

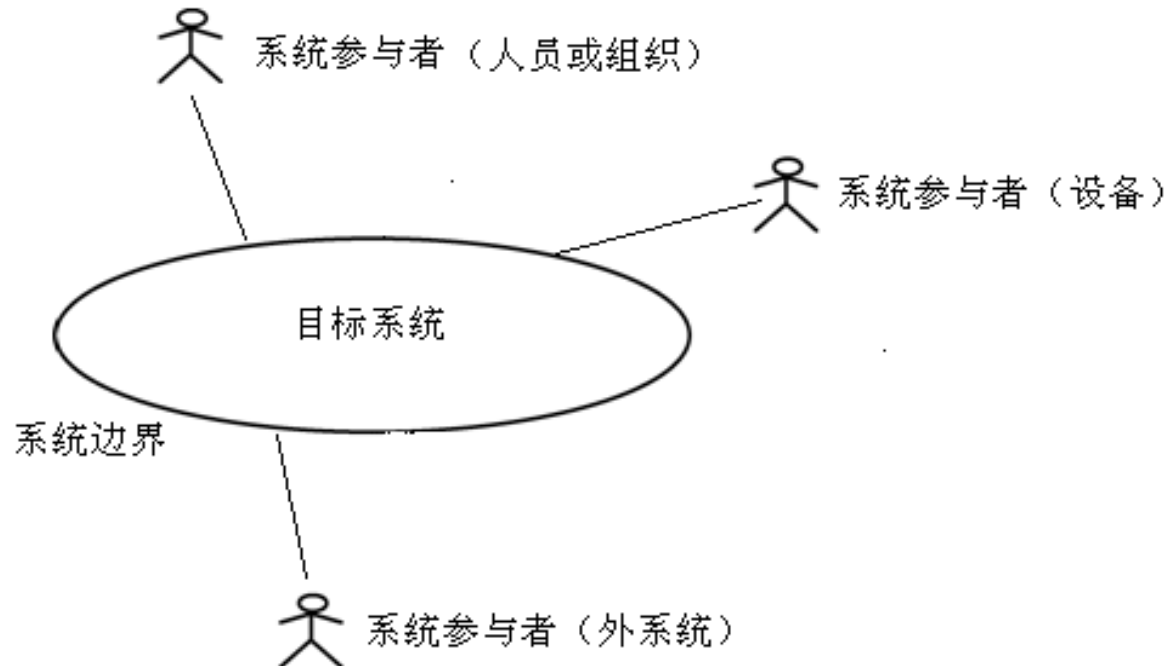
# 第6章 面向对象分析

- 面向对象分析概述
- 建立用例模型
- 建立对象模型
- 建立动态模型

# 6.1 面向对象分析概述

## • 确定系统边界

- 系统边界是系统的所有内部成分与系统以外各种事物的分界线。系统只通过边界上有限数量的接口与外部的系统参与者（人员、组织、设备或外系统）进行交互。



# 6.1 面向对象分析概述

- 面向对象分析的3种模型
  - 用例模型：用例和场景表示的功能模型；
  - 对象模型：用类和对象表示的静态模型；
  - 交互模型：由状态图和顺序图表示的动态模型。

## 6.2 建立用例模型

- 建立用例模型的过程

- (1) 确定业务参与者——标识目标系统将支持的不同类型的用户，可以是人、事件或其他系统。
- (2) 确定业务需求用例——参与者需要系统提供的完整功能。
- (3) 创建用例图——标识参与者与用例之间、用例与用例之间的关系。

**例: 选课系统。** 给教师分配课程和学生注册课程。

- 在每个学期选课开始之前，**系统管理员**需要对系统中的教师信息、课程信息和学生信息进行维护。学期结束后，将本学期成绩归档到**学籍档案系统**。
- **学生**登录系统后会得到一份本学期将要开设的课程目录。每门课程包含的信息有开课系别、教师、上课时间、教室、容纳的学生数量和选择课程的先决条件。
- 当学生选择了一门课程后，系统需访问学籍档案系统，查询是否符合选课的先决条件。如果不符合，系统给出提示信息。
- 每个学期有一段时间让学生可以改变计划，学生可以在这段时间内访问联机系统以增选课程或退选课程。

**例: 选课系统。** 给教师分配课程和学生注册课程。

■ **教师**可以访问在线系统，查看将要教授哪些课程和每门课程有哪些学生报名，课程考试结束后可以提交成绩，系统可以生成带有成绩分布统计结果的成绩单。

# 1. 确定业务参与者

- 通过关注系统的业务参与者，我们可以将重点放在如何使用系统，而不是如何构造系统上，并且有助于进一步明确系统的范围和边界。
- 当系统比较庞大和复杂时，要搞清楚系统的需求往往比较困难，通过明确参与者，可以针对参与者确定系统需求，有助于保证系统需求的完整性。

# 1. 确定业务参与者

- 可通过以下资料来确定系统的参与者：
  - 标识系统范围和边界的环境图；
  - 现有系统（如果有的话）的文档和用户手册；
  - 项目会议和研讨会的记录；
  - 现有的需求文档、工作手册等。



# 1. 确定业务参与者

- 还可以通过提出以下问题，明确系统的参与者：
  - 谁或者什么为系统提供输入？
  - 谁或者什么接收系统的输出？
  - 需要与其他系统连接的接口吗？
  - 是否存在在预定的时间自动触发的事件？
  - 谁将维护系统中的信息？

# 1. 确定业务参与者

- 从选课系统的需求描述中，可以确定4类参与者：
  - 学生 (Student)
  - 教师 (Teacher)
  - 系统管理员 (Administrator)
  - 学籍档案系统 (Archive System)

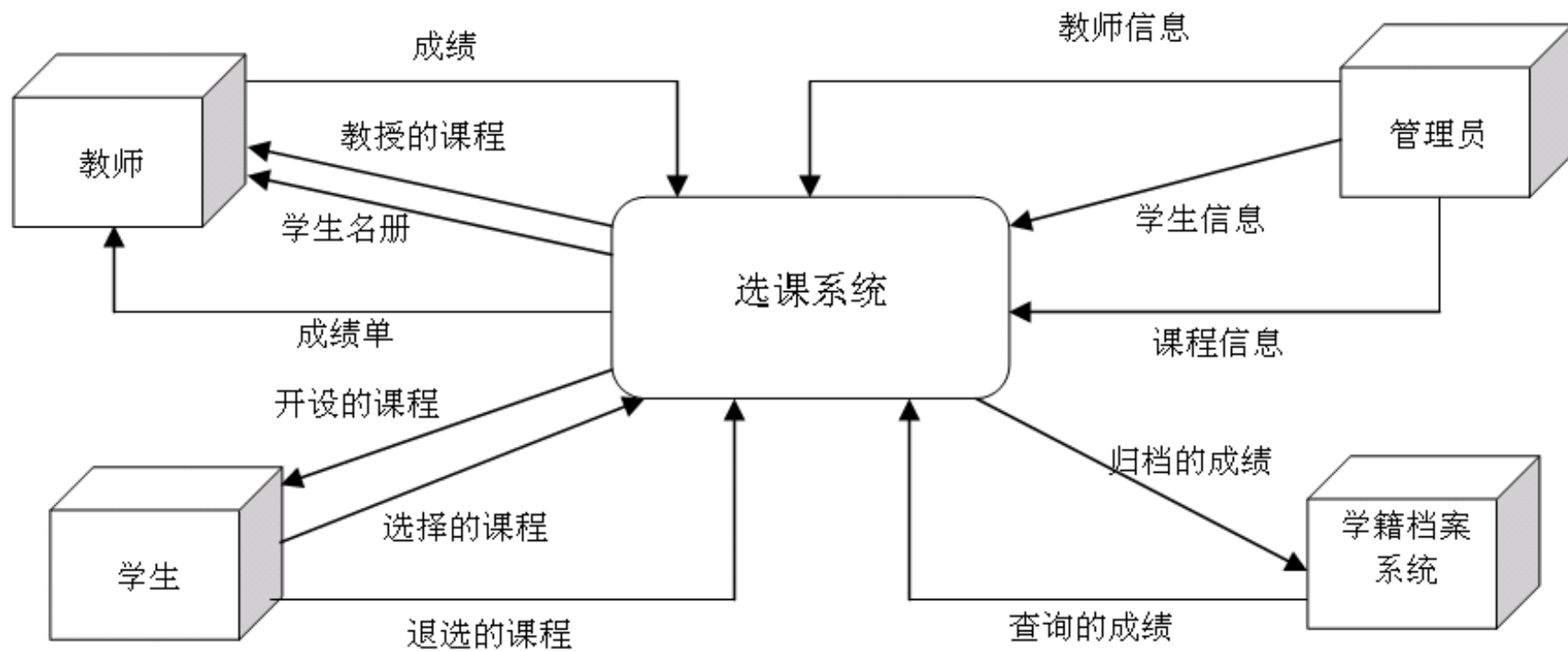
## 2. 确定业务需求用例

- 以下问题可以帮助我们更好地标识系统的用例：
  - 每个参与者的特定任务是什么？
  - 是否每个参与者都要从系统中创建、存储、改变、移动或读取信息？
  - 是否任何参与者需要通知系统有关突发性的、外部的改变？
  - 哪些用例支持或维护系统？
  - 目前的用例是否覆盖了所有功能需求？

## 2. 确定业务需求用例

- 环境图是分析参与者和发现潜在用例的极好来源，它不仅可以用在结构化分析方法中，也可以用于面向对象的分析方法中。
- 通过环境图，可以确定系统的主要输入输出，通过提交和接收输入输出的各方确定潜在的用例。

## • 选课系统的环境图



## • 选课系统的业务列表

参与者	用例	说明
学生 (Student)	选择课程 (Register for a course)	选择一门课程注册。
	退选课程 (Withdraw from a course)	从已经注册的课程中取消注册。
	取得课程目录 (Get a course catalogue)	得到本学期开设课程的课程目录。
教师 (Teacher)	查看教授的课程 (Get teaching courses)	查看本学期教授的课程信息。
	得到学生名册 (Get student list of a course)	得到教授的课程下所注册的学生名单。
	提交成绩 (Submit score)	输入、修改学生的成绩，并提交。
	得到成绩单 (Get score report)	系统生成带有成绩统计信息的成绩单。
系统管理员 (Administrator)	维护课程信息 (Maintain course information)	修改课程信息。
	维护教师信息 (Maintain teacher information)	修改教师信息。
	维护学生信息 (Maintain student information)	修改学生信息。
	成绩归档 (Pigeonhole)	将学生本学期的成绩提交给学籍档案系统。
学籍档案系统 (Archive System)	查找成绩信息 (Search score information)	从档案系统中查找学生某门课的成绩。
	接收归档的成绩 (Accept pigeonhole)	存储来自选课系统的归档成绩。

- 用例的规格说明

- 对用例的完整描述包括用例名称、执行者、前置条件、后置条件、一个主事件流、零到多个备选事件流。
- **主事件流**表示正常情况下执行者与系统之间的信息交互及动作序列，**备选事件流**则表示特殊情况或异常情况下的信息交互及动作序列。
- 应给出每个用例的规格说明。

## 用例“选择课程”的规格说明

用例名称: Register for a course (选择课程)	执行者: 学生
1.1 前置条件: 供选择的课程信息存在于系统数据库中。	
1.2 后置条件: 如果此用例执行成功, 则此学生已选课程列表下增加了一门课程, 同时, 选择此课程的学生列表中增加了一名学生。如果执行不成功, 系统状态不变。	
1.3 <u>主事件流</u>	
1) 当学生在菜单栏中选择“选择课程”菜单时, 此用例开始。	
2) 学生在本学期开设课程的列表中选择一门课程;	
3) 系统给出提示, 说明此课程的先修课都有哪些 (仅供学生参考);	
4) 单击“选择”按钮;	
5) 系统检查已选修此课程的学生数量及此学生已修课程是否符合此课程的先决条件;	
6) 系统更新此课程的选修人数、此学生的选课信息及此课程下的选课学生信息;	
7) 系统提示选择成功。	



## • 用例“选择课程”的规格说明

### 1.4 备选事件流

E-1: 若此课程的选修人数已达到最到容量，系统给出提示信息“此课程的选课人数已满，请选择其他课程!”，此用例结束。

E-2: 若学生已修课程不符合此课程的先决条件，系统给出提示信息“你目前还不适合选择此课程，请通过其先修课程后再选择此课程!”，此用例结束。

E-3: 若系统不能成功更新数据库，则提示“更新选课信息失败，请稍候再试!”，此用例结束。

### 3. 创建用例图

- 用例图是若干个参与者和用例，以及它们间的关系构成的图形表示。
- 每个系统通常都有一个总体视图（Global View of Actors and Use Cases），如果总体视图过于复杂，则可以创建多个用例图，每个用例图关注系统的某一方面。
- 通常是围绕参与者创建用例图。

# 使用Rose创建用例模型

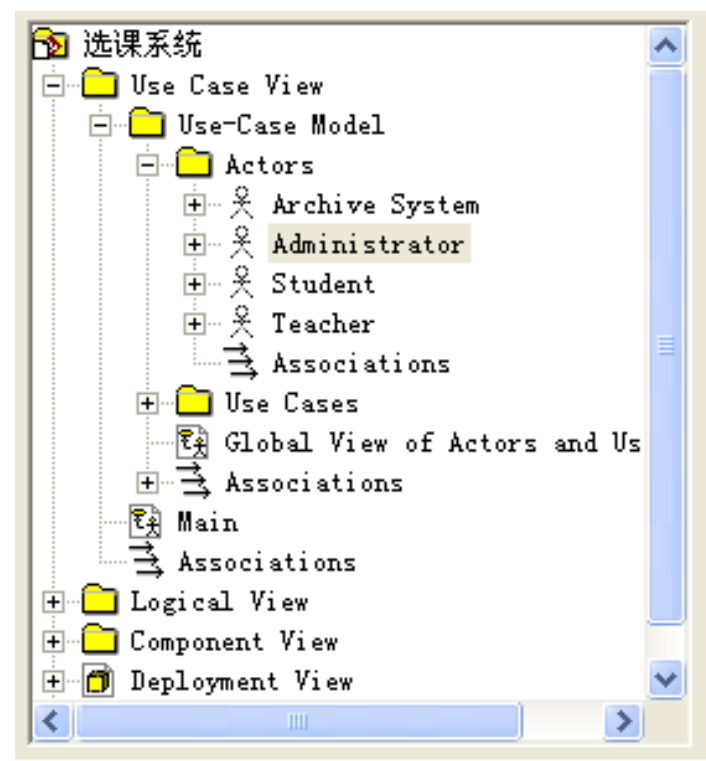
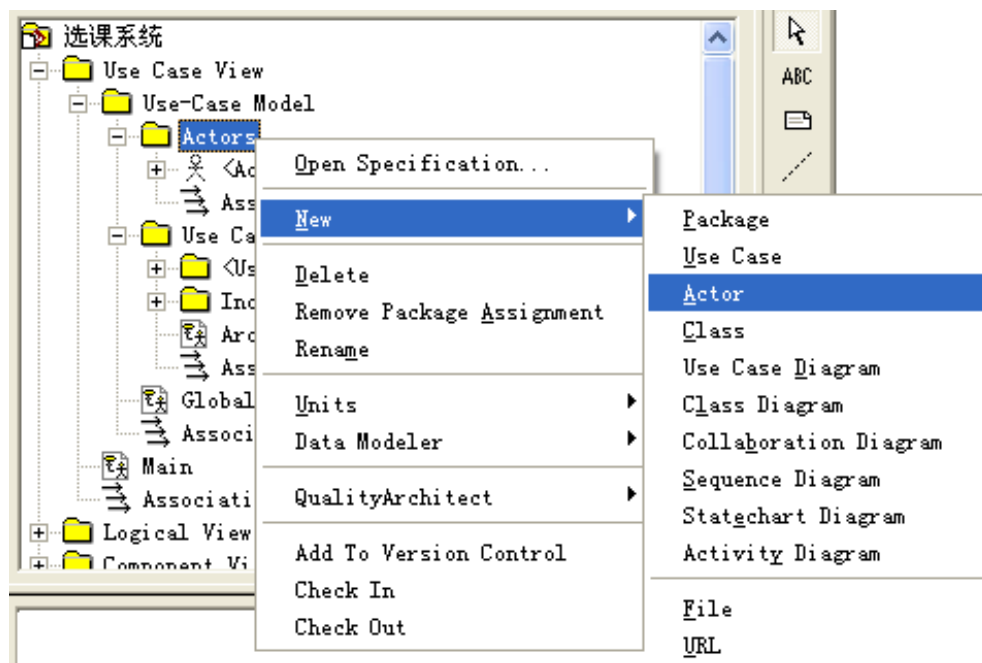
## 1. 在Rose中创建参与者（actors）

创建的角色应放在Use-Case Model包中的Actors包中比较好，如果模型结构中没有这样的包，可以先创建Actors包，之后按下面的步骤创建角色：

- (1) 在浏览器窗口中的Actors包上单击鼠标右键，显示弹出菜单；
- (2) 选择New: Actor菜单选项，系统创建名为New Class的参与者；
- (3) 选中新创建的参与者，更名为设计的名字。

# 使用Rose创建用例模型

## 1. 在Rose中创建参与者（actors）



# 使用Rose创建用例模型

## 2. 给参与者添加文档

- 在Rose中我们对模型元素都可以（在多数情况下是必须的）添加文档以描述更多的信息。
- 应该给模型中的每个actor添加摘要描述，摘要描述表示actor和系统交互的规则。
- 课程注册系统中actor的摘要描述如下：

Student——在学校注册上课的人；

Teacher——学校授权上课的人；

Administrator——系统认同的维护人员；

Archive System——管理学生学籍档案的外部系统。

# 使用Rose创建用例模型

## 3. 在Rose中创建用例（Use Case）

- 应该将用例放在Use cases包中，如果您的模型结构中没有这样的包，则应该先创建包。
- 也可以将每个用例放在单独的包中，因此可以先创建相应的包，再创建用例。
- 另外，在多个用例中使用的公用用例，如验证用户（Validate User），可以放在一个单独的包中，如可以在Use cases包中创建Included Use Cases包来存放公用用例。

# 使用Rose创建用例模型

## 3. 在Rose中创建用例（Use Case）

在Rose中创建用例的过程如下：

(1) 在浏览器的相应包上单击鼠标右键，弹出菜单显示；

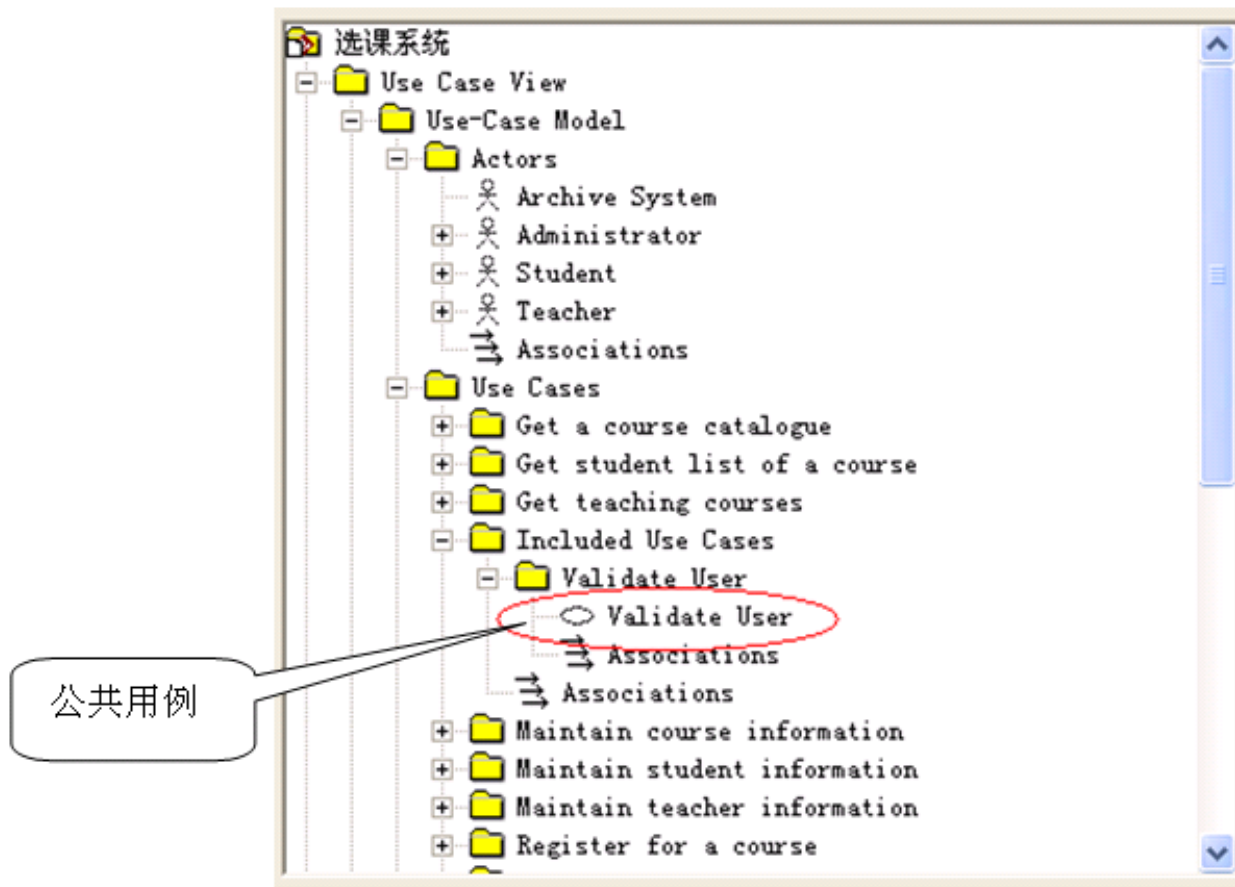
(2) 选择New: Use Case选项，则在浏览器中生成名为NewUseCase的新用例；

(3) 选中创建的新用例，键入设计的名字。

重复上面的步骤，直到将所有需要的用例都创建完成。

# 使用Rose创建用例模型

## 3. 在Rose中创建用例（Use Case）



将每个用例放在单独的包中



# 使用Rose创建用例模型

## 4. 为用例增加文档描述

将每个用例的规格说明增加到Rose系统中。

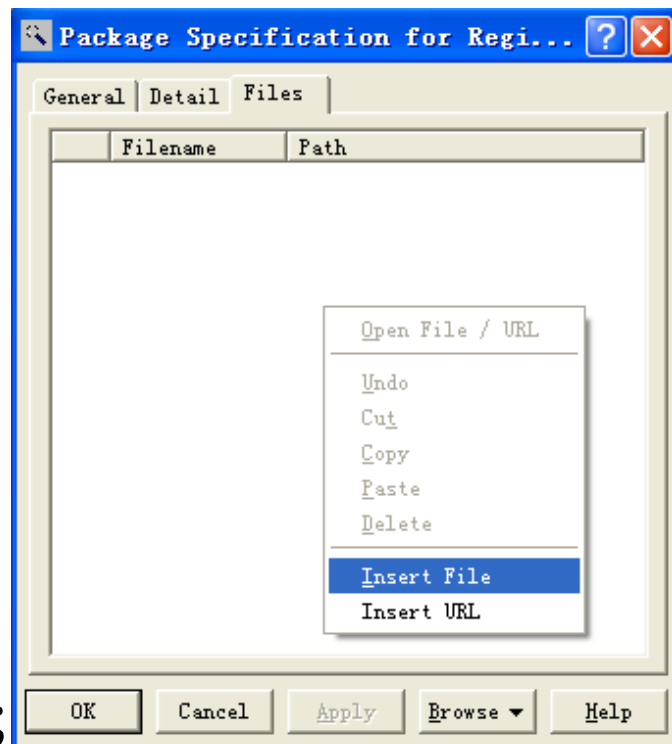
例如，可以将选择课程的规格说明存放在一个单独的Word文档“**选择课程的规格说明.doc**”中，并将此文件作为附加文档添加到Rose中的  
“Register for a course”用例下。

# 使用Rose创建用例模型

## 4. 为用例增加文档描述

将用例的规格说明文档关联到用例的方法如下：

- (1) 在浏览器中的用例（如“Register for a course”）上单击鼠标右键弹出菜单；
- (2) 选择Specification菜单；
- (3) 选择Files标签页；
- (4) 单击右键弹出快捷菜单；
- (5) 选择Insert File菜单，如图所示；



# 使用Rose创建用例模型

## 4. 为用例增加文档描述

- (6) 浏览目录列表选择要关联的文件;
- (7) 单击**打开**按钮, 系统将选择的文件加到说明窗口中, 如图所示;
- (8) 单击**OK**按钮关闭 Specification窗口。



# 使用Rose创建用例模型

## 5. 创建用例图

在Rose中创建用例图的过程如下：

### (1) 加入参与者及用例

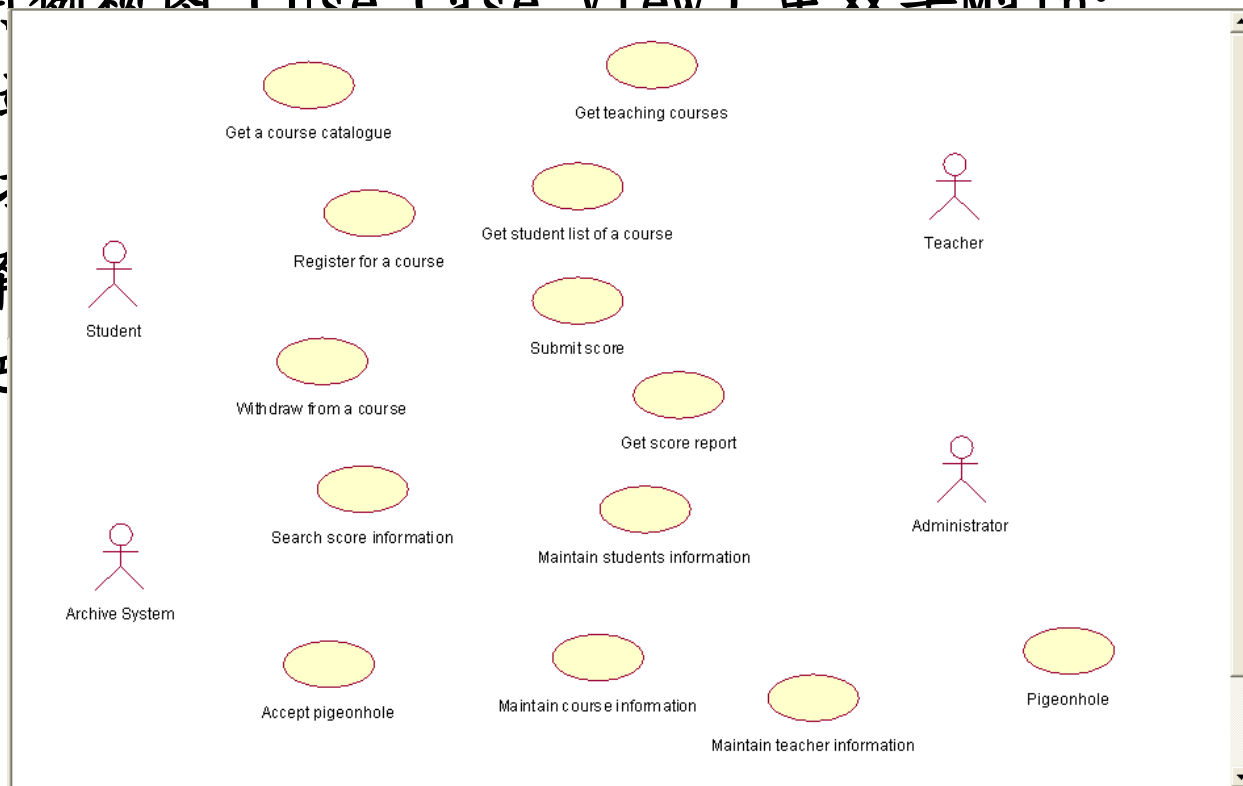
① 在浏览器中的用例视图 (Use Case View) 中双击Main。

② 单击一个actor。

③ 重复步骤 ②，

④ 在浏览器中选择

⑤ 重复步骤 ④ 把



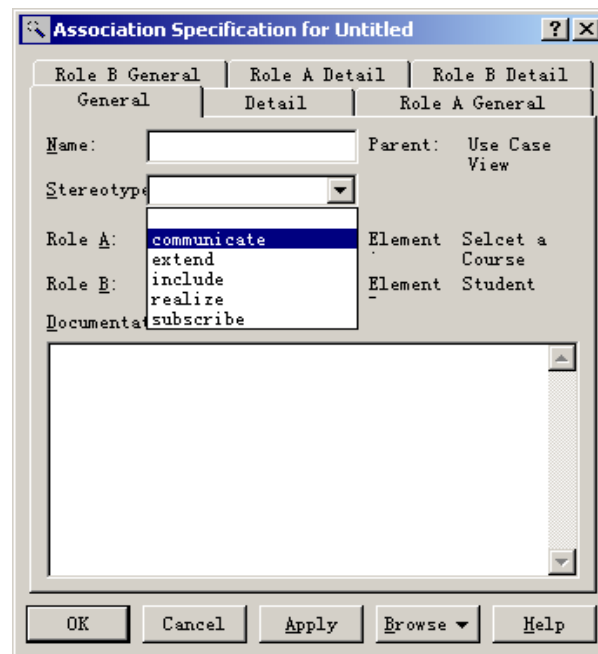
# 使用Rose创建用例模型

## (2) 创建参与者与用例之间的关联关系

在图形工具栏上单击Association（双向关联）或Unidirectional Association（单向关联）图标，在起始actor上单击并拖动到use case上。

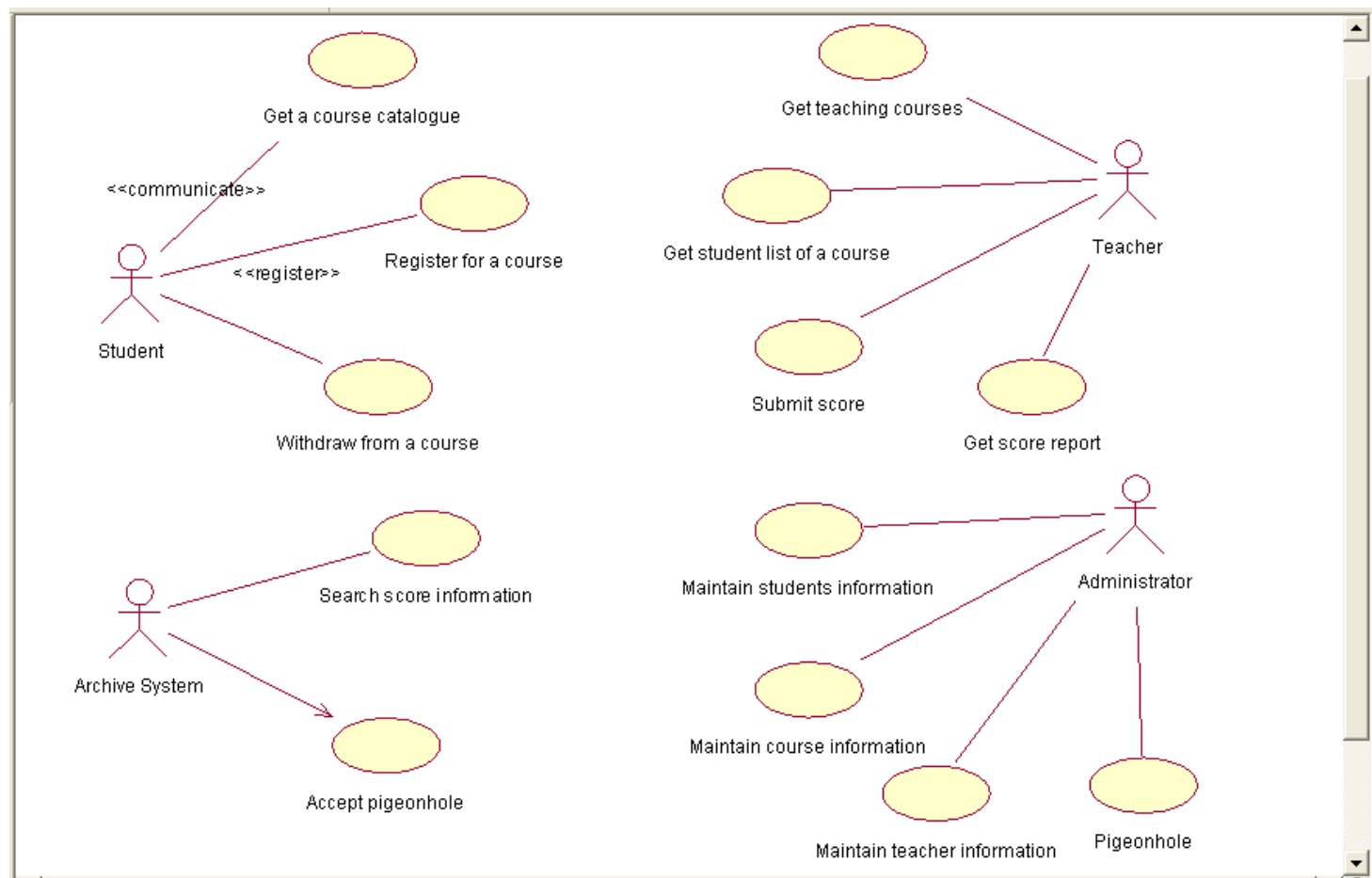
可以按下面的步骤给关联关系增加关系类型（构造型，stereotype）：

- ① 双击Association线，弹出Specification窗口；或使用右键菜单的“Open Specification”项。
- ② 从stereotype后面的下拉框中选择一种类型，如图所示。如果下拉框中没有，则可以输入关系名称。
- ③ 单击OK，关闭specification窗口。



# 使用Rose创建用例模型

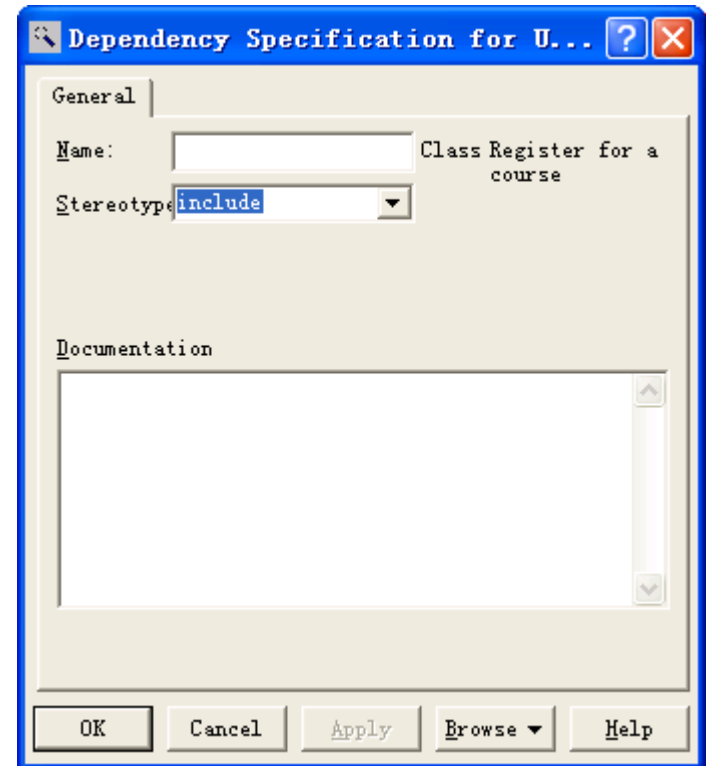
- 添加了参与者与用例之间关系的用例图



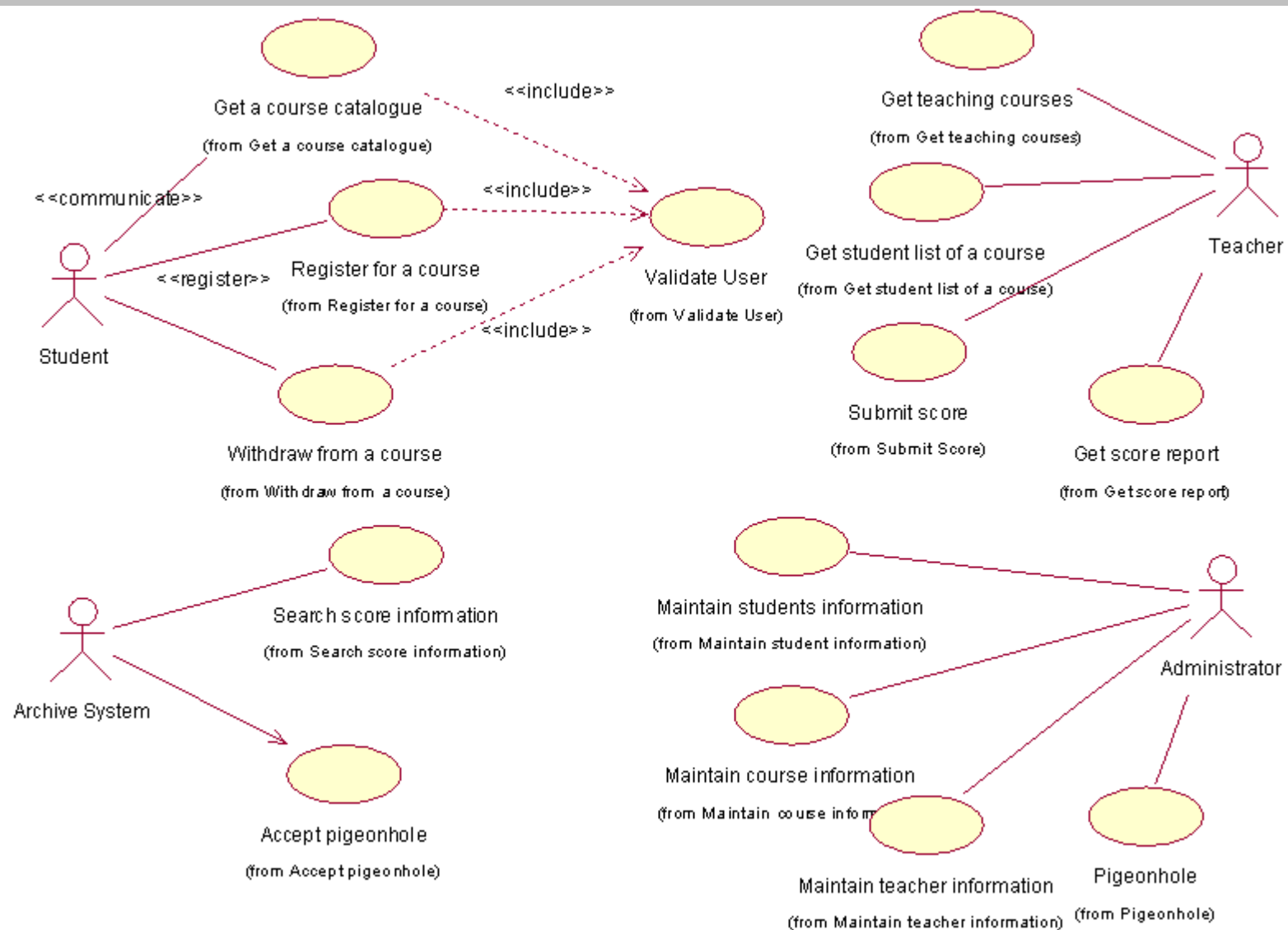
# 使用Rose创建用例模型

## (2) 创建用例之间的关系

- 用例之间主要主要有三种关系：包含(include)、使用(use)、扩展(extend)。
- 在Rose中创建包含( include )关系的步骤如下：
  - ① 在工具条上选择dependency图标；
  - ② 在《包含》用例上单击，拖动到被《包含》的用例上；
  - ③ 双击dependency线，弹出specification窗口；
  - ④ 在stereotype下拉列表中选择“include”，如图所示。
  - ⑤ 单击OK关闭specification窗口。



# 使用Rose创建用例模型





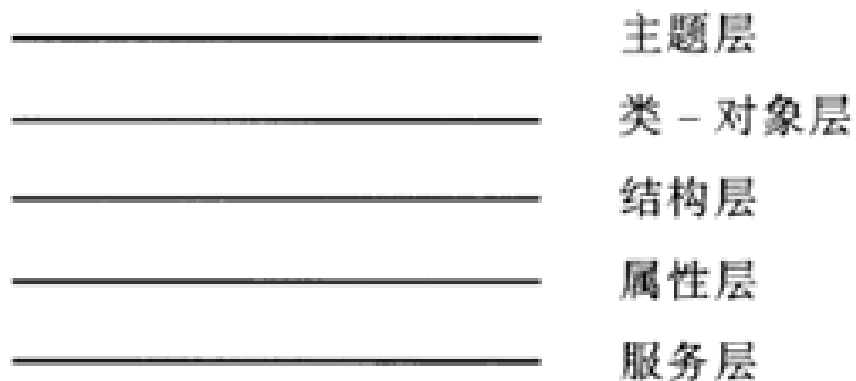
## 6.3 建立对象模型

- 在系统分析阶段，对象建模的主要任务是建立问题域的概念模型。
- 这个模型描述了现实世界中的“类与对象”以及它们之间的关系。
- 在UML中，通过建立类图来表示对象模型。

## 6.3 建立对象模型

- 对象模型的5个层次

- Coad & Yourdon提出，复杂问题(大型系统)的对象模型应该由下述5个层次组成：主题层(也称为范畴层)、类-对象层、结构层、属性层和服务层，如图所示。



# 划分主题

- 在开发大型、复杂系统的过程中，为了降低复杂程度，人们习惯于把系统再进一步划分成几个不同的主题。
- 应该按问题领域而不是用功能分解方法来确定主题。此外，应该按照使不同主题内的对象相互间依赖和交互最少的原则来确定主题。
- 主题可以采用UML中的包来展现。

# 确定类与对象

## 1. 找出候选的类与对象

类与对象是对问题域中有意义的事物的抽象，它们既可能是可见的物理实体，也可能是抽象的概念。我们可以将客观事物分为以下五类：

- 可感知的物理实体，如教学楼、教室等。
- 人或组织的角色，如教师、计算机系等。
- 应该记忆的事件，如演出、交通事故等。
- 两个或多个对象的相互作用，通常带有交易或接触的性质，如购买、教学等。
- 需要说明的概念，如保险法、政策等。

# 确定类与对象

## 1. 找出候选的类与对象

- 另一种更简单的非正式分析方法，是以自然语言书写的需求陈述为依据，把陈述中的名词作为类与对象的候选者，用形容词作为确定属性的线索，把动词作为服务(操作)的候选者。
- 例如，在选课系统中，可以初步确定Teacher（教师）、Student（学生）、Course（课程）、CourseTask（课程任务，指一门课程划分为多个任务）、StudentList（学生名册）、ScoreReport（成绩单）等类与对象。

# 确定类与对象

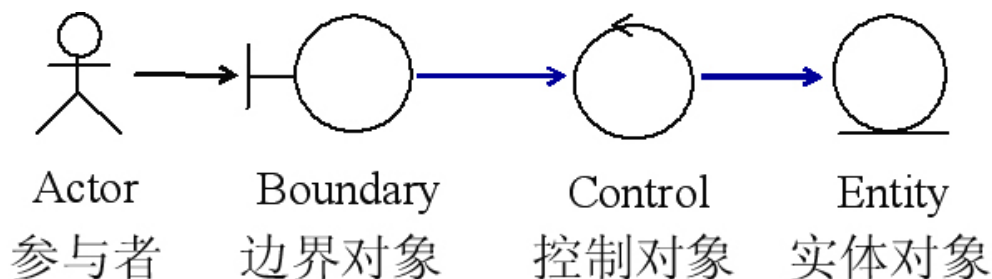
## 2. 筛选出正确的类与对象

严格考察每个候选对象，从中去掉不正确的或不必要的类与对象，仅保留确实应该记录其信息或需要其提供服务的那些类与对象。

# 确定类与对象

## 3. 区分实体类、边界类和控制类

- 在类分析时首先从问题域的实体类入手，如果在建立分析对象模型时区分实体类、边界类和控制类，将有助于理解系统。
- 实体类表示系统将跟踪的持久信息；边界类表示参与者与系统之间的交互；控制类负责用例的实现。其图形表示如图所示。



# 确定结构

## 1. 确定泛化（继承）关系

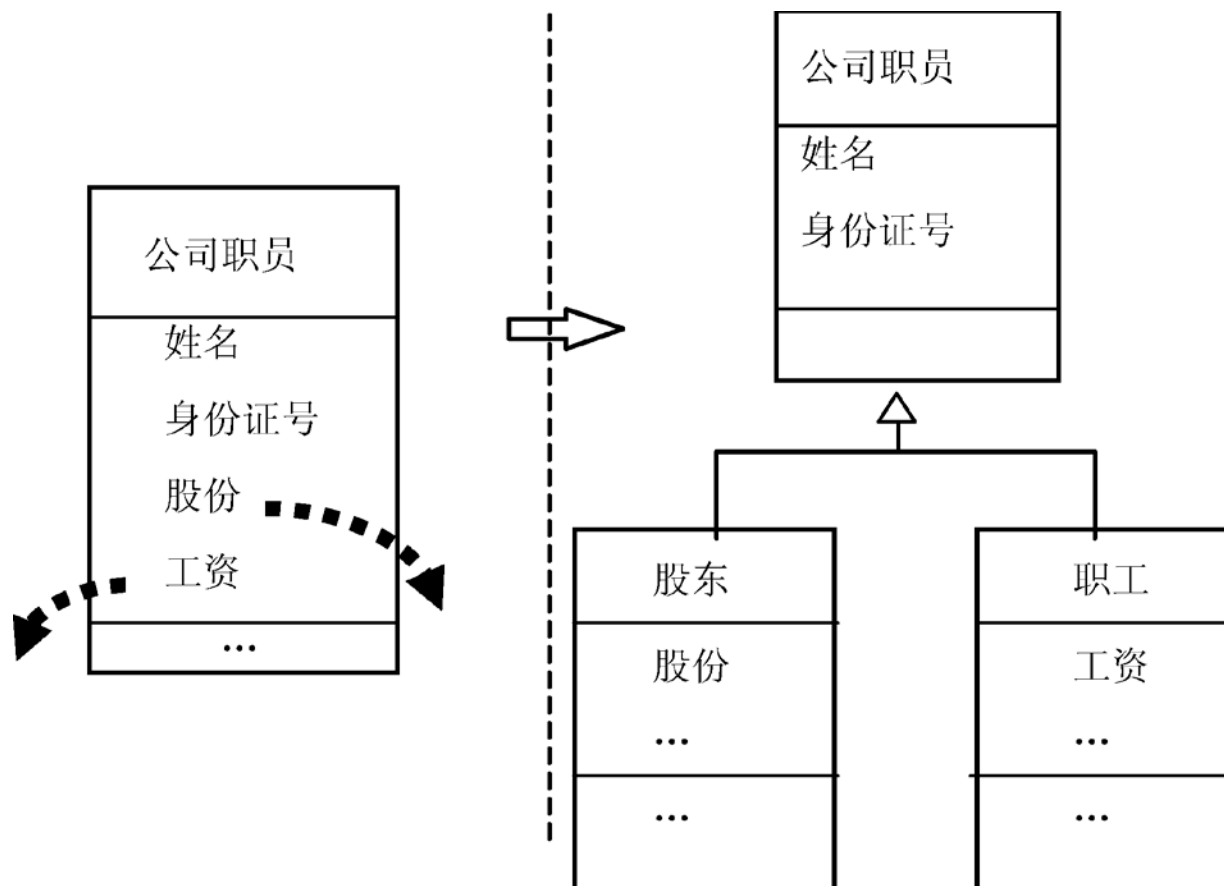
- 学习当前领域的分类学知识。领域分类法往往比较正确地反映了事物的特征、类别以及各种概念的一般性与特殊性。
- 按常识考虑事物的分类。如果问题域没有可供参考的分类方法，可以按自己的常识，从各种不同的角度考虑事物的分类，从而发现泛化（继承）关系。



# 确定结构

## 1. 确定泛化（继承）关系

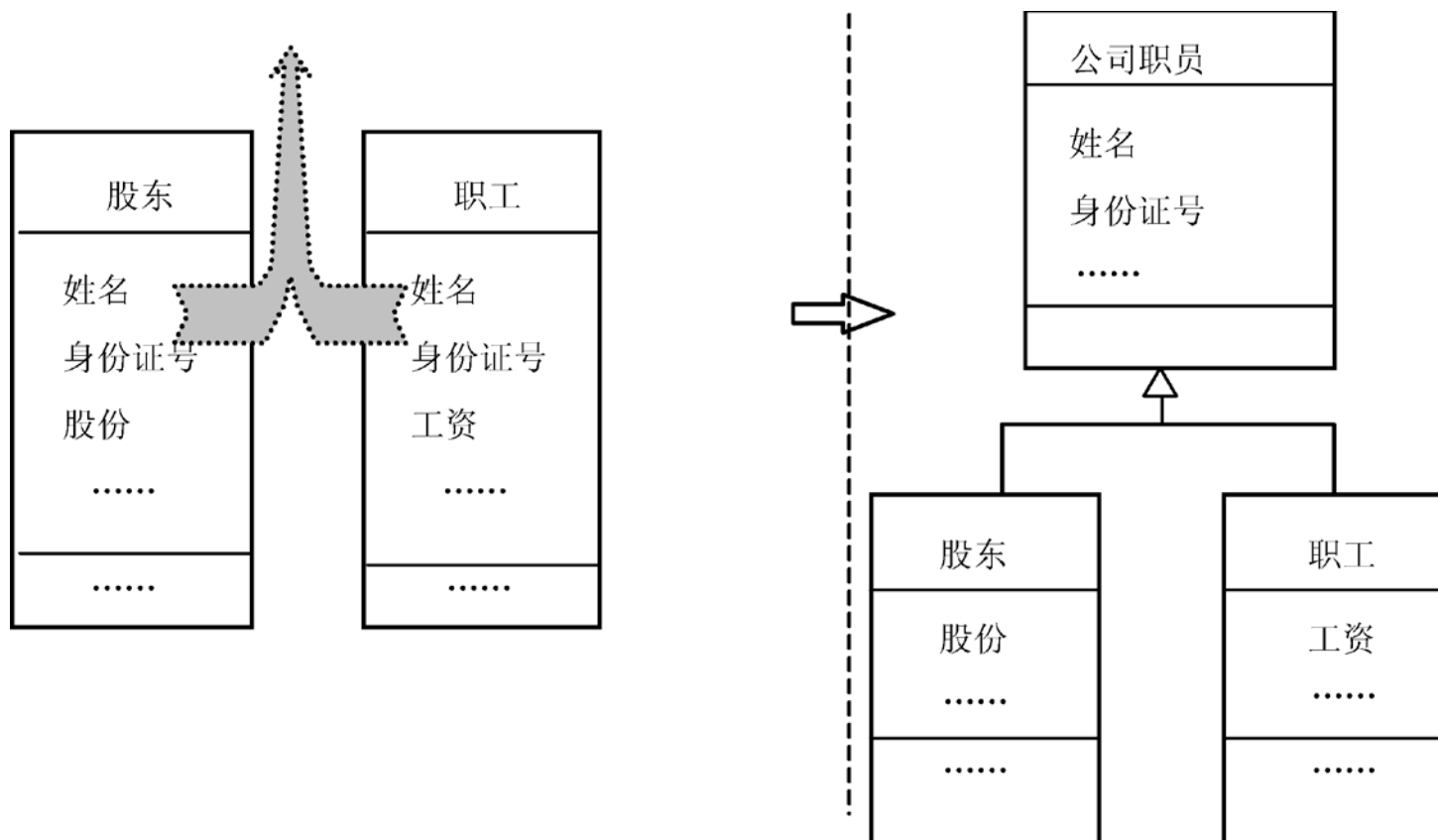
- 考察类的属性与操作，“自上而下”地从一般类发现特殊类。



# 确定结构

## 1. 确定泛化（继承）关系

- “自下而上”地从特殊类抽取出一般类



# 确定结构

## 2. 确定关联

- 标识关联的启发式准则如下：

- (1) 检查指示状态的动词或动词短语，识别动作的主体和客体，从角色寻找关联；
- (2) 准确地命名关联和角色；
- (3) 尽量使用常用的修饰词标识出名字空间和关键属性；
- (4) 应消除导出其他关联的关联；
- (5) 在一组关联被稳定之前先不必考虑实例之间的多重性；
- (6) 过多的关联使得一个模型不可读。

# 确定属性

- 应该仅考虑与具体应用直接相关的属性，不要考虑那些超出所要解决的问题范围的属性。
- 在分析过程中应该首先找出最重要的属性，以后再逐渐把其余属性增添进去。
- 在分析阶段不要考虑那些纯粹用于实现的属性。

# 确定属性

- 标识属性的启发性准则如下：

- (1) 每个对象至少需包含一个属性，例如\_id。
- (2) 属性取值必需适合对象类的所有实例。例如，属性“会飞”并不属于所有的鸟，有的鸟不会飞，因此可以建立鸟的泛化结构，把不同的鸟划分到“会飞的鸟”和“不会飞的鸟”两个子类中。
- (3) 出现在泛化关系中的对象所继承的属性必须与泛化关系一致。
- (4) 系统的所有存储数据必须定义为属性；
- (5) 对象的导出属性应当略去。例如，“年龄”是由属性“出生日期”导出，它不能作为基本属性存在。
- (6) 在分析阶段，如果某属性描述了对象的外部不可见状态，应将该属性从分析模型中删去。

# 确定属性

- 选课系统中类与对象的属性

类与对象	属性
Teacher（教师）	教工号，姓名，系别
Student（学生）	学号，系别，专业，班级，姓名，性别
Course（课程）	课程编号，课程名称，学分，开设学期
<u>CourseTask</u> （课程任务）	课程任务编号，课程编号，任课教师，选课人数，上课时间，教室，学生容量
<u>StudentList</u> （学生名册）	课程任务编号，学号，姓名，专业
<u>ScoreReport</u> （成绩单）	课程任务编号，成绩列表，统计信息

# 确定服务

- 在标识每个对象中必须封装的服务时要注意以下两种服务：
  - (1) **简单的服务**。即每一个对象都应具备的服务，这些服务包括：建立和初始化一个新对象，建立或切断对象之间的关联，存取对象的属性值，释放或删除一个对象。这些服务在分析时是隐含的，在图中不标出，但实现类和对象时有定义。
  - (2) **复杂的服务**。它分为两种：
    - **计算服务**：利用对象的属性值计算，以实现某种功能；
    - **监控服务**：处理对外部系统的输入/输出，外部设备的控制 and 数据的存取。

# 确定服务

- 选课系统中类与对象的服务

类与对象	服务
Teacher（教师）	查询教授的课程信息，查询学生名册。
Student（学生）	累计选课门数。
Course（课程）	累计选课人数。
<u>CourseTask</u> （课程任务）	累计选课人数，将选课的学生加入其学生名册。
<u>StudentList</u> （学生名册）	排序
<u>ScoreReport</u> （成绩单）	成绩分布统计



# 使用Rose建立类图

## 1. 创建类

创建类的过程如下：

- ① 在浏览器中选中逻辑视图（Logical View），单击鼠标右键。
- ② 选择New: Class菜单，一个叫做New Class的类出现在浏览器中。
- ③ 选择新类，键入类的名字。

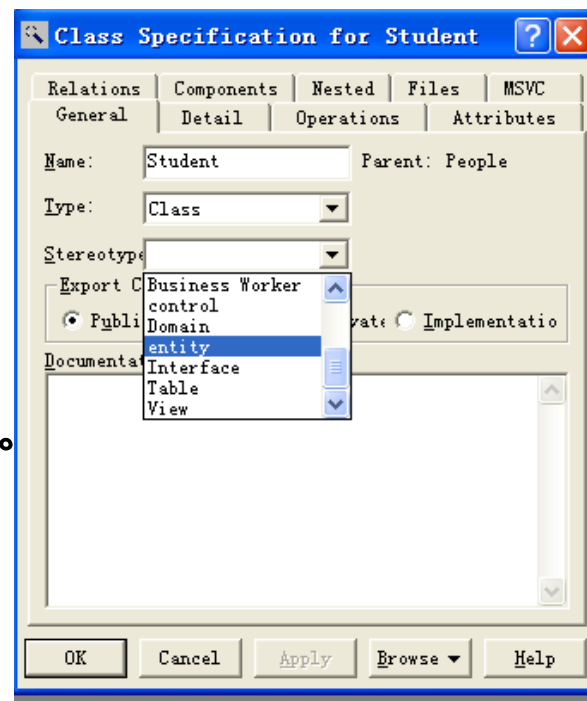
类也有构造型，每个类最少有一种构造型。一些常用的构造型是：**实体类**、**边界类**、**控制类**、**例外类**等。

显然，Student、Teacher和Course类应属于实体类。边界类可能包括RegisterForm（登记表格）、CourseSchedule(课程表)等。

# 使用Rose建立类图

在Rational Rose中为类增加构造型的步骤如下：

- ① 在浏览器中选择类，如Student，并单击鼠标右键，弹出快捷菜单；
- ② 选择Sepecification菜单；
- ③ 选择General页，如图所示；
- ④ 选择或键入stereotype的名字，如选择“entity”表示实体类；
- ⑤ 单击OK按钮关闭Specification窗口。

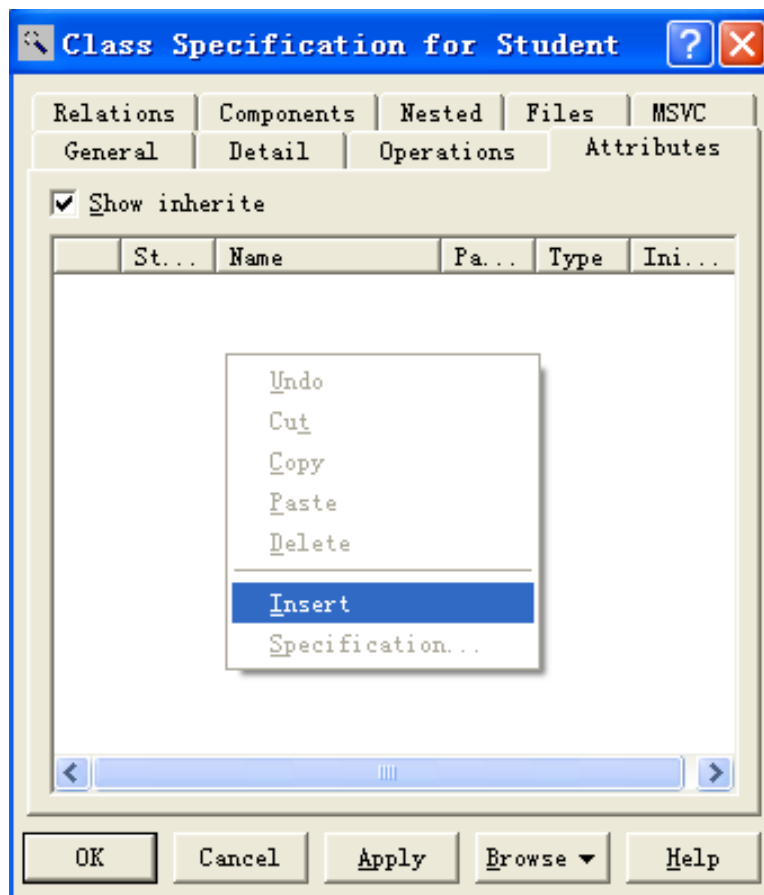


# 使用Rose建立类图

## 2. 给类增加属性和操作

在类的说明窗口中点击

“Attributes”标签页，之后在下面的表格体上单击鼠标右键，从弹出菜单中选择“Insert”菜单，如图所示，就可以输入属性了。



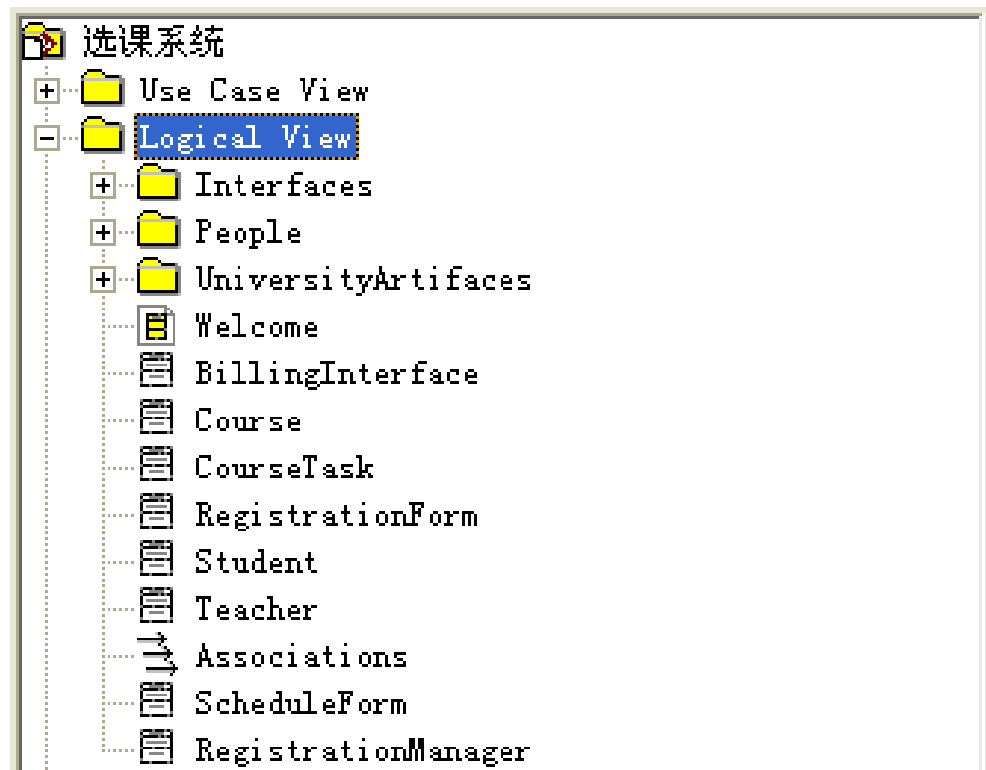
# 使用Rose建立类图

## 3. 将类组织到包中

- 如果系统仅仅包含少数的几个类，我们便可以很容易地管理。
- 通常的系统都包含很多类，你需要一种机制来管理以便更方便使用、维护和复用。这就是包有用的原因。

# 使用Rose建立类图

- 对于选课系统，可以建立如图所示的3个包：界面（Interfaces）、人（People）和学校事件（UniversityArtifaces）。

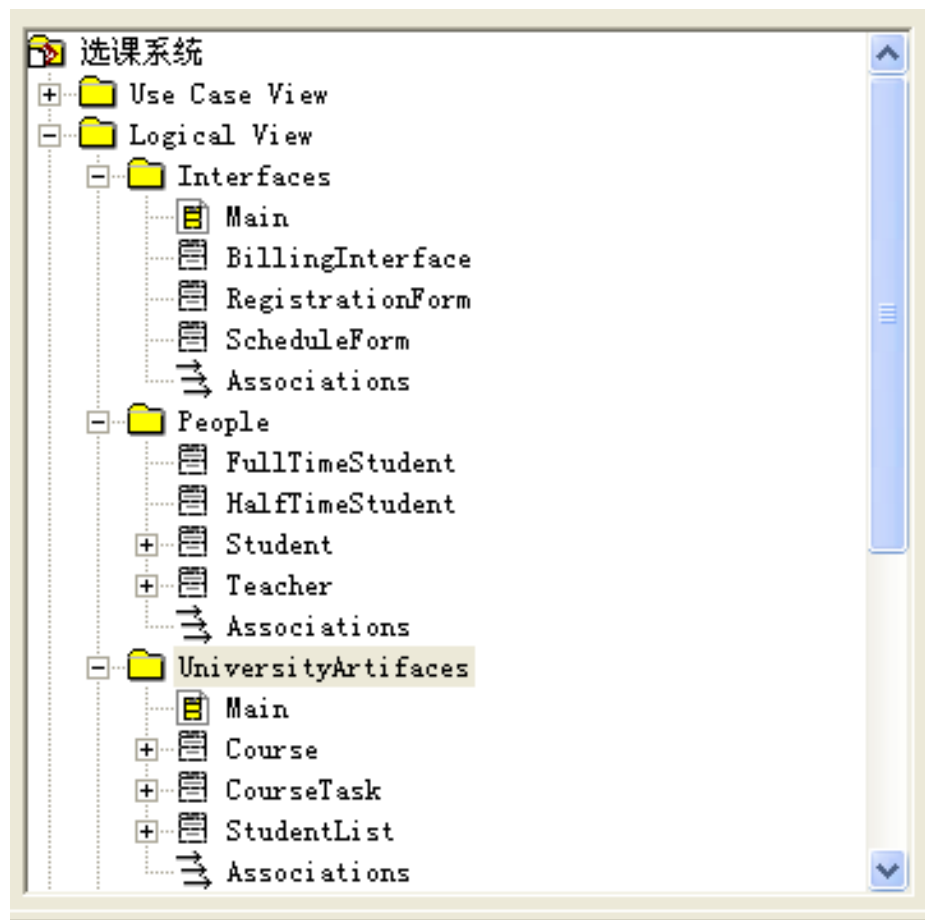


# 使用Rose建立类图

- 当创建好包后，就需要重新安排类的位置，方法如下：

- ① 单击选择浏览器中的类；
- ② 把类拖动到设计的包中；
- ③ 重复上述步骤直到所有的类都重新定位。

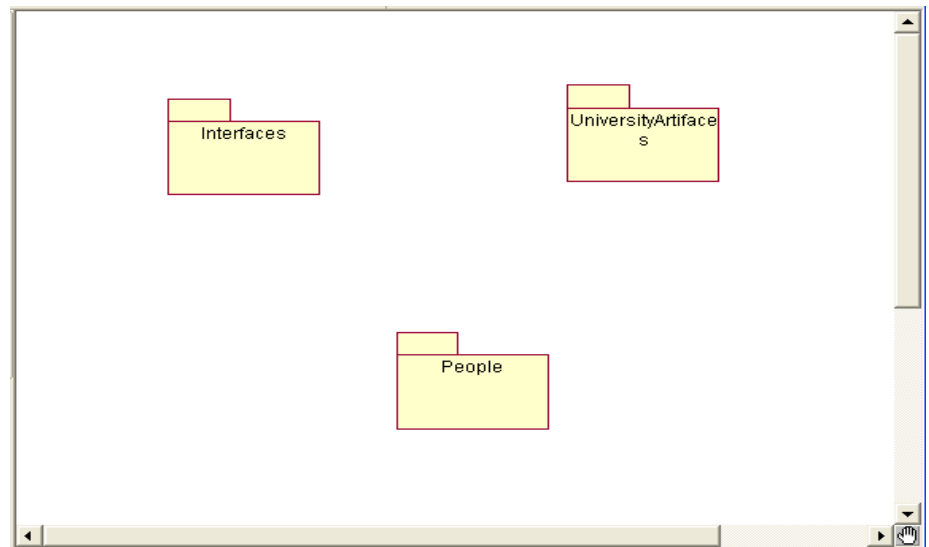
将类组织到包中之后的逻辑视图如图所示。



# 使用Rose建立类图

## 4. 建立和编辑主视图

- 在逻辑视图中的主视图（main）是系统的包图。每个包也有其自己的主要类图，通常显示包的公共类。也可能需要创建其它的类型图。主视图是逻辑视图中典型的高级包视图。
- 增加一个包图的过程是：
  - ① 在浏览器中双击主视图；
  - ② 在浏览器中选择包；
  - ③ 把包拖到视图中。



# 使用Rose建立类图

## 4. 建立和编辑主视图

- 对于每一个包，都可以创建包的主类图，步骤如下：
  - ① 在主视图中双击包；
  - ② **Rose** 会打开包并创建（或显示）这个包的视图；
  - ③ 在浏览器中选择类并拖到视图中；重复步骤③直到所有类都处理完。



# 使用Rose建立类图

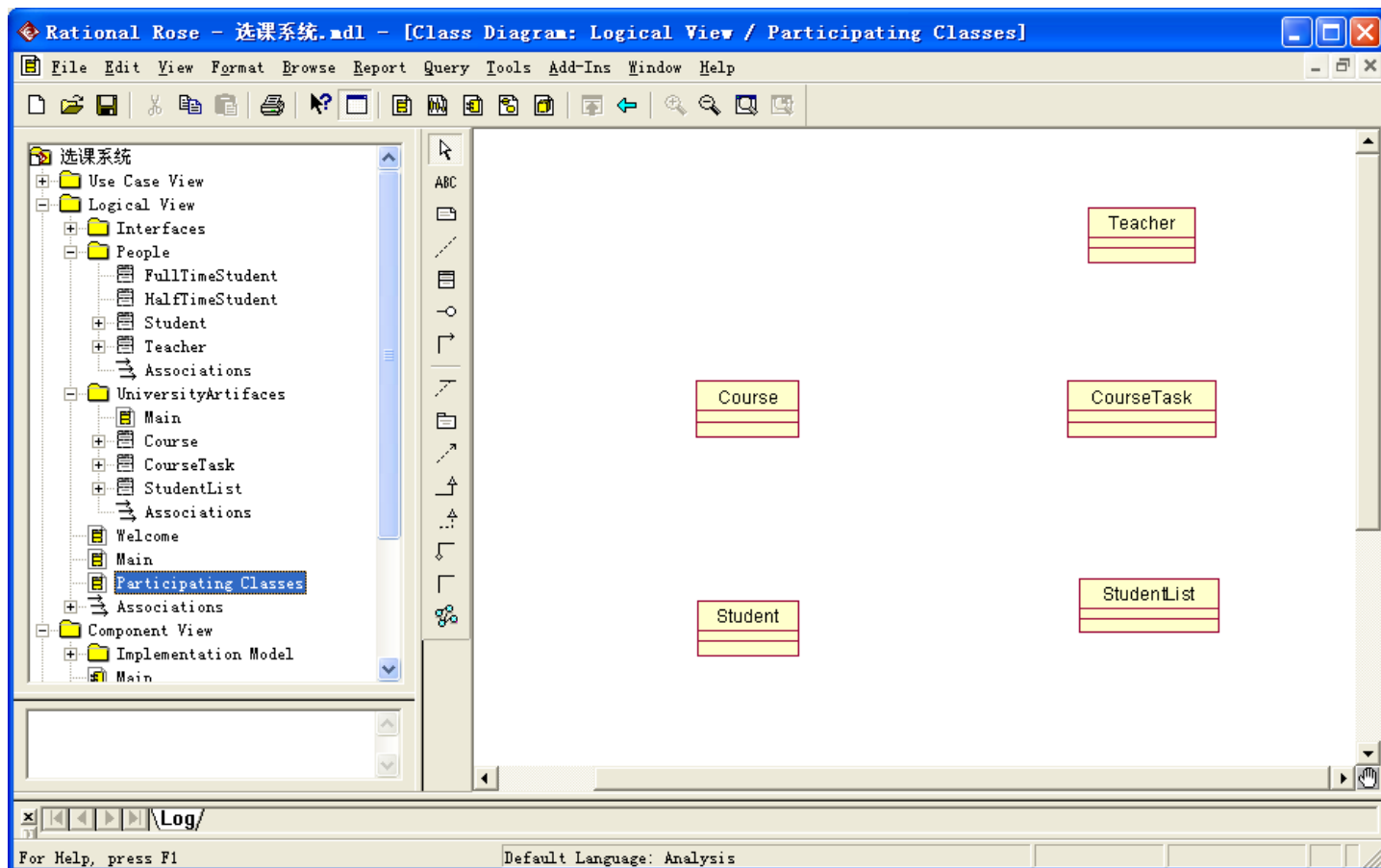
## 5. 建立和编辑分视图

需要时也可以加入类的分视图，类的分视图展现了模型中包和类的另一种“视图”。**添加类的分视图的步骤如下：**

- ① 在浏览器中单击鼠标右键，弹出快捷菜单；
- ② 选择**New:Class Diagram**菜单；
- ③ 系统增加名为“**NewDiagram**”的类图，在此项处于选中状态时，输入类图的名字；
- ④ 在浏览器中双击此类图，将图打开；
- ⑤ 在逻辑视图中选择一个类，并把该类拖到创建的图中；  
重复步骤 ⑤ 直到选择的每个类都放在该图中。

# 使用Rose建立类图

- 在选课系统中增加的分视图



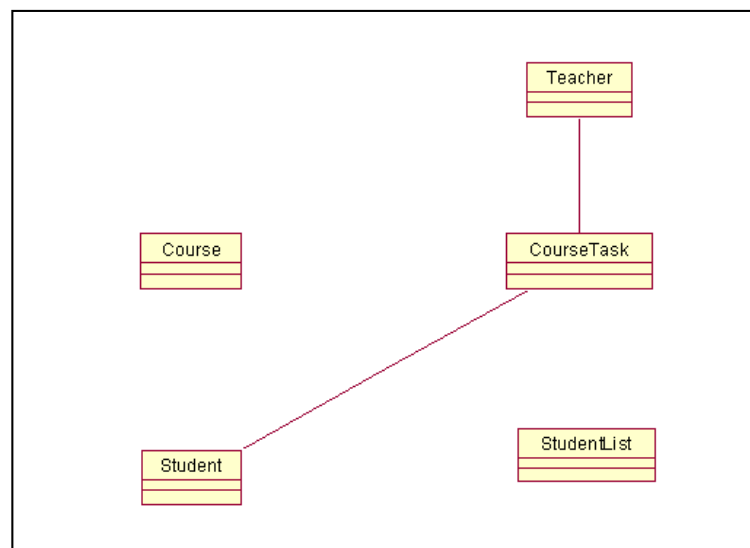
# 使用Rose建立类图

## 6. 为类添加关系

### (1) 关联关系

在Rational Rose中创建关联关系的步骤如下：

- ① 在工具条上选择关联图标；
- ② 在类图中单击关联的一个类；
- ③ 拖动关联关系线到另一个类上。



# 使用Rose建立类图

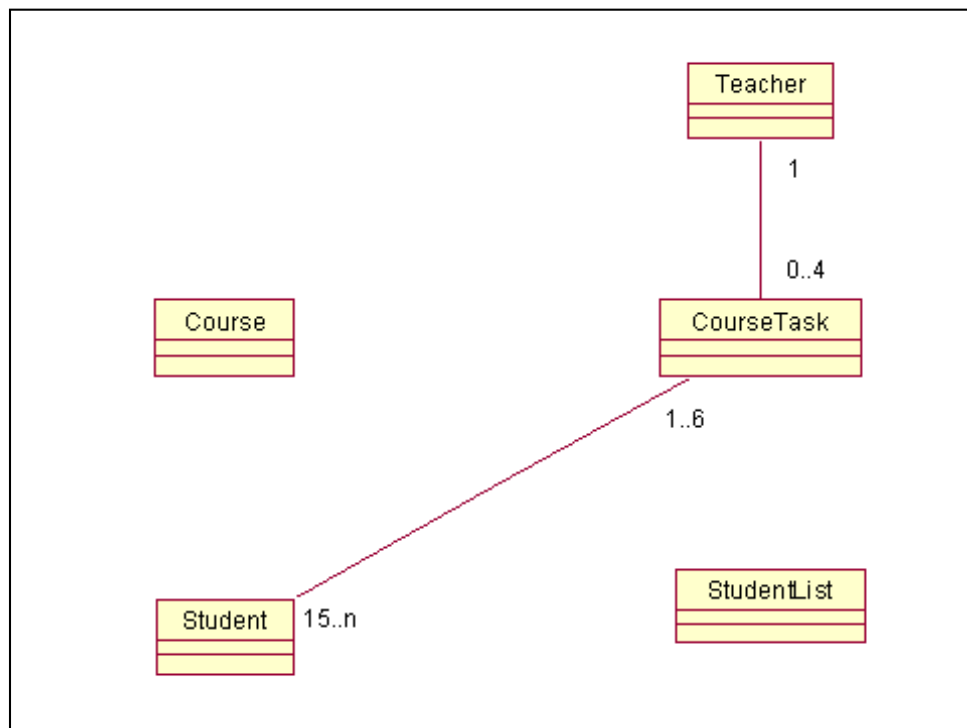
## 标识关联的数量:

- ① 双击关系线，弹出Specification窗口；
- ② 选择Detail页，修改角色 (Role A Detail或Role B Detail)；
- ③ 键入设计的多重性数值。

# 使用Rose建立类图

## 标识关联的数量:

如果规定一名教师每学期最多承担4个教学任务，特殊情况下可以不承担教学任务；少于15人不开课，每名学生每学期选课不超过6门。则增加了关联数量的类图如图所示。



# 使用Rose建立类图

## (2) 聚合关系

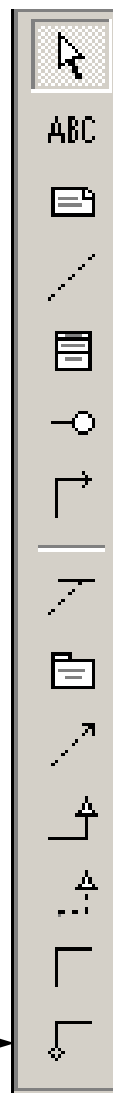
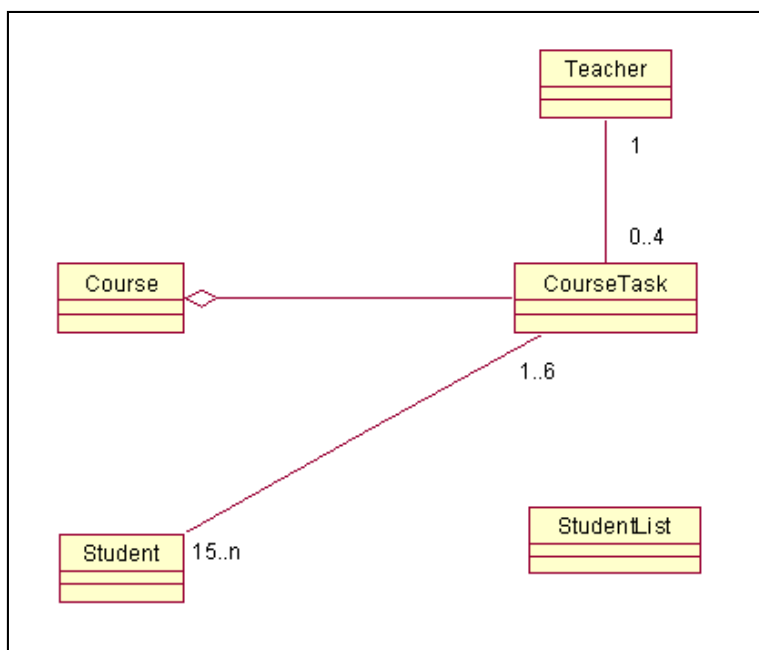
- 聚合是关联关系的特殊形式——整体和部分的关系。聚合是部分或包含的关系。
- 例如：在一个学期中，很多班级的学生都需要上同一门课程（Course），通常会将此门课程的教学划分为多个教学任务（CourseTask）。Course和CourseTask可以认为是聚合关系。

# 使用Rose建立类图

## (2) 聚合关系

在Rational Rose中创建聚合关系的方法如下：

- ① 从工具条上选择聚合关系（aggregation）图标；
  - ② 单击属于“整体”的类，拖动鼠标到属于“部分”的类。
- 增加了聚合关系的类图如图所示。



# 使用Rose建立类图

## (3) 关联类

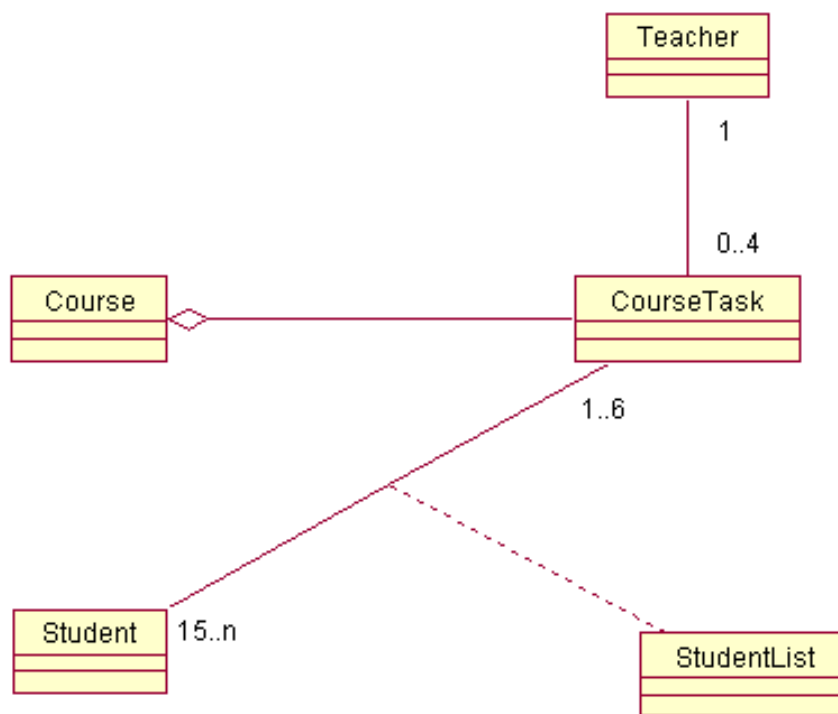
- CourseTask和Student之间是多对多的关系，一个很容易想到的问题是学生选课的成绩存在哪里？
- 显然，对于每个CourseTask对象（一份教学任务），选课结束后都有一份学生名单，也就是StudentList对象，是StudentList将CourseTask对象及Student对象关联在一起的，
- 因此，应将StudentList作为关联类。



# 使用Rose建立类图

## (3) 关联类

- 在Rational Rose的工具条上选择关联类（Association Class）图标，在CourseTask和Student的关联线上按下鼠标，再拉到StudentList上，即完成了关联类的增加。



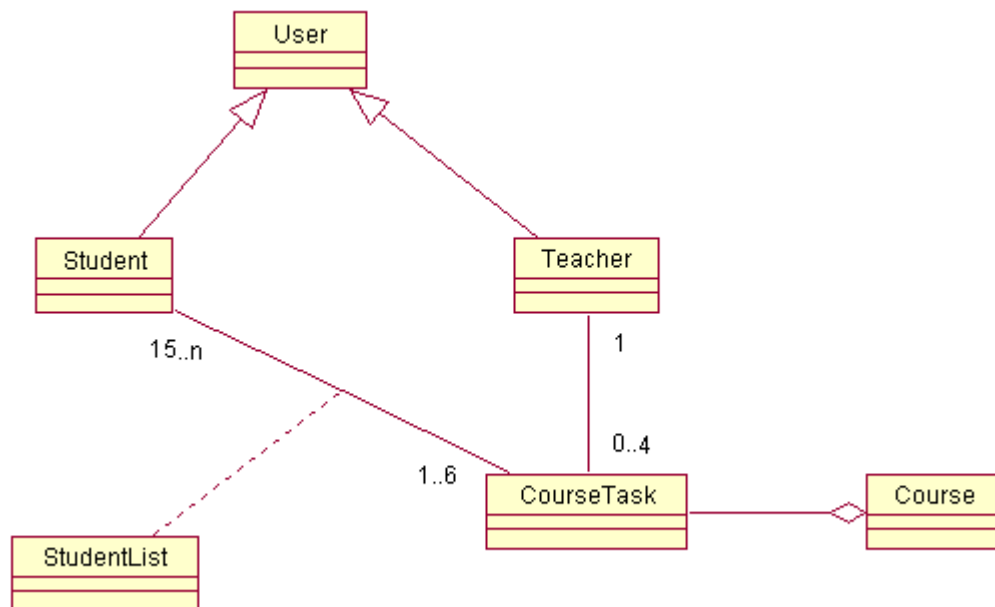
# 使用Rose建立类图

## (4) 泛化关系

在Rational Rose中创建继承关系的方法如下：

- ① 从工具条上选择继承关系 (Generalization) 图标;
- ② 单击属于“子类”的类，拖动鼠标到属于“父类”的类。

在选课系统中，如果考虑到学生、教师或系统管理员登录系统时需要进行权限控制，可以增加用户 (User) 类。



## 6.4 建立动态模型

- 在开发交互式系统时，动态模型起着很重要的作用。
- 在UML中动态模型的描述工具有顺序图、通信图和状态图。

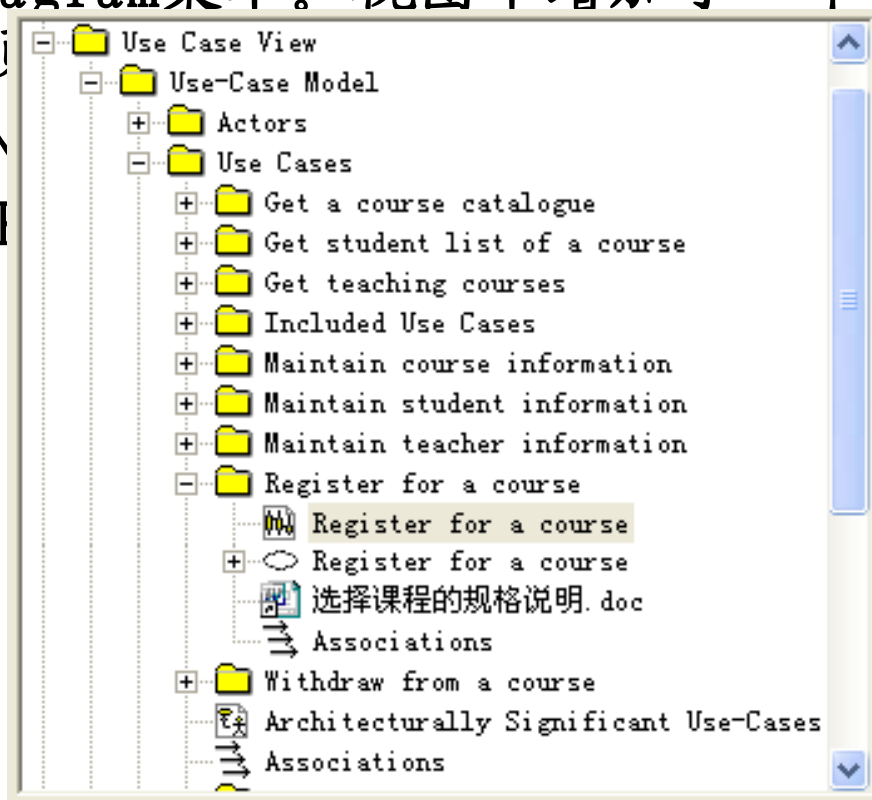
# 顺序图

- 用例图中的事件流是由文本表示的，事件流描述的是用例实现的过程，也称为场景（scenarios），可以用顺序图表示场景。
- 顺序图按照时间顺序显示对象之间的交互关系。它描述场景中的对象和类以及在完成场景中定义的功能时对象间要交换的信息。

# 顺序图

在Rational Rose中创建顺序图的过程如下:

- ① 在浏览器中选择某个用例，单击鼠标右键，弹出快捷菜单；
- ② 选择New: Sequence Diagram菜单。视图中增加了一个名字为“NewDiagram”的顺序图；
- ③ 选择NewDiagram，键入名称。例如，在选课系统中，可以给“注册课程”增加顺序图，如图所示。



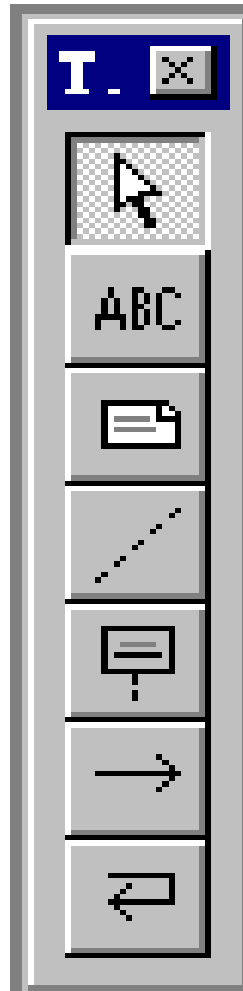
# 顺序图图标

选择工具

注释

对象

反身消息



文本

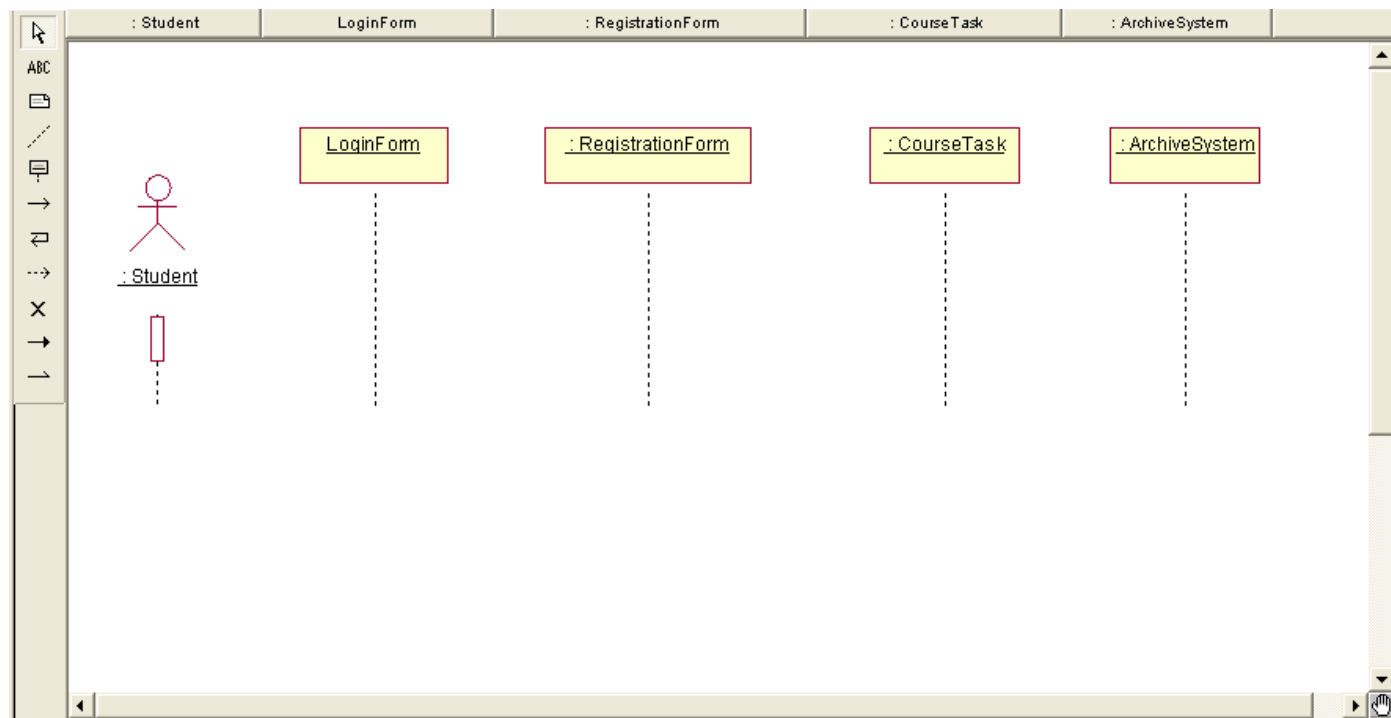
注释指示

消息

# 顺序图

## 创建顺序图的对象和消息:

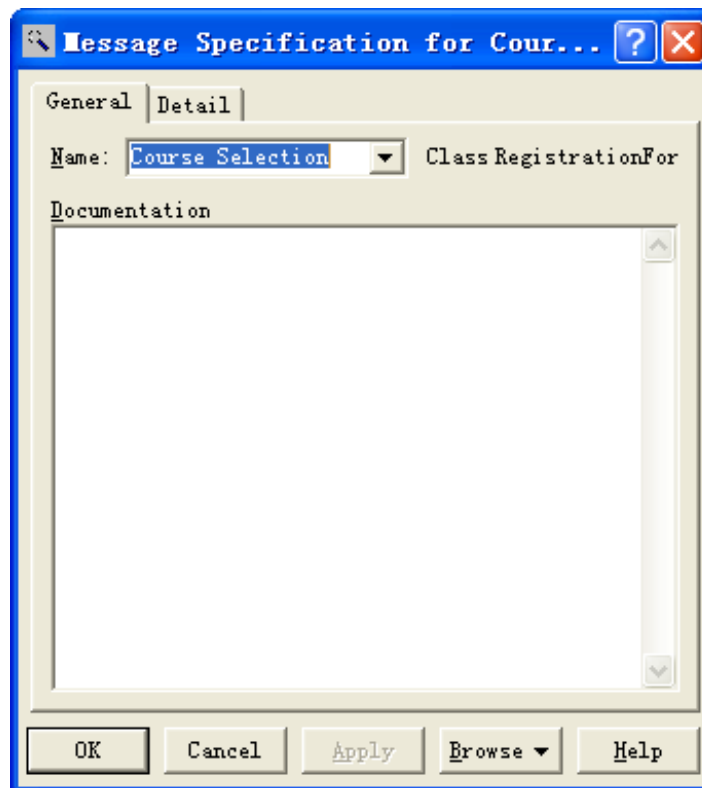
- ① 在浏览器中双击顺序图将其打开;
- ② 在浏览器中选择actor, 将actor拖到顺序图中;
- ③ 在浏览窗口中查找需要的类, 并依次将其拖到顺序图上, 在顺序图上显示的是类对象。



# 顺序图

## 创建顺序图的对象和消息:

- ④ 在工具条上选择消息图标;
- ⑤ 在发出消息的对象上单击鼠标, 把消息线拖到接收消息的对象上;
- ⑥ 用鼠标双击消息线, 弹出如图所示的消息窗口, 在文本框name中键入消息名称;

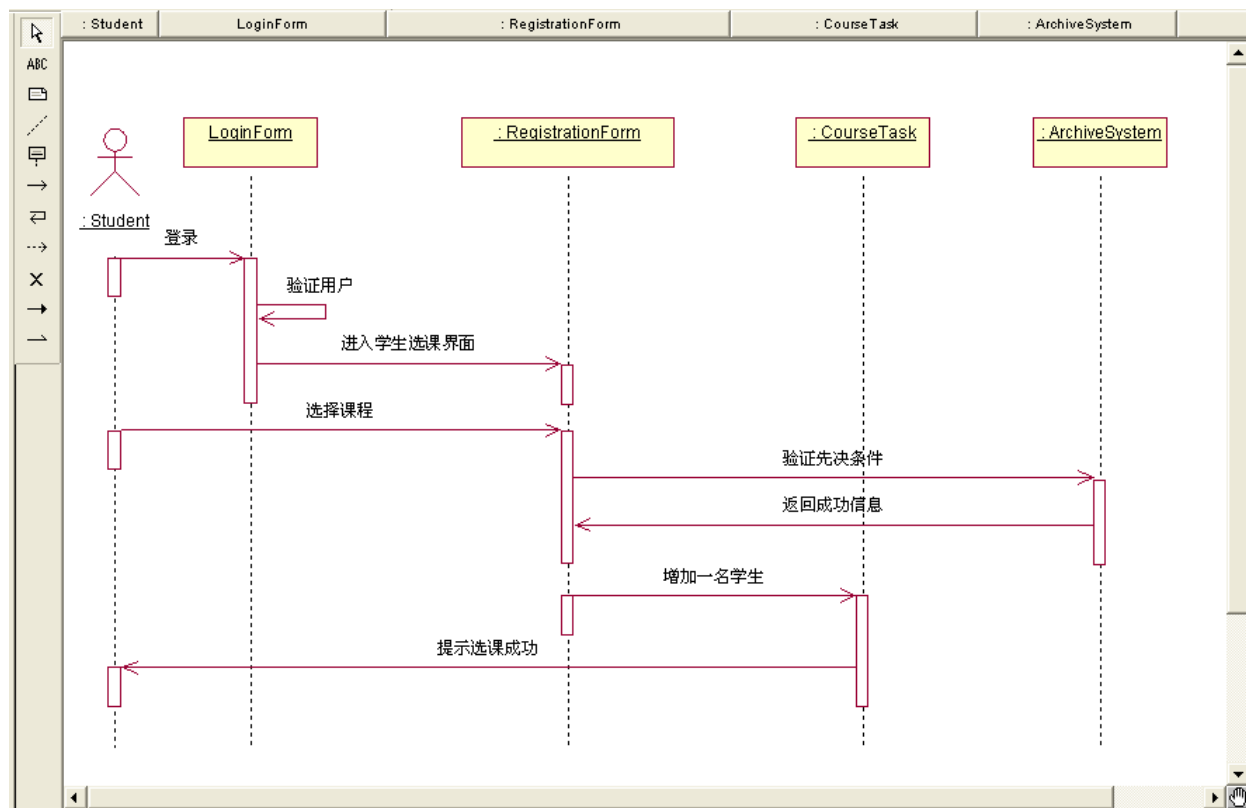




# 顺序图

创建顺序图的对象和消息：

⑦ 重复步骤 ④ 到 ⑥ 直到所有消息都添加到顺序图中。

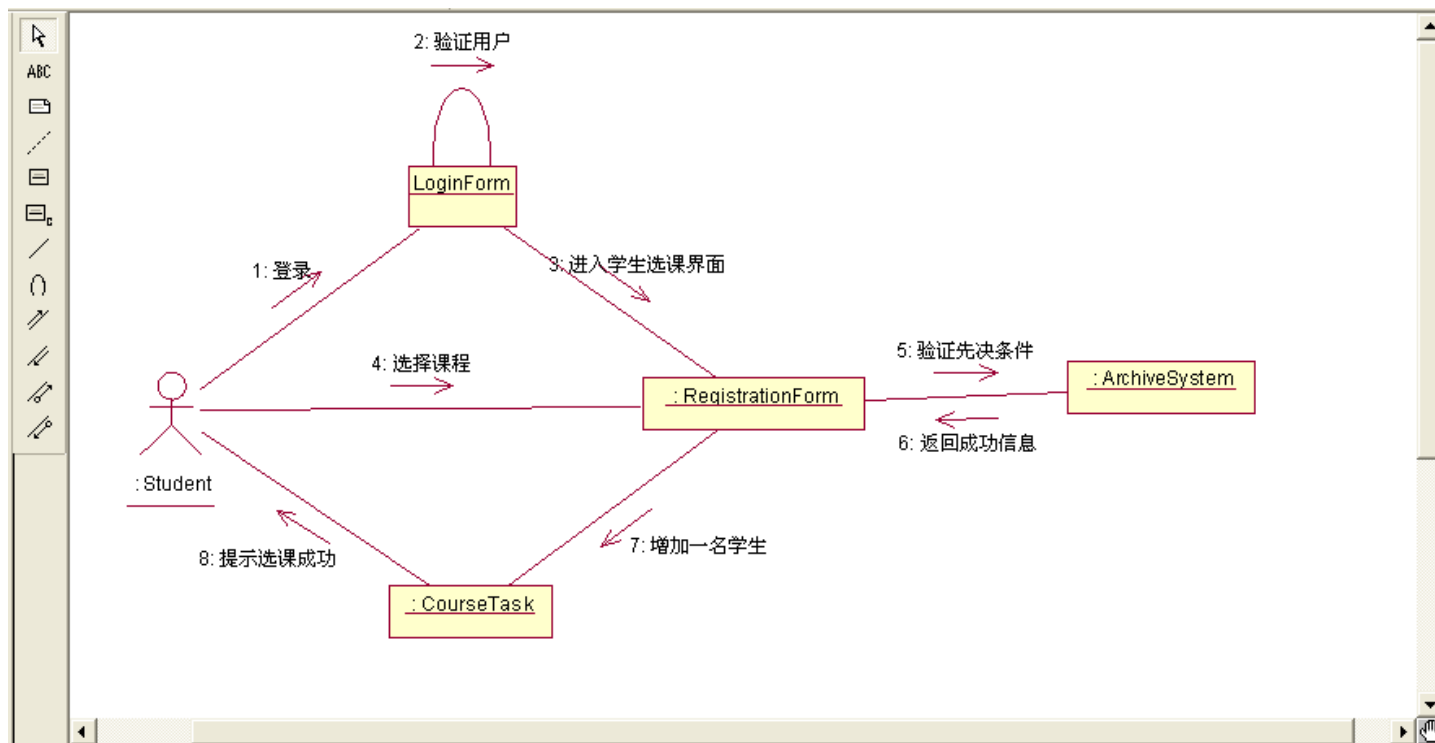


# 协作图

- 协作图也称通信图，是顺序图的另一种表示形式，用于描述相互协作的对象间的交互关系和链接关系。
- 一般情况下，当表示涉及很多对象的模型时，协作图比顺序图更形象。
- 另外，与顺序图不同，对象之间的实线可能表明这些对象的类之间需要关联。

# 通信图

- 在Rose中根据顺序图建立通信图非常简单。
- 只要在顺序图上按“F5”键，就可以自动将顺序图转换成通信图。



# 状态图

- 状态图由对象的各个状态和连接这些状态的转换组成。
- 通常，用一张状态图描绘一类对象的行为，它确定了由事件序列引出的状态序列。
- 不是任何一个类都需要有一张状态图描绘它的行为，只针对具有明显的状态特征并且具有比较复杂的状态—事件—响应行为的类，才需要画状态图。
- 在选课系统中，CourseTask类的对象具有比较明显的状态特征，其状态有：初始状态、可选状态、人满状态、关闭状态。

# 状态图

## 创建状态图的过程如下：

- ① 在浏览器中选择某个类，如选择CourseTask，单击鼠标右键，弹出快捷菜单；
- ② 选择New: Statechart Diagram菜单。视图中增加了一个名字为“NewDiagram”的状态图；
- ③ 选中NewDiagram，键入名字，如StatechartDiagram。
- ④ 双击StatechartDiagram打开应用窗口，就可以利用UML工具栏的图标元素画状态图了。

# 状态图图标

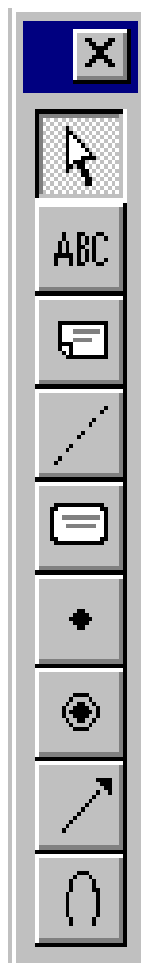
选择工具

注释

状态

终止状态

反身转换



文本

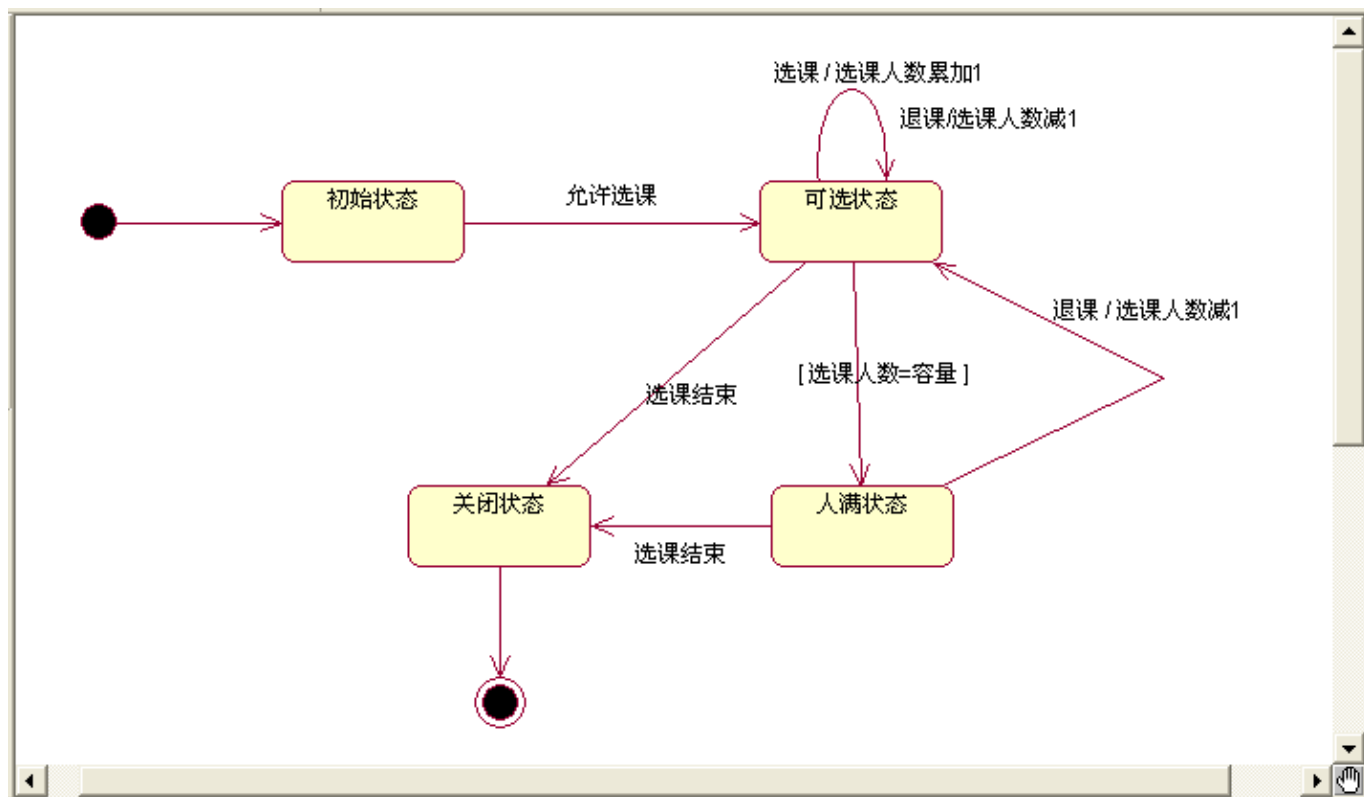
注释锚

初始状态

转换

# 状态图

在选课系统中，可以给CourseTask增加状态图。





That's All!