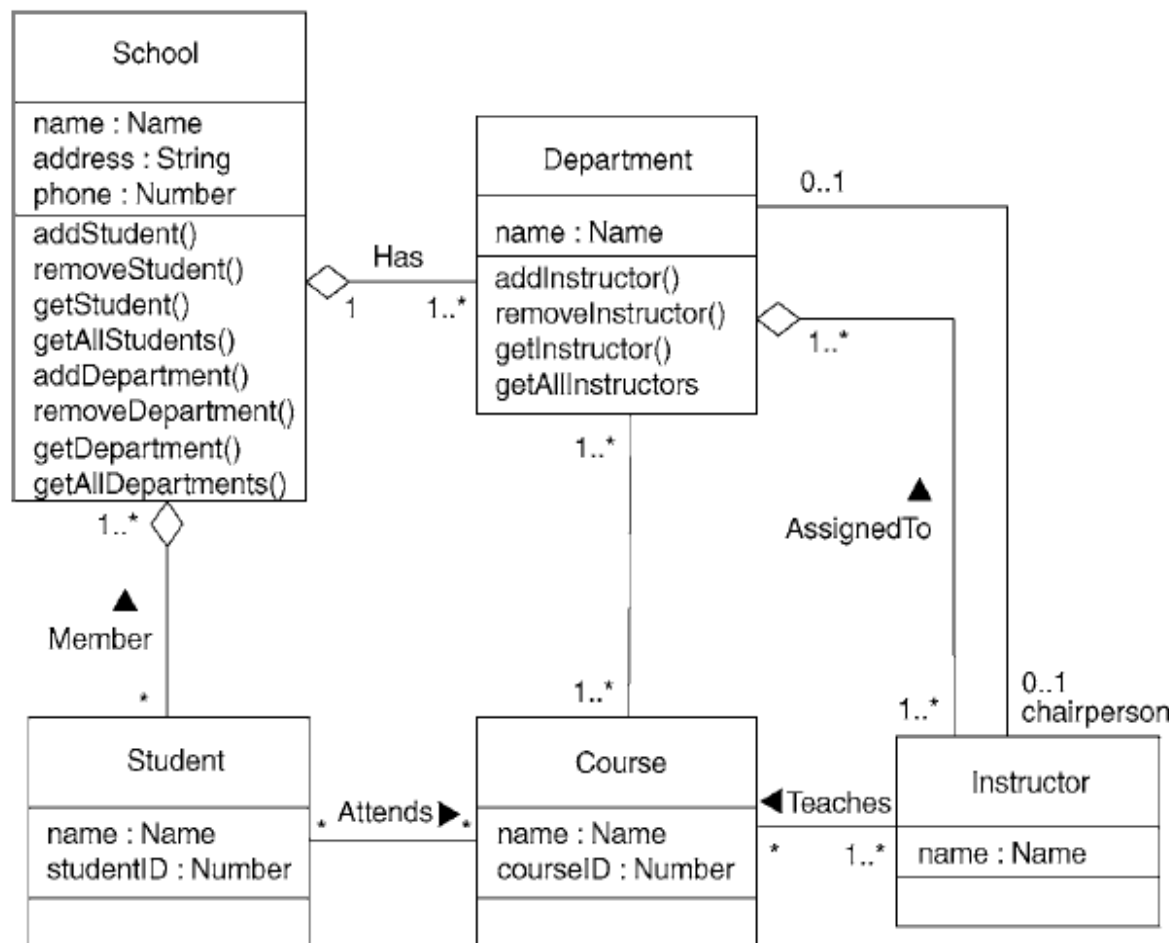
这就是一个协作



北京大学

## ④ 对逻辑数据库模式建模 (附加)

当需要给出数据库概念设计的指导,可对要在数据库中存储的信息,采用类图对相应的数据库模式进行中建模。



下图显示了某学校的信息系统的类图, 该图表明, 学校有多个系, 每个系有多名教员, 每位教员教授多门课程; 该学校还有学生, 每位学生要参加多门课。

本图显示的这些类的细节足以构造一个物理数据库。  
School 和 Department 显示了几个操纵其部件的操作。这些操作对维护数据的完整性很重要。

