



《软件工程》实验指导书

蔡勇

宜宾学院计算机科学与技术系

2008 年 3 月



目录（Contents）

第一章 概述（Overview）	4
一、实验目的：	4
二、面向专业：	4
三、先修课程：	4
四、实验环境:.....	4
五、实验考查的目的.....	5
六、实验要求.....	5
七、实验成绩的评分方法和标准	6
八、具体实验内容评测： :.....	6
第二章 实验内容(Experiments).....	8
实验一 Microsoft Visio软件的使用	8
实验二 软件需求分析	9
实验三 软件概要设计	10
实验四 软件详细设计	11
实验五 软件测试和调试	12
实验六 UML用例图的设计和实现	13
实验七 UML活动图的设计和实现	14
实验八 UML序列图和协作图的设计和实现	15
实验九 UML类图的设计和实现	16
实验十 UML构件图和布署图的设计和实现	17
第三章 实验报告(Reports).....	18
实验一 Microsoft Visio软件的使用	18
实验二 软件需求分析	20
实验三 软件概要设计	24
实验四 软件详细设计	27
实验五 软件测试和调试	30
实验六 UML用例图的设计和制作	32
实验七 UML活动图的设计和实现	34



实验八 UML序列图和协作图的设计和实现	36
实验九 UML类图的设计和实现	38
实验十 UML构件图和布署图的设计和实现	41
附录1 如何实现数据库操作	43
附录2 设计范例1	46
附录3 设计范例2	55
附录4 Rose2003 软件简介	97
附录5 UML软件实验部分操作	113
学校网上选课系统的用例图的设计和实现	113
Add Course（添加课程）用例的活动图设计和实现	127
选课用例(select Course)的序列图和协作图设计和实现。	144
选课系统的类图设计和实现	170
选课系统的组件图和布局图设计和实现。	205
附录6 课程设计题目示例	227
附录7 Microsoft Visio软件的使用	232



第一章 概述（Overview）

一、实验目的：

软件工程是一门实践性很强的课程，上机实验是其重要的环节。本实验指导书配合《软件工程》课程的学习而制订的。实验目的是：

1. 能按照软件工程的思想，采用面向过程或面向对象的方法开发出一个小型软件系统。
2. 在软件系统开发过程中，能综合利用一门编程语言、数据库概论和软件工程三门课程的知识。
3. 培养良好的软件开发习惯，了解软件企业文化。

二、面向专业：

计算机科学与技术

三、先修课程：

一门计算机高级语言、数据库概论

四、实验环境：

1. 单机模式

操作系统： Windows9x, Windows2000

开发工具： C 语言或 VB 或 VC 或 Java



2. 客户机/服务器模式 (C/S)

操作系统: Windows9x, Windows2000

数据库管理系统: Access, SQL Server, Oracle8i

开发工具: VB6.0, Delphi6.0, PB7.0, C++Builder, VC++, C#, VB.Net

3. Web 服务器模式 (B/S)

操作系统: Windows2000 Server

数据库管理系统: Access, SQL Server, Oracle8i

开发工具: C#, ASP.Net, Java

五、实验考查的目的

1、通过实验考查使学生更好地巩固已学的知识技能,进一步提高软件开发能力和独立分析问题,解决问题的能力。

2、通过实验考查,使教师掌握教学效果,检查实验教学质量,总结经验,研究规律,进一步改进实验教学工作,使之提高到一个新的水平。

3、用一个学校的网上选课系统作为实例来贯穿整个 UML 建模过程,让学生从实践中了解 UML 图例的各种用法,并从软件开发的实例中掌握标准的建模语言及对应的开发方法。而不是单纯地学习 UML 图的绘制。

六、实验要求

1、预习(报告)

要求对实验的名称、目的、原理、仪器设备、数据记录方式、实验项目的调查研究、分析设计、测试修改等进行必要的准备,未经准备,不得进实验室操作。

2、实验操作

要求实验操作准确,实验现象明显,数据记录及时、数据处理和测试修改准确,如出现违章操作,拼凑,抄袭数据等情况,则应严格扣分。

3. 实验纪律



学生应遵守实验室纪律，爱护仪器设备，实验物品，保持实验室环境卫生。如出现下列情况则应严格扣分：

- a) 实验迟到或未完成实验擅自离开实验室。
- b) 操作违章或不守纪律，损坏公物。
- c) 不讲卫生，影响实验室环境。
- d) 实验完成后，未做好整理工作

七、实验成绩的评分方法和标准

1、实验成绩实行优秀、良好、中等、及格、不及格五级记分，或者采用百分制记分。

2、实验成绩的评定采用实验报告、作品（软件系统和建模）和考勤三项来考核，然后取比例值。具体比例实验报告 40%，作品 40%，考勤 20%，合计 100%。

3、课程成绩计算

理论和实验合为一门课程的期末总成绩中理论考试占总成绩的 60%，实验考查成绩占总成绩的 30%，平时成绩占总成绩的 10%。

八、具体实验内容评测：

1、文档部分实验评价标准

不能按要求或不能按时完成，成绩评为“不及格”（60 分以下）；

能按软件工程规范要求完成文档，软件调试、测试、运行通过，成绩评为“及格”（60—74 分）；

全部文档质量较好，软件有一定实用价值，成绩评为“良”（75—89 分）；

文档质量好，软件有难度，能反映同学较强能力，成绩评为“优”（90—100 分）

2、UML 建模实验评价标准

因为已经有参考建模方案，所以一般情况下都能完成预定设计。如果只是按照给定步骤去做，其分数只能在 85 分以下。为了证明学生已经掌握设计所涵盖的知识点，应该向学生提一些问题，例如如何设计及其原理等问题。



另外，设计的可读性均包含在正确性中记分，不再单列考核标准。

一般遵循如下规律评价：

- (1) 严格控制 90 分，其标准是有创意。
- (2) 85 分以上，必须全部正确，并有一定改进或者能正确回答设计中的问题。
- (3) 有少许失误，可给 75--83 分。
- (4) 错误不多，给 60--73 分。
- (5) 没有完成特殊规定的要求，则不予及格。

因为 UML 建模实验主要是锻炼学生，所以应该鼓励他们不要作为负担，提高钻研问题的兴趣，放手去做。另外，评分标准也可以只设“通过”、“没通过”和“优秀”3 档以激发学生的学习兴趣。

为了保证实验的质量，要求同学们对实验内容进行预习，重视对实验内容的理解和体会，认真完成每次“实验小结”，“实验小结”作为实验成绩的主要评价成分。

实验完后若需交建模电子文件的，其文件名为：<学号>_<姓名>_<实验名称>，例如 021103049_张三_程序流程图.vsd。



第二章 实验内容(Experiments)

实验题目：在以下范围内任选一道题目或自定一个题目。

1. 编制一个万年历程序
2. 开发科研管理系统
3. 开发住房管理系统
4. 开发航空公司订票系统
5. 开发人事管理系统

可参照附录 6 选择题目。

实验一 Microsoft Visio 软件的使用

1. 实验目的：

- 1) 熟悉 Visio 的工作环境及组成；
- 2) 掌握用 Visio 软件绘制图表的基本操作；
- 3) 能熟练全用 Visio 软件绘制各种较复杂的专业图表；
- 4) 掌握各种图表文档创建方法。

2. 实验要求：

要求能够熟练运用 Visio 软件所提供的菜单、工具、模型等制作图形或图表；能用 Visio 软件所提供专业图形模板，来自行绘制出专业化、高质量的图形或图表。

3. 实验内容：

使用 Microsoft Visio2002 来设计一个基本流程图模型。

4. 实验步骤：

- 1) 通过打开模板并向图表添加形状来开始创建图表。
- 2) 在图表中移动形状并调整形状的大小。
- 3) 向图表添加文本。
- 4) 连接图表中的形状。
- 5) 设置图表中形状的格式。
- 6) 保存图表以示完成，并向演示图表。

5. 实验学时：2 学时。

6. 实验结果：上交所制作的流程图文件和实验报告。



实验二 软件需求分析

1. 实验目的：

- 1) 根据所选定题目进行需求分析工作；
- 2) 通过实例掌握结构化数据流分析技术；
- 3) 进行业务需求分析、用户需求、功能需求、非功能需求分析；
- 4) 写出需求规格说明书（含数据流图）

2. 实验要求：

要求做到使用结构化数据流分析技术分析课题需求，写出详细的数据流图和数据字典，数据流图的基本处理的个数不得少于 5 个。

3. 实验内容：

用结构化数据流分析技术进行软件系统需求分析，得出系统得数据流图和数据字典。

4. 实验步骤：

- 1) 到相关单位进行需求分析。
- 2) 综合利用 Internet 网和相关书籍整理并完善需求分析。
- 3) 画出系统数据流图（分清系统是事务型还是加工型）。
- 4) 得出系统数据字典。

5. 实验学时：4 学时。

6. 实验结果：上交实验报告。



实验三 软件概要设计

1. 实验目的:

- 1) 建立初始结构图;
- 2) 对初始结构图进行改进、优化;
- 3) 写出概要设计说明书。

2. 实验要求:

用面向数据流的软件设计技术,对上一实验的数据流图和数据字典,进行总体设计;要求做到对上一实验的软件需求进行软件结构设计,模块数不少于四个。

3. 实验内容:

主要解决实现该系统需求的程序模块设计问题。(包括如何把该系统划分成若干个模块、决定各个模块之间的接口、模块之间传递的信息,以及数据结构、模块结构的设计等。)

4. 实验步骤:

1. 首先确定系统总体设计方案(分清系统是事务型还是加工型)。
2. 完成系统的模块结构图及模块的功能说明
3. 完成系统的接口设计
4. 完成系统的结构设计

5. 实验学时: 4 学时。

6. 实验结果: 上交实验报告。



实验四 软件详细设计

1. 实验目的：

- 1) 完成系统的总体设计；
- 2) 完成系统的详细设计工作；
- 3) 完成主要程序（标识符）设计说明；
- 4) 写出详细设计说明书；

2. 实验要求：

确定应该如何具体地实现所要求的系统，从而在编码阶段可以把这个描述直接翻译成用具体的程序语言书写的程序。

3. 实验内容：

进行软件系统的结构设计、逐个模块的程序描述（包括各模块的功能、性能、输入、输出、算法、程序逻辑、接口等等）

4. 实验步骤：

- 1) 首先进行程序系统的结构设计。
- 2) 然后对主要程序进行描述。

注：应该同时进行用户界面设计。

5. 实验学时：4 学时。

6. 实验结果：上交实验报告。



实验五 软件测试和调试

1. 实验目的:

- 1) 通过实例使同学们掌握软件测试的一般步骤。
- 2) 进行黑盒、白盒测试用例设计, 形成测试用例表;
- 3) 进行黑盒测试, 得出测试记录;
- 4) 进行白盒测试, 得出测试记录;
- 5) 写出测试报告。

2. 实验要求:

要求掌握如何设计测试方案、撰写测试说明书, 并掌握程序修改的常用技术。要求对上一实验所编的程序进行测试, 要分步进行, 要有较详细的测试说明书, 要测试通过。

3. 实验内容:

用课堂上介绍的方法, 对上一实验的程序, 进行测试, 并要测试通过, 然后撰写软件测试说明书。

4. 实验步骤:

- 1) 首先设计测试方案 (分别采用白盒法和黑盒法)。
- 2) 进行测试。
- 3) 对于出错的程序进行修改。
- 4) 并不断循环直至程序符合要求。
- 5) 写出测试说明书。

5. 实验学时: 2 学时。

6. 实验结果: 上交实验报告。



实验六 UML 用例图的设计和实现

1. 实验目的：

- 1) 能够根据系统的功能分析系统的用例组成。
- 2) 正确确定用例图中的角色。
- 3) 能够分析系统的用例、活动者以及它们之间的关系。
- 4) 使用 Rose 软件能正确画出用例图，并说明每一个用例的事件流。

2. 实验要求：

根据需求文档确定每一个用例的名称、参与执行者(活动者)、前置条件、主事件流、辅事件流和后置事件流。

3. 实验内容：

学校的网上选课系统的用例图的设计和实现

4. 实验步骤：

- 1) 系统需求分析
- 2) 确定事件流
- 3) 用 Rose 画出用例图

5. 实验学时：2 学时。

6. 实验结果：上交所制作的 UML 用例图文件和实验报告。



实验七 UML 活动图的设计和实现

1. 实验目的：

- 1) 熟悉活动图的创建，根据事件流，准确地勾画出活动图。
- 2) 能够分析系统的每一个用例的事件流、画出相应的活动图。
- 3) 能使用 Rose 软件正确画出活动图。

2. 实验要求：

掌握活动图的基本概念和组成。用活动图来描述事件的流程。描述采取何种动作、做什么(对象状态改变)、何时发生(动作序列)以及在何处发生(泳道)。

确定活动图的起始状态、终止状态、状态转移、决策、守护条件、同步棒和泳道。

3. 实验内容：

通过实际制作选课系统中的 Add Course（添加课程）用例来学习设计和实现活动图。

4. 实验步骤：

- 1) 活动图的基本概念和组成
- 2) 分析用例的事件流
- 3) 用 Rose 软件画出活动图

5. 实验学时：2 学时。

6. 实验结果：上交所制作的 UML 活动图文件和实验报告。



实验八 UML 序列图和协作图的设计和实现

1. 实验目的:

- 1)能够掌握两种交互图的不同点和区别。
- 2) 能够根据事件流, 准确确定对象, 画出序列图和协作图。
- 3) 熟悉使用 Rose 软件创建两种交互图。

2. 实验要求

掌握两种交互图即序列图和协作图的概念和组成, 掌握通过交互图显示对象之间的关系和对象之间处理的消息来建模系统的动态特性。

3. 实验内容:

通过选课用例(select Course)为例, 来学习序列图和协作图的设计和实现。

4. 实验步骤:

- 1) 分析选课事件流;
- 2) 用 Rose 创建 select Course 序列图
- 3) 用 Rose 创建 select Course 协作图

5. 实验学时: 2 学时。

6. 实验结果: 上交所制作的 UML 序列图和协作图文件和实验报告。



实验九 UML 类图的设计和实现

1. 实验目的:

- 1) 根据对系统的分析, 准确确定相应的类, 确定类的属性和操作。并熟练应用 Rose 软件正确定义类及其属性和方法
- 2) 掌握正确定义继承关系并在 Rose 软件中建立继承关系, 正确分析属性和操作的可继承性。
- 3) 掌握如何正确分析类之间的关系, 画出类图的, 正确使用 Rose 软件定义类和类的关系和画出完整的类图。

2. 实验要求:

能从一类单一的对象中抽象出类的定义, 并依据实际要求, 设计类的属性和操作。

能对现实世界中的对象存在各种各样的关系, 分析出这些关系, 主要有: 继承关系; 类之间的关联。并能在类图中反映这些关系。

3. 实验内容:

本实验通过分析选课系统寻找相应的类, 然后设计出它们的类图, 在类图中, 对涉及到的用户包括管理员 (Registrar) 和学生 (Student), 建立统一基类 People, 而 Registrar 和 Student 则由 People 派生。然后着重对选课系统的关联关系进行分析和设计。

4. 实验步骤:

- 1) 确定选课系统中的类
- 2) 进行类图的创建
- 3) 确定类的继承关系
- 4) 分析选课系统的关联关系
- 5) 确定选课系统中的类、属性和操作
- 6) 确定并创建类图的关联关系

5. 实验学时: 4 学时。

6. 实验结果: 上交所制作的 UML 类图文件和实验报告。



实验十 UML 构件图和布署图的设计和实现

1. 实验目的:

- 1) 掌握系统的构件图组成, 并画出选课系统的构件图
- 2) 掌握系统的布署图组成, 并画出该图
- 3) 通过学习布署图的概念, 能正确分析选课系统的布署图的组成和连接关系, 并使用 Rose 正确画出布署图, 定义每一个要素的属性。

2. 实验要求:

通过本实验的学习, 要求能够掌握构件的基本概念、构件图的设计和实现。能够掌握布署图的设计和实现。

3. 实验内容:

本实验通过实际的选课系统的实例来说明构件图和布署图的设计和制作方法。

4. 实验步骤:

- 1) 分析并创建选课系统构件图
- 2) 分析选课系统布署图
- 3) 布署图的创建

5. 实验学时: 2 学时。

6. 实验结果: 上交所制作的 UML 构件图和布署图文件和实验报告。



第三章 实验报告(Reports)

实验一 Microsoft Visio 软件的使用

实验项目名称：Microsoft Visio 软件的使用

实 验 目 的：

- 1) 熟悉 Visio 的工作环境及组成；
- 2) 掌握用 Visio 软件绘制图表的基本操作；
- 3) 能熟练运用 Visio 软件绘制各种较复杂的专业图表；
- 4) 掌握各种图表文档创建方法。

实 验 内 容：使用 Microsoft Visio2002 来设计一个基本流程图模型。

实 验 步 骤：

- 1) 通过打开模板并向图表添加形状来开始创建图表。
- 2) 在图表中移动形状并调整形状的大小。
- 3) 向图表添加文本。
- 4) 连接图表中的形状。
- 5) 设置图表中形状的格式。
- 6) 保存图表以示完成，并向演示图表。

1.Visio 软件的功能与特点



2. Visio 软件绘制流程图的主要操作步骤.

步骤 1:

步骤 2:

步骤 3:

.....

3. 实验小结:



实验二 软件需求分析

实验项目名称： 软件需求分析

实 验 目 的：

- 1) 根据所选定题目进行需求分析工作；
- 2) 通过实例掌握结构化数据流分析技术；
- 3) 进行业务需求分析、用户需求、功能需求、非功能需求分析；
- 4) 写出需求规格说明书（含数据流图）

实 验 内 容： 用结构化数据流分析技术进行软件系统需求分析，得出系统得数据流图和数据字典。

实 验 步 骤：

1. 到相关单位进行需求分析。
2. 综合利用 Internet 网和相关书籍整理并完善需求分析。
3. 画出系统数据流图（分清系统是事务型还是加工型）。
4. 得出系统数据字典。

1. 软件系统需求描述：（从功能，性能上进行描述）



2. 软件系统数据流图（由加工、数据流、文件、源点和终点四种元素组成）：

1) 顶层数据流图

2) 1 层数据流图



3) 2 层数据流图

3. 软件系统数据字典:

1. 数据流条目



2. 加工条目

3. 文件条目

4. 实验小结:



实验三 软件概要设计

实验项目名称： 软件概要设计

实 验 目 的：

- 1) 建立初始结构图；
- 2) 对初始结构图进行改进、优化；
- 3) 写出概要设计说明书。

实 验 内 容：

主要解决实现该系统需求的程序模块设计问题。(包括如何把该系统划分成若干个模块、决定各个模块之间的接口、模块之间传递的信息，以及数据结构、模块结构的设计等。)

实 验 步 骤：

- 1) 首先确定系统总体设计方案（分清系统是事务型还是加工型）。
- 2) 完成系统的模块结构图及模块的功能说明
- 3) 完成系统的接口设计
- 4) 完成系统的数据结构设计

1. 软件系统模块结构图：



2. 接口设计

3. 数据结构设计



4. 出错处理设计

5. 实验小结:



实验四 软件详细设计

实验项目名称：软件详细设计

实 验 目 的：

- 1) 完成系统的总体设计；
- 2) 完成系统的详细设计工作；
- 3) 完成主要程序（标识符）设计说明；
- 4) 写出详细设计说明书；

实 验 内 容：进行软件系统的结构设计、逐个模块的程序描述（包括各模块的功能、性能、输入、输出、算法、程序逻辑、接口等等）

实 验 步 骤：

- 1) 首先进行程序系统的结构设计。
- 2) 然后对主要程序进行描述。

注：应该同时进行用户界面设计。

1. 软件系统其中的 3 个模块的详细设计（画程序流程图）：



2. 程序描述:



3. 实验小结:



实验五 软件测试和调试

实验项目名称：软件测试和调试

实 验 目 的：

- 1)通过实例使同学们掌握软件测试的一般步骤。
- 2)进行黑盒、白盒测试用例设计，形成测试用例表；
- 3)进行黑盒测试，得出测试记录；
- 4)进行白盒测试，得出测试记录；
- 5)写出测试报告。

实 验 内 容：

用课堂上介绍的方法，对上一实验的程序，进行测试，并要测试通过，然后撰写软件测试说明书。

实 验 步 骤：

1. 首先设计测试方案（分别采用白盒法和黑盒法）。
2. 进行测试。
3. 对于出错的程序进行修改。
4. 并不断循环直至程序符合要求。
5. 写出测试说明书。

1. 测试项目



2. 设计测试用例 15 个（白盒法 10 个，黑盒法 5 个）：

3.对所开发软件测试结果的评介：

4.. 实验小结：



实验六 UML 用例图的设计和制作

实验项目名称：UML 用例图的设计和制作

实 验 目 的：

- 1) 能够根据系统的功能分析系统的用例组成。
- 2) 正确确定用例图中的角色。
- 3) 能够分析系统的用例、活动者以及它们之间的关系。
- 4) 使用 Rose 正确画出用例图，并说明每一个用例的事件流。

实 验 内 容：

学校的网上选课系统的用例图的设计和实现

实 验 步 骤：

- 1) 系统需求分析
- 2) 确定事件流
- 3) 用 Rose 画出用例图

实 验 学 时：2 学时

1. 网上选课系统需求分析



2. 主要事件流

3. Rose2003 软件设计选课系统的用例图主要操作步骤:

步骤 1:

步骤 2:

步骤 3:

步骤 4:

.....

4. 实验小结:



实验七 UML 活动图的设计和实现

实验项目名称：UML 活动图的设计和实现

实 验 目 的：

- 1) 熟悉活动图的创建，根据事件流，准确地勾画出活动图。
- 2) 能够分析系统的每一个用例的事件流、画出相应的活动图。
- 3) 能使用 Rose 软件正确画出活动图。

实验内容：

通过实际制作选课系统中的 Add Course（添加课程）用例来学习设计和实现活动图。

实验步骤：

- 1) 活动图的基本概念和组成
- 2) 分析用例的事件流
- 3) 用 Rose 画出活动图

1.活动图的基本概念和组成

2.添加课程用例的事件流：



3.创建添加课程用例的活动图步骤:

4. 实验小结:



实验八 UML 序列图和协作图的设计和实现

实验项目名称：UML 序列图和协作图的设计和实现

实验目的：

- 1) 能够掌握两种交互图的相同点和区别。
- 2) 能够根据事件流，准确确定对象，画出序列图和协作图。
- 3) 熟悉使用 Rose 软件创建两种交互图。

实验内容：

通过选课用例(select Course)为例，来学习序列图和协作图的设计和实现。

实验步骤：

- 1) 分析选课事件流；
- 2) 用 Rose 创建 select Course 序列图
- 3) 用 Rose 创建 select Course 协作图

1.序列图和协作图的基本概念和组成

2.分析选课事件流：



3.用 Rose 创建选课系统序列图的操作步骤:

4.用 Rose 创建选课系统协作图的操作步骤:

5. 实验小结:



实验九 UML 类图的设计和实现

实验项目名称：UML 类图的设计和实现

实验目的：

- 1) 根据对系统的分析，准确确定相应的类，确定类的属性和操作。并熟练应用 Rose 正确定义类及其属性和方法
- 2) 掌握正确定义继承关系并在 Rose 中建立继承关系，正确分析属性和操作的可继承性。
- 3) 掌握如何正确分析类之间的关系，画出类图的，正确使用 Rose 定义类和类的关系和画出完整的类图。

实验内容：

本实验通过分析选课系统寻找相应的类，然后设计出它们的类图，在类图中，对涉及到的用户包括管理员 (Registrar) 和学生 (Student)，建立统一基类 People，而 Registrar 和 Student 则由 People 派生。然后着重对选课系统的关联关系进行分析和设计。

实验步骤：

- 1) 确定选课系统中的类
- 2) 进行类图的创建
- 3) 确定类的继承关系
- 4) 分析选课系统的关联关系
- 5) 确定选课系统中的类、属性和操作
- 6) 确定并创建类图的关联关系

1.分析选课系统中的主要类：



2.用 Rose 创建类图的基本步骤:

3.用 Rose 创建类的继承的基本步骤:

4.选课系统的主要关联关系分析:



5.选课系统中主要类的属性和操作:

6.用 Rose 创建类的关联的基本操作步骤:

5. 实验小结:



实验十 UML 构件图和布署图的设计和实现

实验项目名称：UML 构件图和布署图的设计和实现

实验目的：

- (1) 掌握系统的构件图组成，并画出选课系统的构件图
- (2) 掌握系统的布署图组成，并画出该图
- (3) 通过学习布署图的概念，能正确分析选课系统的布署图的组成和连接关系，并使用 Rose 正确画出布署图，定义每一个要素的属性。

实验内容：

本实验通过实际的选课系统的实例来说明构件图和布署图的设计和制作方法。

实验步骤：

- 1) 分析并创建选课系统构件图
- 2) 分析选课系统布署图
- 3) 布署图的创建

1.构件图和布署图的基本概念：

2.分析选课系统主要类：



3.用 Rose 创建选课系统构件图的基本操作步骤:

4.用 Rose 创建选课系统部署图的基本操作步骤:

5. 实验小结:



附录 1 如何实现数据库操作

一般来说，信息管理系统的开发需要两方面的知识：可视化编程语言；数据库管理系统。如何实现前台开发工具与后台数据库的连接和操作是信息管理系统开发的关键。

从往届的同学来看，一般对开发一个软件系统非常感兴趣，感觉自己不再只会读书，还能作点实际的东西，但往往开始时信心百倍，遇到困难时就放弃。

信息系统的开发到底有多难，其实不难。关键要理解和熟练，然后才会灵活运用。

一、如何链接数据库

无论那种编程语言，链接数据库需要三个类：

1. 链接类（connection） 负责连上数据库，它有一个属性：连接字符串。连接字符串指出服务器名称、数据库名称，连接机制等内容。

2. 命令类（command）

连上数据库后，需要发出操作命令：查、插入、更新、删除。这时需要用到 SQL 语言的：select; insert; update; delete。

3. 记录集（recordset）

对数据库的更改往往不是直接进行，首先对中间缓冲区 recordset 更改，然后将这种修改提交到数据库。

二、数据库链接的途径

1. 完全由控件实现，如：ADO 控件，初始编程者往往用这种方式。

2. 写代码来实现，这种方式对数据库操作达到更大的灵活度。

三、用 VB6.0 来举例

1. 首先在模块中定义两个全局变量：连接类和连接字符串。

 '定义两个全局变量

 Global Gcon As ADODB.Connection '定义连接类

 Global Gstr As String '定义连接字符串



2. 在登录窗体的窗体装载事件中

Private Sub Form_Load()

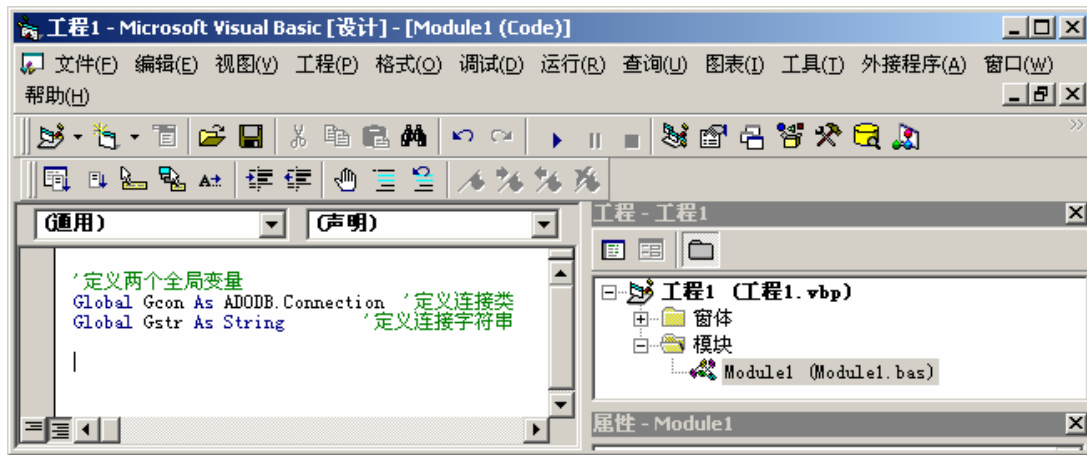
```
Gstr = "Provider=SQLOLEDB.1;Integrated Security=SSPI;Persist Security
Info=False;User ID=sa;Initial Catalog=物业;Data Source=CZX-57FE7C18F5D"
```

```
Set Gcon = New ADODB.Connection
```

```
Gcon.Open Gstr
```

```
Set rec = New ADODB.Recordset
```

End Sub



3. 下面是删除和保存功能的代码

Public Sub SaveAllTexts()

```
On Error GoTo errnumber
```

```
Gcon.BeginTrans
```

将界面的文本框等控件的内容保存到数据表中

```
Sql = "delete from userfees where usercode=" & Txtusercode.Text & " "
```

```
Gcon.Execute Sql 完成删除
```

```
Sql = "select * from userfees"
```

```
rec.Open Sql, Gcon, adOpenDynamic, adLockOptimistic, adCmdText
```

```
rec.AddNew
```



```
rec.Fields("usercode").Value = Txtusercode.Text
rec.Fields("username").Value = Txtusername.Text
rec.Update          完成保存
rec.Close
Gcon.CommitTrans
Exit Sub
errnumber:
If Err.Number = -2147217865 Then
    MsgBox ("该用户已存在不能保存!")
    Exit Sub
End If
Gcon.RollbackTrans
End Sub
```



附录 2 设计范例 1

设计题目：开发基于 Web 的高校教学资料管理系统

一、需求描述

利用 C#.NET 作为开发工具、SQL Server2000 作为数据库开发一个基于 WEB 的高校教学资料网络管理系统。要求实现如下功能：

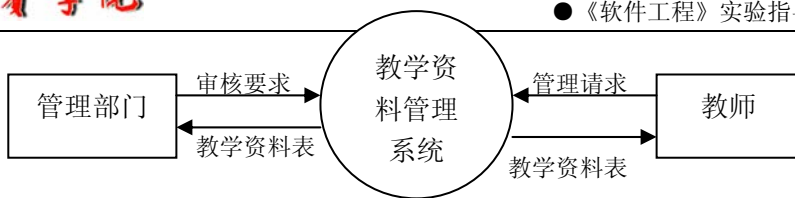
1. 教学资料录入；
2. 教学资料修改；
3. 教学资料审核；
4. 教学资料打印；

随着信息时代的到来，高校的信息化建设显得愈来愈重要。本系统是从安徽工业大学的实际出发，针对学校的教学资料管理的现状，经过详细的系统调查，为学校“量体裁衣”开发出的一个基于 Web 的教学资料管理系统。其目的是改善高校工作运行效率和效果，提高高校资料的管理水平和整体素质，加快高校的信息化建设。

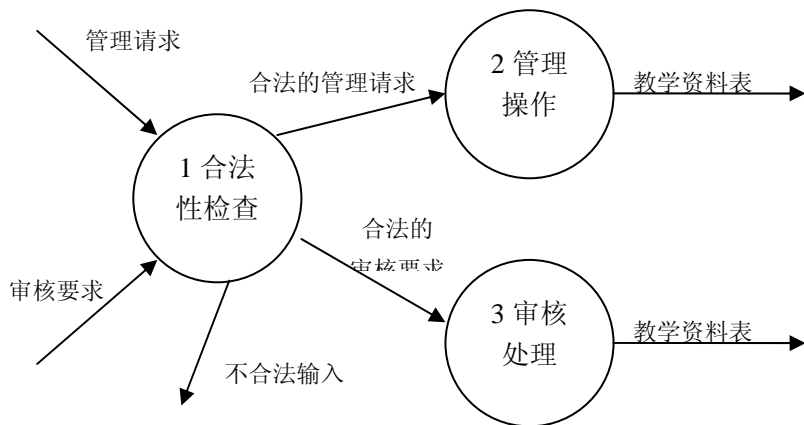
本管理系统是采用当今流行的 C#开发而成。作为一种面向对象的编程语言,它对系统的设计起到了很大的帮助。另外作为开发环境的 Asp.net 是一种建立在通用语言上的程序框架,能够只用一台 Web 服务器来建立强大的 Web 应用程序。后台数据库使用的是 SQL Server 2000,在此数据库中进行相关的操作直观、简单，为本系统的开发提供了很大的方便。

教学资料管理系统包括对教学日历，课程教学小结，教学时间分配，试卷分析以及毕业论文送审表的录入、修改、删除和审核。

二、数据流图（DFD）



顶层图



1层数据流图

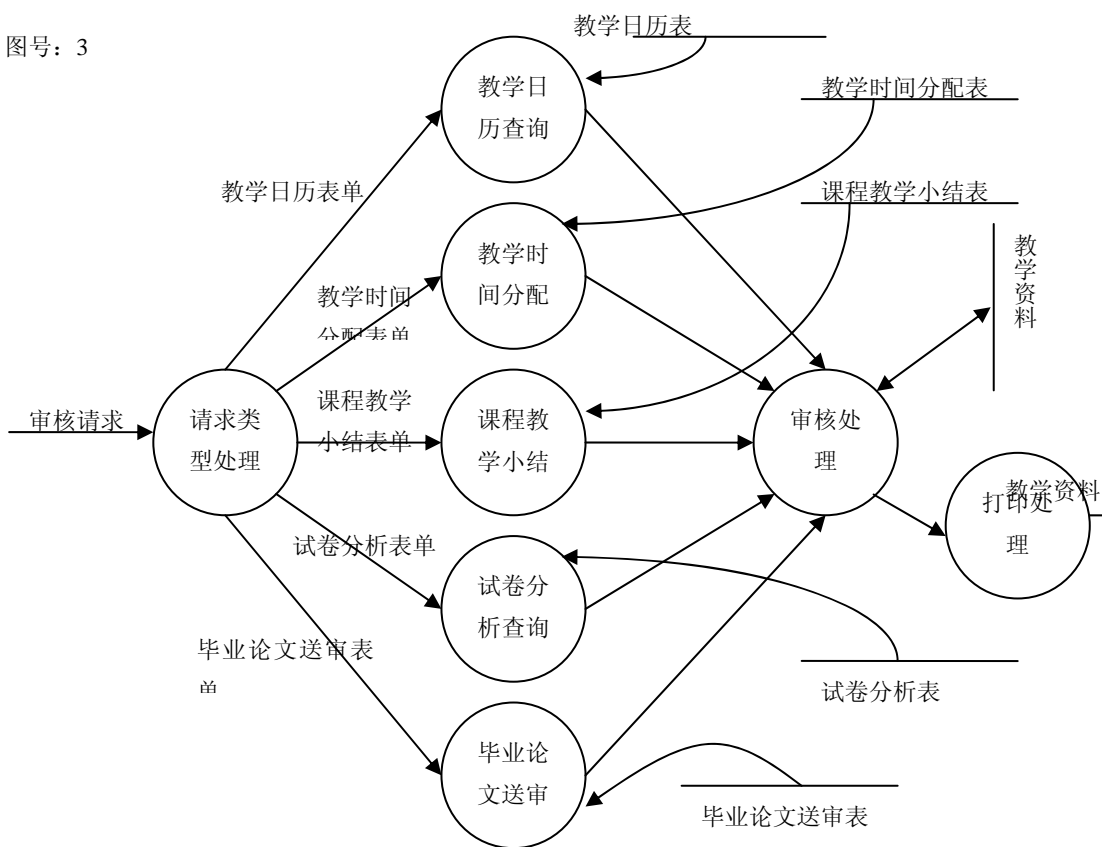
图号：2

```
graph LR; MP[管理请求] --> PTH((请求类型处理)); PTH -- "教学日历表单" --> TLM((教学日历管理)); PTH -- "教学时间分配表单" --> TAT((教学时间分配)); PTH -- "课程教学小结表单" --> CTX((课程教学小结)); PTH -- "试卷分析表单" --> TAM((试卷分析管理)); PTH -- "毕业论文送审表" --> BLSA((毕业论文送审)); TLM --> TL[教学日历表]; TLM --> THA((打印处理)); TAT --> THA; TAT --> TATB[教学时间分配表]; TAT --> CTS(课程教学小结表); CTX --> THA; CTX --> CTS; TAM --> THA; TAM --> TA[试卷分析表]; BLSA --> THA; BLSA --> BLAS[毕业论文送审表]; THA --> TZL[教学资料表];
```

The diagram illustrates the workflow for handling management requests. It begins with a 'Management Request' (管理请求) input to the 'Request Type Processing' (请求类型处理) node. This central node branches into five parallel processing paths: Teaching Calendar Management (教学日历管理), Teaching Time Allocation (教学时间分配), Course Teaching Summary (课程教学小结), Exam Paper Analysis Management (试卷分析管理), and Graduate Thesis Submission Review (毕业论文送审). Each path involves a specific form being submitted to its respective management node. The management nodes then either generate a final output table (e.g., Teaching Calendar Table, Teaching Time Allocation Table, Course Teaching Summary Table, Exam Paper Analysis Table, Graduate Thesis Submission Review Table) or feed into a common 'Printing Processing' (打印处理) node. The 'Printing Processing' node produces the final 'Teaching Materials Table' (教学资料表).



图号：3

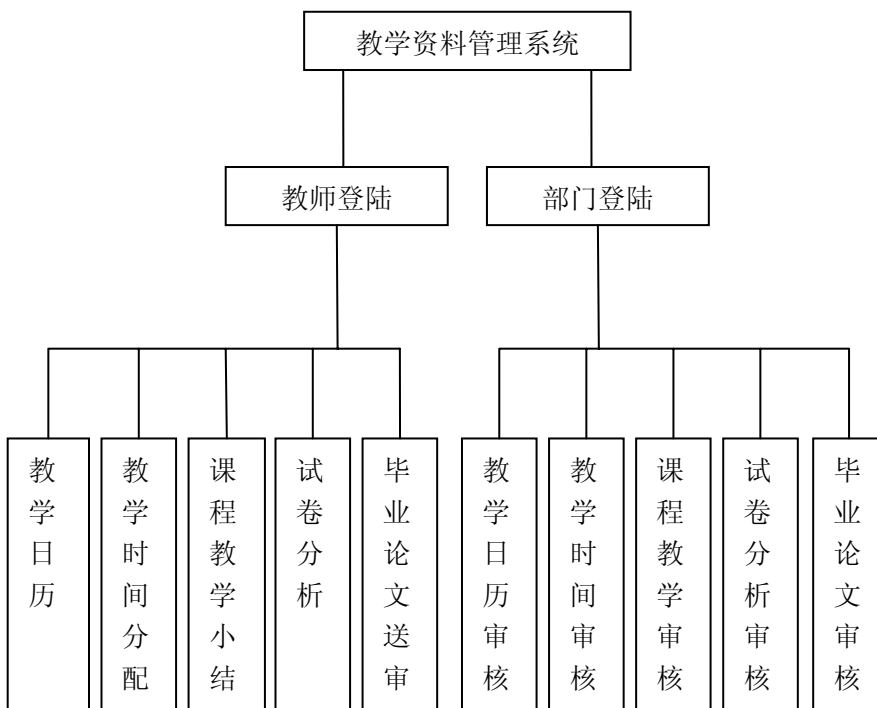


2 层数据流图

三、设计（包括总体设计和详细设计）



1、系统模块结构图和界面



欢迎进入 Totti老师，您好！

旧密码错误！

登陆信息

旧密码 *	●●●●●●	
新密码 *		
确认密码 *		

个人信息

姓 名 *	Totti	
性 别 *	男	
院 系 *	机械学院	
职 称	讲师	

联系方式

联系电话 *	2301609	
电子邮件	tottiwei@21cn.com	
联系地址		

教学日历

教学时间分配

课程教学小结

试卷分析

毕业论文送审

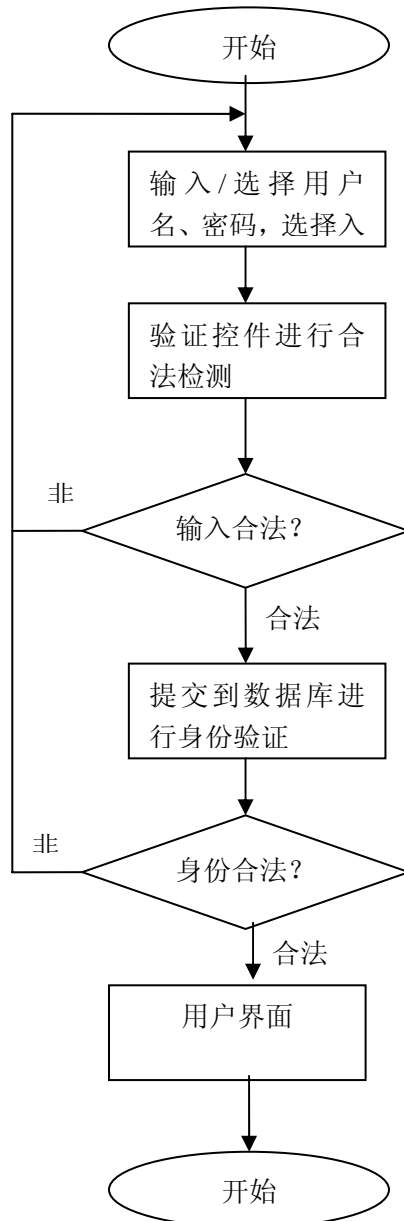
个人信息修改

退出登陆



详细设计

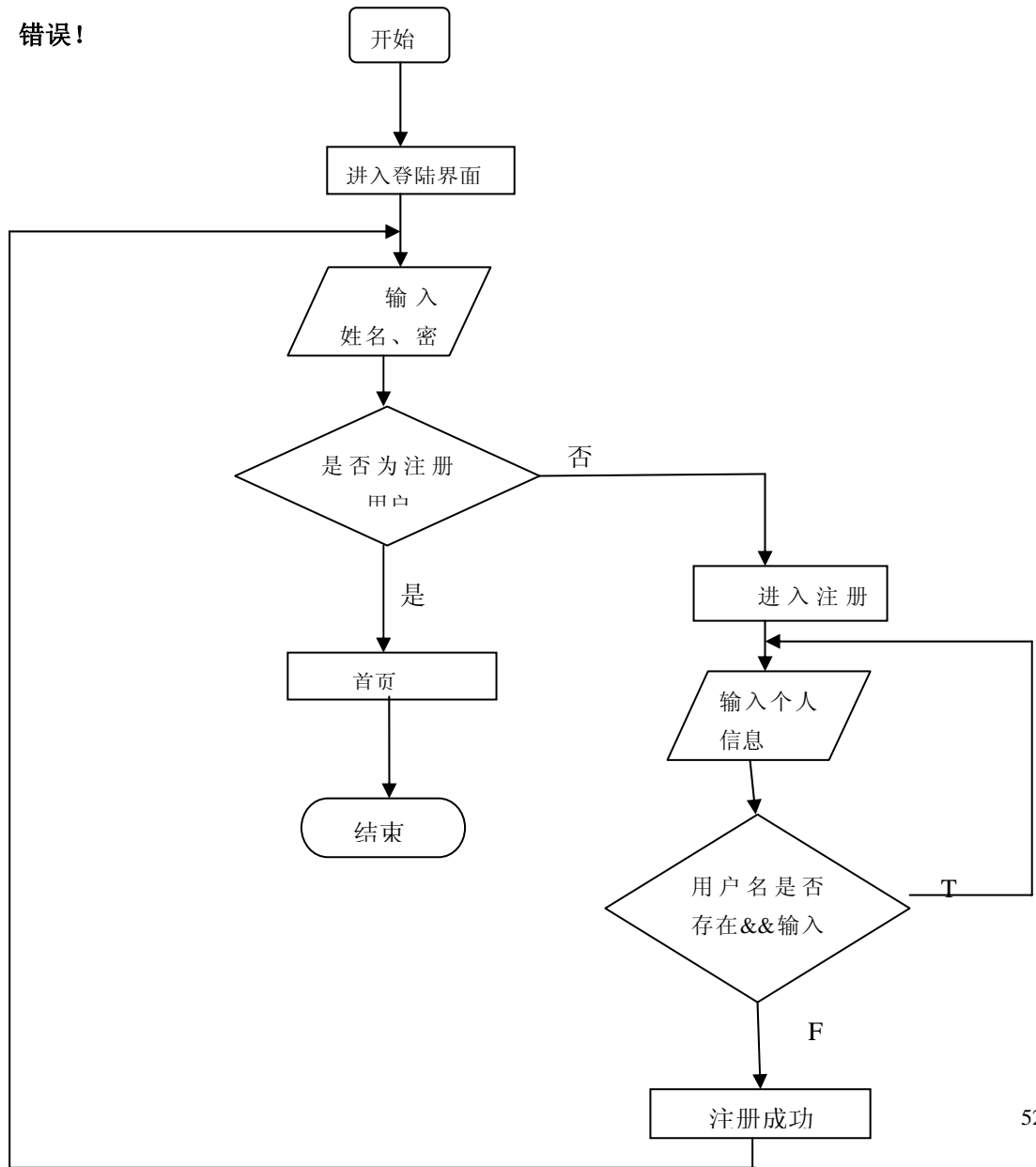
1) 登陆模块





2) 用户注册模块

错误!



四、编码

编码就是将详细设计的程序流程图翻译成某种语言所写成的代码，同学们可以选择自己熟悉的编程语言来进行，VB6.0, Dephi6.0, PB7.0, C++Builder, VC++, C#, VB.Net 等都可以。

五、测试和调试



现在测试**登录**模块，只有输入正确的用户名和密码，才能打开下一个窗体，如果用户名或密码有一个不对或两个都不对，则不能打开下一个窗体，提示用户不是一个合法用户。

假设正确的用户名和密码是：Bajao; Totti，因此可设计一组测试用例：

- | | |
|----------------|------------|
| 1) 用户名<> Bajao | 密码<> Totti |
| 2) 用户名<> Bajao | 密码= Totti |



3) 用户名= Bajao 密码<> Totti

4) 用户名= Bajao 密码= Totti

这一组测试可以检测出**登录**模块的编码是否满足功能要求。

调试就是依据测试发现的问题而修改程序代码，直到测试不再发现问题。



附录 3 设计范例 2

设计题目：机票预定系统

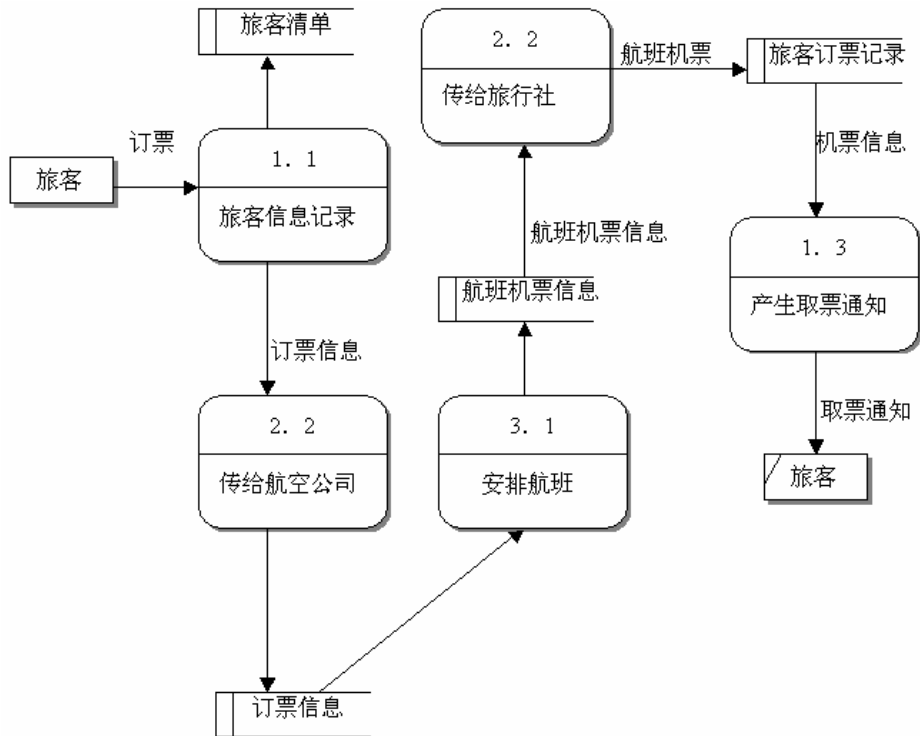
机票预定系统需求分析

机票预定系统的功能要求

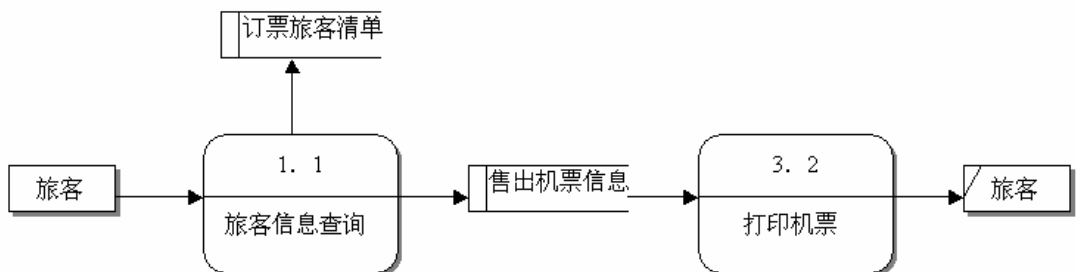
机票预定系统的总目标是：在计算机网络，数据库和先进的开发平台上，利用现有的软件，配置一定的硬件，开发一个具有开放体系结构的、易扩充的、易维护的、具有良好人机交互界面的机票预定系统，实现航空公司的机票销售的自动化的计算机系统，为企业的决策层提供准确、精细、迅速的机票销售信息。

根据可行性研究的结果和客户的要求，分析现有情况及问题，采用 **Client/Server** 结构，将机票预定系统划分为两个子系统：客户端子系统，服务器端子系统。

旅客订票数据流图：



旅客取票数据流图：





下面分析各个子系统的功能需求：

1. 客户端子系统：

在客户端系统的功能实现上，可以分为以下几个部分：

[1]旅客信息的输入和统计

旅行社把旅客要求订票的信息由专人负责输入。这部分功能是客户端子系统的基本部分，这个功能是以后各个部分的基础。系统要求做到即能够从其它子系统中共享一部分信息，又有方便的操作界面手工输入旅客信息。这部分要求对输入的数据进行简单的统计，供航空公司进行查询和宏观调控。

[2]旅客信息的存储：

将旅客的信息存储到旅行社的客户端系统中，以备以后的取票确认以及查询。

[3]机票信息的传递及接收：

将旅客所须的机票信息由旅行社客户端由网络传到航空公司的服务器上，并且接受航空公司返回的航班信息，然后存储起来。

[4]取票通知及帐单的生成和打印：

把已存储的从航空公司返回的航班机票信息打印出来，并且生成帐单打印出来一起交给旅客。

印出机票给已经订票的旅客：根据旅客的取票通知及帐单，经过确认无误后，接受旅客的付款后把机票印出来交给旅客。

[5]机票销售情况的核算



这一功能是在上一功能的基础上，对机票销售额进行单项核算，得到该旅行社的销售情况并把核算结果作为企业报表输出。

2. 服务器端的功能要求:

通过计算机网络将客户端与服务器的数据库相连，将从客户端得到的信息进行处理，实现航班查询，机票生成，销售统计，综合信息查询等子系统。以计算机成本核算为中心，实现销售业务的计算机自动化，为航空公司降低成本、提高销售额、经营决策提供及时精确的依据。

在客户端系统的功能实现上，可以分为以下几个部分：

[1]接收由旅行社客户端发回的所需机票信息：

通过网络接收机票信息并存入到服务器的数据库中。

[2]生成航班信息：

根据所需机票信息（时间，地点），在数据库中查询并得到正确的航班的信息（价格，时间，等级），分配所需的机票数并在数据库中做出已售出的标记。

[3]传递航班信息到客户端（旅行社）：

把得到的航班信息通过网络传递到旅行社。

[4]接收旅行社的反馈信息：

对旅行社的反馈信息进行分析，把已经售出的机票进行统计，对被旅客所退掉的机票要进行数据库的恢复。

[5]印出机票给已经订票的旅客：

根据旅客的取票通知及帐单，经过确认无误后，接受旅客的付款后把机票印出来交给旅客。

[6]销售额的分析和管理的

这一步骤的功能要求包括对销售的机票进行分析，这一工作是在前面的基础上，以计算机为工具，对机票预定系统的功能和目标进行扩充。它以财务管理学为理论基础，以辅助决策为目标，以机票销售数据为中心，广泛采用统计学、运筹学的分



析方法，对销售信息进行深层加工，建立反映不同航班需求的模型，提供管理上所需的各种辅助决策信息和财务信息。这一要求是机票预定系统的最高目标，将通过系统运行后获得的大量销售历史数据基础上，实现这一目标。

机票预定系统的性能需求

为了保证系统能够长期、安全、稳定、可靠、高效的运行，机票预定系统应该满足以下的性能需求：

1. 系统处理的准确性和及时性

系统处理的准确性和及时性是系统的必要性能。在系统设计和开发过程中，要充分考虑系统当前和将来可能承受的工作量，使系统的处理能力和响应时间能够满足企业对信息处理的需求。

由于机票预定系统的查询功能对于整个系统的功能和性能完成举足轻重。作为系统的很多数据来源，而机票数量和时间又影响企业的决策活动，其准确性很大程度上决定了机票预定系统的成败。在系统开发过程中，必须采用一定的方法保证系统的准确性。

2. 系统的开放性和系统的可扩充性

机票预定系统在开发过程中，应该充分考虑以后的可扩充性。例如订票系统的方式的改变（网上订票），用户查询的需求也会不断的更新和完善。所有这些，都要求系统提供足够的手段进行功能的调整和扩充。而要实现这一点，应通过系统的开放性来完成，既系统应是一个开放系统，只要符合一定的规范，可以简单的加入和减少系统的模块，配置系统的硬件。通过软件的修补、替换完成系统的升级和更新换代。



3. 系统的易用性和易维护性

机票预定系统是直接面对使用人员的，而使用人员往往对计算机并不时非常熟悉。这就要求系统能够提供良好的用户接口，易用的人机交互界面。

要实现这一点，就要求系统应该尽量使用用户熟悉的术语和中文信息的界面；针对用户可能出现的使用问题，要提供足够的在线帮助，缩短用户对系统熟悉的过程。

机票预定系统中涉及到的数据是航空公司的相当重要的信息，系统要提供方便的手段供系统维护人员进行数据的备份，日常的安全管理，系统意外崩溃时数据的恢复等工作。

4. 系统的标准性

系统在设计开发使用过程中都要涉及到很多计算机硬件、软件。所有这些都符合主流国际、国家和行业标准。例如在开发中使用的操作系统、网络系统、开发工具都必须符合通用标准。如规范的数据库操纵界面、作为业界标准的 TCP/IP 网络协议及 ISO9002 标准所要求的质量规范等；同时，在自主开发本系统时，要进行良好的设计工作，制订行之有效的软件工程规范，保证代码的易读性、可操作性和可移植性。

5. 系统的先进性

目前计算系统的技术发展相当快，做为机票预定系统工程，应该保证系统在下个世纪仍旧是先进的，在系统的生命周期尽量做到系统的先进，充分完成企业信息处理的要求而不至于落后。这一方面通过系统的开放性和可



扩充性，不断改善系统的功能完成。另一方面，在系统设计和开发的过程中，应在考虑成本的基础上尽量采用当前主流并先进且有良好发展前途的产品。

6. 系统的响应速度

机票预定系统在日常处理中的响应速度为秒级，达到实时要求，以及时

反馈信息。在进行统计分析时，根据所需数据量的不同而从秒级到分钟级，原则是保证操作人员不会因为速度问题而影响工作效率。

机票预定系统的数据需求

机票预定系统的数据需求包括如下几点：

1. 数据录入和处理的准确性和实时性

数据的输入是否准确是数据处理的前提，错误的输入会导致系统输出的不正确和不可用，从而使系统的工作失去意义。数据的输入来源是手工输入。手工输入要通过系统界面上的安排系统具有容错性，并且对操作人员要进行系统的培训。

在系统中，数据的输入往往是大量的，因此系统要有一定的处理能力，以保证迅速的处理数据。

2. 数据的一致性与完整性

由于系统的数据是共享的，在不同的旅行社中，机票是共享数据，所以如何保证这些数据的一致性，是系统必须解决的问题。要解决这一问题，要有一定的人员维护数据的一致性，在数据录入处控制数据的去向，并且要求对数据库的数据完整性进行严格的约束。



对于输入的数据，要为其定义完整性规则，如果不能符合完整性约束，系统应该拒绝该数据。

3. 数据的共享与独立性

整个机票预定系统的数据是共享的。然而，从系统开发的角度上看，共享会给设计和调试带来困难。因此，应该提供灵活的配置，使各个分系统能够独立运行，而通过人工干预的手段进行系统数据的交换。这样，也能提供系统的强壮性。

机票预定系统的数据字典

系统中的数据元素：

名字：旅客信息
别名：
描述：旅客的个人信息，用于对旅客的确认
定义：旅客信息=姓名+性别+工作单位+身份证号码+旅行时间
+旅行目的地
位置：输入到旅行社端（client 端）

名字：订票旅客清单
别名：
描述：已订票的旅客的记录
定义：订票旅客清单=旅客信息的合集
位置：输入到旅行社端（Client 端）

名字：订票信息
别名：
描述：旅客的旅行时间和目的地，用于确定旅客的航班
定义：订票信息=旅客旅行时间+旅客旅行目的地
位置：传输到航空公司端（Server 端）



名字：航班机票信息

别名：

描述：旅客的航班信息，根据旅客的旅行时间和目的地确定

定义：航班机票信息=旅客机票时间+旅客机票班次

位置：传输到旅行社端（Client 端）

名字：取票通知

别名：

描述：旅客领取机票的凭证

定义：取票通知=旅客姓名+领票时间

位置：输出到打印机

名字：售出机票信息

别名：

描述：旅客的航班机票的信息

定义：售出机票信息=旅客的航班机票

位置：输出到打印机



名字：旅客订票记录

别名：

描述：已订票的旅客在航空公司的记录

定义：旅客订票记录=旅客姓名+航班机票信息

位置：输出到航空公司端（Server 端）

机票预定系统的运行要求

机票预定系统中的各个子系统的硬件和软件的配置如下：

1. 服务器端子系统的运行要求：

系统软件： Window NT Server

数据库管理系统： SQL Server

硬件要求： Pentium III 450 以上, 258M RAM, 14G HD

2. 客户端子系统的运行要求：

系统软件： Window NT Workstation

数据库管理系统： SQL Server

硬件要求： Pentium 133 以上, 32M RAM, 4.3G HD

建立机票预定系统的约束

1. Client/Server 结构总体设计方案对它的约束：

机票预定系统做为 **Client/Server** 结构的一个应用系统，不可避免的要受到 **Client/Server** 结构的约束。在其实施的各个阶段都要服从它的一些规划，包括功能设计、系统配置和计划。同时，由于信息的共享，机票预定系统还受到其它系统的信息约束。



2. 人力、资金、时间的约束

机票预定工程实施的目标就是要带给航空公司看得出见的效益，其开发过程中也要考虑到人力、资金和时间的约束。因此，在设计中，重点是销售系统中的方便快捷，能提供给旅客以优质高效的服务，并提高销售的效率和便捷，为航空公司带来良好的效益。

3. 技术发展规律的约束

计算机技术和产品的发展日新月异，将会给信息处理带来更多的手段，同时也会带来更加丰富的信息表达形式。例如图象和语音技术的进步，多媒体技术的发展，这些都要求系统在设计时考虑技术变化的可能性，为可能的变化预留一定的系统处理能力。

概要设计说明书

1.编写目的

在本机票预定系统项目的前一阶段，也就是需求分析阶段中，已经将系统用户对本系统的需求做了详细的阐述，这些用户需求已经在上一阶段中对航空公司、各旅行社及机场的实地调研中获得，并在需求规格说明书中得到详尽得叙述及阐明。

本阶段已在系统的需求分析的基础上，对机票预定系统做概要设计。主要解决了实现该系统需求的程序模块设计问题。包括如何把该系统划分成若干个模块、决定各个模块之间的接口、模块之间传递的信息，以及数据结构、模块结构的设计等。在以下的概要设计报告中将对在本阶段中对系统所做的所有概要设计进行详细的说明。

在下一阶段的详细设计中，程序设计员可参考此概要设计报告，在概要设计对机票预定系统所做的模块结构设计的基础上，对系统进行详细设计。在以后的软件测试以及软件维护阶段也可参考此说明书，以便于了解在概要设计过程中所完成的各模块设计结构，或在修改时找出在本阶段设计的不足或错误。

2.目标



2. 2 运行环境

系统将由两部分程序组成，安装在各旅行社客户机上的客户程序及航空公司内的数据服务器程序。

根据调研得知所有旅行社的计算机配置均在 Pentium 133 级别以上，客户程序应能够在 Pentium 133 级别以上, Win NT 环境下运行。

2. 3 需求概述

XX 航空公司为方便旅客，需开发一个机票预定系统。为便于旅客由旅行社代替航空公司负责为旅客定票，旅行社把预定机票的旅客信息，包括姓名、性别、工作单位、身份证号码、旅行时间、旅行目的地，输入机票预定系统的客户端程序，系统经过查询航空公司内的航班数据服务器后，为旅客安排航班，印出取票通知。旅客在飞机起飞前一天凭取票通知和帐单交款后取票，系统校对无误后即印出机票给旅客。

要求系统能有效、快速、安全、可靠和无误的完成上述操作。并要求客户机的界面要简单明了，易于操作，服务器程序利于维护。

3. 总体设计

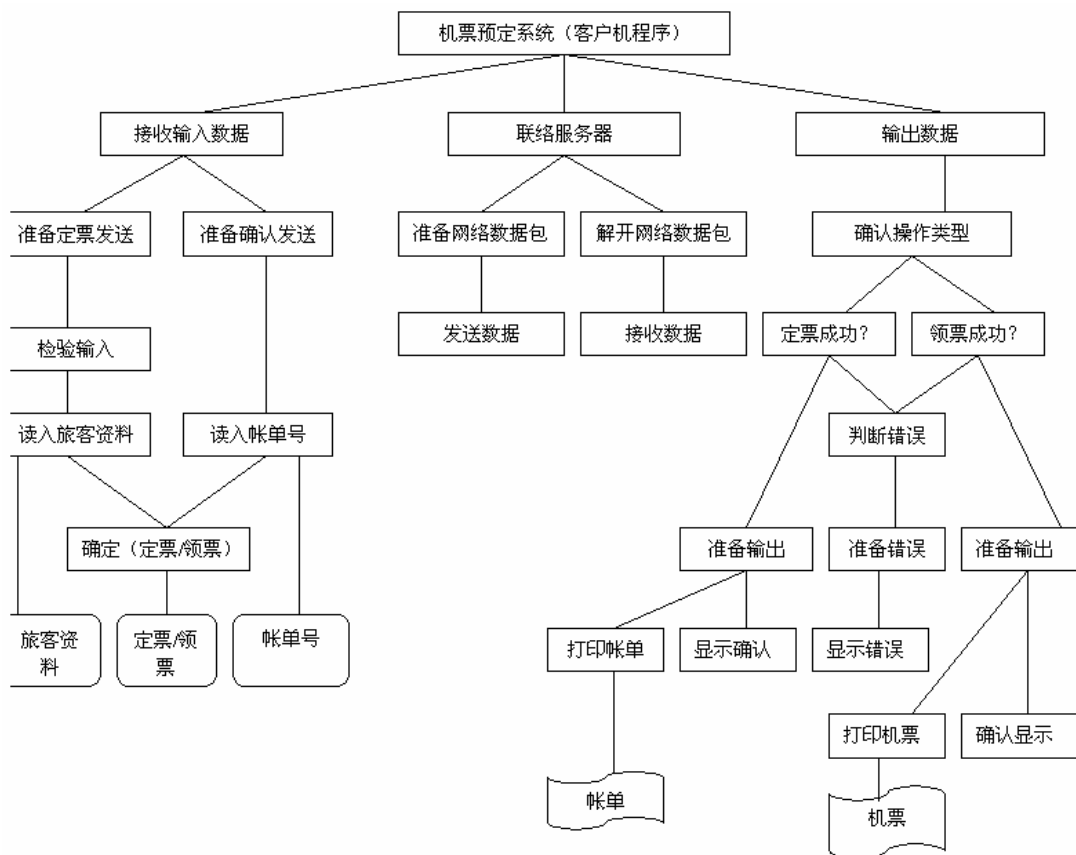
3. 1 处理流程

下面将使用（结构化设计）面向数据流的方法对机票预定系统的处理流程进行分析。系统可分为两大部分：一、客户机上的程序，二、服务器上的程序。以下将分别对系统的这两大部分进行流程分析：

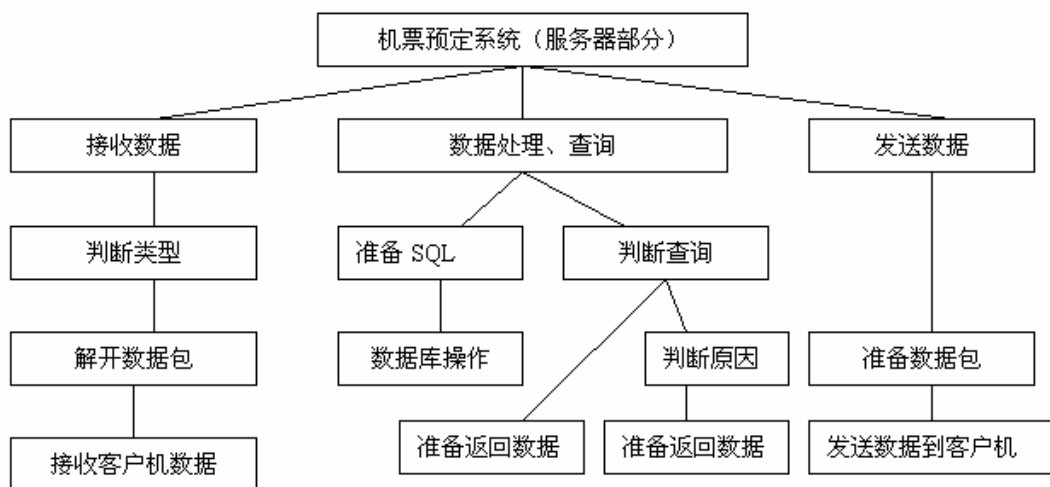
3. 2 总体结构和模块外部设计

下面以结构图来描述机票预定系统的软件总体结构。框内注明了模块的名字；方框之间的直线表示模块的调用关系。

3. 2. 1 客户机部分



3. 2. 2 服务器程序部分



其中在数据处理、查询下作了省略...只给出了一部分模块，实际上是有两部分，分别对应于定位及确认。

3.3 功能分配

各项模块的功能可参照 3.1 中的说明。客户机程序主要有三大块：接收数据、网络通信及输出部分。服务器程序主要也是由三大功能：接收网络数据、数据库操作及发送网络数据部分。服务器程序需与已建立的 SQL SERVER 数据库互连，其接口将于下面部分阐述。

4. 接口设计

4.1 外部接口

4.1.1 用户界面

在用户界面部分，根据需求分析的结果，用户需要一个用户友善界面。在界面设计上，应做到简单明了，易于操作，并且要注意到界面的布局，应突出的显示重要以及出错信息。外观上也要做到合理化，考虑到用户多对 WINDOW 风格较熟悉，应尽量向这一方向靠拢。在设计语言上，已决定使用 MS VISUAL C++ 进行编程，在界面上可使用 VISUAL C++ 所提供的可视化组件，向 WINDOWS 风格靠近。其中服务器程序界面要



做到操作简单，易于管理。在设计上采用下拉式菜单方式，在出错显示上可调用 VISUAL C++ 库中的错误提示函数。

总的来说，系统的用户界面应作到可靠性、简单性、易学习和使用

4. 1. 2 软件接口

服务器程序可使用 VISUAL C++ 提供的对 SQL SERVER 的接口，进行对数据库的所有访问。

服务器程序上可使用 SQL SERVER 的对数据库的备份命令，以做到对数据的保存。

在网络软件接口方面，使用一种无差错的传输协议，采用滑动窗口方式对数据进行网络传输及接收。

4. 1. 3 硬件接口

在输入方面，对于键盘、鼠标的输入，可用 VISUAL C++的标准输入/输出，对输入进行处理。

在输出方面，打印机的连接及使用，也可用 VISUAL C++的标准输入/输出对其进行处理。在网络传输部分，在网络硬件部分，为了实现高速传输，将使用高速 ATM。

4. 2 内部接口

内部接口方面，各模块之间采用函数调用、参数传递、返回值的方式进行信息传递。具体参数的结构将在下面数据结构设计的内容中说明。接口传递的信息将是数据结构封装了的数据，以参数传递或返回值的形式在各模块间传输。

5. 数据结构设计

5. 1 数据库数据结构设计

DBMS 的使用上系统将采用 SQL SERVER，系统主要需要维护 3 张数据表：

1. 航班班次表

	航班号	起飞日期/时间	飞行目的地	座位空数	商务仓票价	经济仓票价



航班班次表保存了所有从本地起飞的班机信息，它的主键为航班号。在定位前先查询座位空数，在定位后将修改座位空数。

2. 定位信息表

	航班号	帐单号	是否领票

定位信息表保存了已定位信息，主键为帐单号，外部主键为航班号。在旅客领票过程时将用到此表，领票后修改是否领票域

3. 帐单信息/旅客资料表

	帐单号	旅客姓名	旅客性别	旅客身份证号	工作单位

帐单信息/旅客资料表保存已定位的旅客信息资料，它的主键为旅客身份证号，外部主键为帐单号。主要用于领票时将旅客信息一同返回。

4. 机位信息表

	班机号	机位信息



机位信息表保存各班机各机位定票状况。它的外部主键为**班机号**。主要用于定票时确定旅客机位，并返回。

5. 2 物理数据结构设计

物理数据结构设计主要是设计数据在模块中的表示形式。数据在模块中都是以结构的方式表示。

1. 旅客信息

1. 旅客姓名	String
2. 旅客性别	Char
3. 身份证号码	LONG INT
4. 工作单位	String

用于定票时接收输入，以及领票时服务器返回。

2. 帐单号

可使用一 LONG INT 实现。

用于领票时接收输入，以及定票时服务器返回。

3. 飞行目的

1. 航班号	String
2. 目的地	String
3. 起飞时间	Date/Time
4. 座位号	String

用于定票时接收输入，以及领票时服务器返回。

4. 网络包



- | | | | |
|---|-------------|--------|---|
| { | 1. 网络包 Head | INT | } |
| | 2. 传输数据 | String | |

用于数据在网络上的传输。

5. 错误信息

- | | | | |
|---|---------|--------|---|
| { | 1. 错误类型 | INT | } |
| | 2. 输出语句 | String | |

用于关于错误操作。

6. 机票信息

- | | | |
|---|----------|---|
| { | 1. 航班号 | } |
| | 2. 目的地 | |
| | 3. 起飞时间 | |
| | 4. 旅客姓名 | |
| | 5. 旅客性别 | |
| | 6. 身份证号码 | |
| | 7. 座位号 | |

用于打印机票准备工作。数据类型可参照上面所述。

7. 帐单信息

- | | | |
|---|----------|---|
| { | 1. 帐单号 | } |
| | 2. 旅客姓名 | |
| | 3. 身份证号码 | |
| | 4. 航班号 | |
| | 5. 起飞时间 | |
| | 6. 目的地 | |



用于打印帐单准备工作。数据类型可参照上面所述。

5.3 数据结构与程序的关系

服务器程序在对定票/领票进行操作时需对数据库数据库数据结构，也就是数据表进行查询和修改：在定票/领票过程中都需要对数据库中的所有表，进行联合查询、修改。

物理数据结构主要用于各模块之间函数的信息传递。接口传递的信息将是以数据结构封装了的数据，以参数传递或返回值的形式在各模块间传输。出错信息将送入显示模块中，机票结构，帐单结构，送入打印准备模块中准备打印格式。

6.运行设计

6.1 运行模块的组合

客户机程序在有输入时启动接收数据模块，通过各模块之间的调用，读入并对输入进行格式化。在接收数据模块得到充分的数据时，将调用网络传输模块，将数据通过网络送到服务器，并等待接收服务器返回的信息。接收到返回信息后随即调用数据输出模块，对信息进行处理，产生相应的输出。

服务器程序的接收网络数据模块必须始终处于活动状态。接收到数据后，调用数据处理/查询模块对数据库进行访问，完成后调用网络发送模块，将信息返回客户机。

6.2 运行控制

运行控制将严格按照各模块间函数调用关系来实现。在各事务中心模块中，需对运行控制进行正确的判断，选择正确的运行控制路径。

在网络传方面，客户机在发送数据后，将等待服务器的确认收到信号，收到后，再



次等待服务器发送回答数据，然后对数据进行确认。服务器在接到数据后发送确认信号，在对数据处理、访问数据库后，将返回信息送回客户机，并等待确认。

6. 3 运行时间

在软体的需求分析中，对运行时间的要求为必须对作出的操作有较快的反应。网络硬件对运行时间有最大的影响，当网络负载量大时，对操作反应将受到很大的影响。所以将采用高速 ATM 网络，实现客户机与服务器之间的连接，以减少网络传输上的开销。其次是服务器的性能，这将影响对数据库访问时间即操作时间的长短，影响加大客户机操作的等待时间，所以必须使用高性能的服务器，建议使用 Pentium III 处理器。硬件对本系统的速度影响将会大于软件的影响。

7. 出错处理设计

7. 1 出错输出信息

程序在运行时主要会出现两种错误：1、由于输入信息，或无法满足要求时产生的错误，称为软错误。2、由于其他问题，如网络传输超时等，产生的问题，称为硬错误。

对于软错误，须在定票/领票操作成功判断及输入数据验证模块由数据进行数据分析，判断错误类型，再生成相应的错误提示语句，送到输出模块中。

对与硬错误，可在出错的相应模块中输出简单的出错语句，并将程序重置。返回输入阶段。

出错信息必须给出相应的出错原因，例：

《全部班机座位已满!》

《查无此定位信息!》等。

7. 2 出错处理对策

所有的客户机及服务器都必须安装不间断电源以防止停电或电压不稳造成的数据丢失的损失。若真断电时，客户机上将不会有太大的影响，主要是服务器上：在断电后恢复过程可采用 SQL SERVER 的日志文件，对其进行 ROLLBACK 处理，对数据进行恢复。

在网络传输方面，可考虑建立一条成本较低的后备网络，以保证当主网络断路时数据的通信。



在硬件方面要选择较可靠、稳定的服务器机种，保证系统运行时的可靠性。

1 安全保密设计

由于数据的传输上需要通过网络传输，为了客户资料进行保密，需要在网络的传输过程中对数据进行加密。

这个工作主要是在准备网络包，及解开网络包这两个模块完成，它们各对数据进行加密及解密还原工作。

在加密算法选择上将使用 RSA 加密算法。具体算法可参照参考资料中《Computer Network》p.598。

2. 维护设计

维护方面主要为对服务器上的数据库数据进行维护。可使用 SQL SERVER 的数据库维护功能机制。例如，定期为数据库进行 Backup，维护管理数据库死锁问题和维护数据库内数据的一致性。

详细设计说明书

1、引言：

1、1 编写目的：

在前一阶段（概要设计说明书）中，已解决了实现该系统需求的程序模块设计问题。包括如何把该系统划分成若干个模块、决定各个模块之间的接口、模块之间传递的信息，以及数据结构、模块结构的设计等。在以下的详细设计报告中将对在本阶段中对系统所做的所有详细设计进行说明。

在本阶段中，确定应该如何具体地实现所要求的系统，从而在编码阶段可以把这个描述直接翻译成用具体的 程序语言书写的程序。主要的工作有：根据在《需求分析说明书》中所描述的数据、功能、运行、性能需求，并依照《概要设计说明书》所确定的处理流程、总体结构和模块外部设计，设计软件系统的结构设计、逐个模块的程序描述（包括各模块的功能、性能、输入、输出、算法、程序逻辑、接口等等），解决如何

1. 接受：旅客信息及取票通知和帐单；
2. 输出：取票通知和帐单及机票；
3. 网络输出和加密，输入和解密；



4. 分辨信息的种类并采取相应的处理步骤;
5. 判断信息的正误并采取相应的处理步骤;
6. 进行数据库的查询、修改工作;
7. 接受并判断错误, 输出相应的出错消息;

1、2 文中特殊的定义和缩写:

1. 2. 1 定义

SQL SERVER: 系统服务器所使用的数据库管理系统 (DBMS)。

SQL: 一种用于访问查询数据库的语言

事务流: 数据进入模块后可能有多种路径进行处理。

主键: 数据库表中的关键域。值互不相同。

外部主键: 数据库表中与其他表主键关联的域。

ROLLBACK: 数据库的错误恢复机制。

1. 2. 2 缩写

系统: 若未特别指出, 统指本机票预定系统。

SQL: Structured Query Language (结构化查询语言)。

ATM: Asynchronous Transfer Mode (异步传输模式)。

2. 总体设计:

2、1 需求概要:

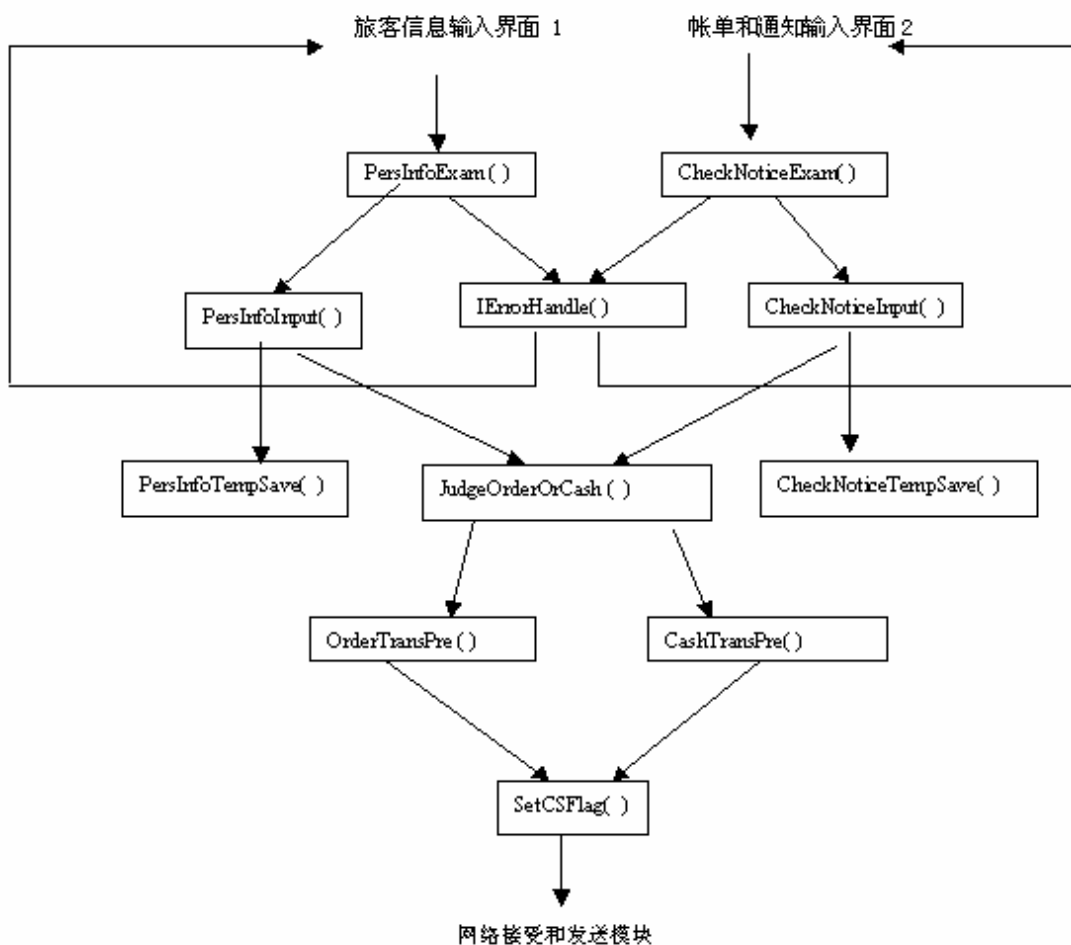
XX 航空公司为方便旅客, 需开发一个机票预定系统。为便于旅客由旅行社代替航空公司负责为旅客定票, 旅行社把预定机票的旅客信息, 包括姓名、性别、工作单位、身份证号码、旅行时间、旅行目的地, 输入机票预定系统的客户端程序, 系统经过查询航空公司内的航班数据服务器后, 为旅客安排航班, 印出取票通知。旅客在飞机起飞前一天凭取票通知和帐单交款后取票, 系统校对无误后即印出机票给旅客。

要求系统能有效、快速、安全、可靠和无误的完成上述操作。并要求客户机的界面要单明了, 易于操作, 服务器程序利于维护。

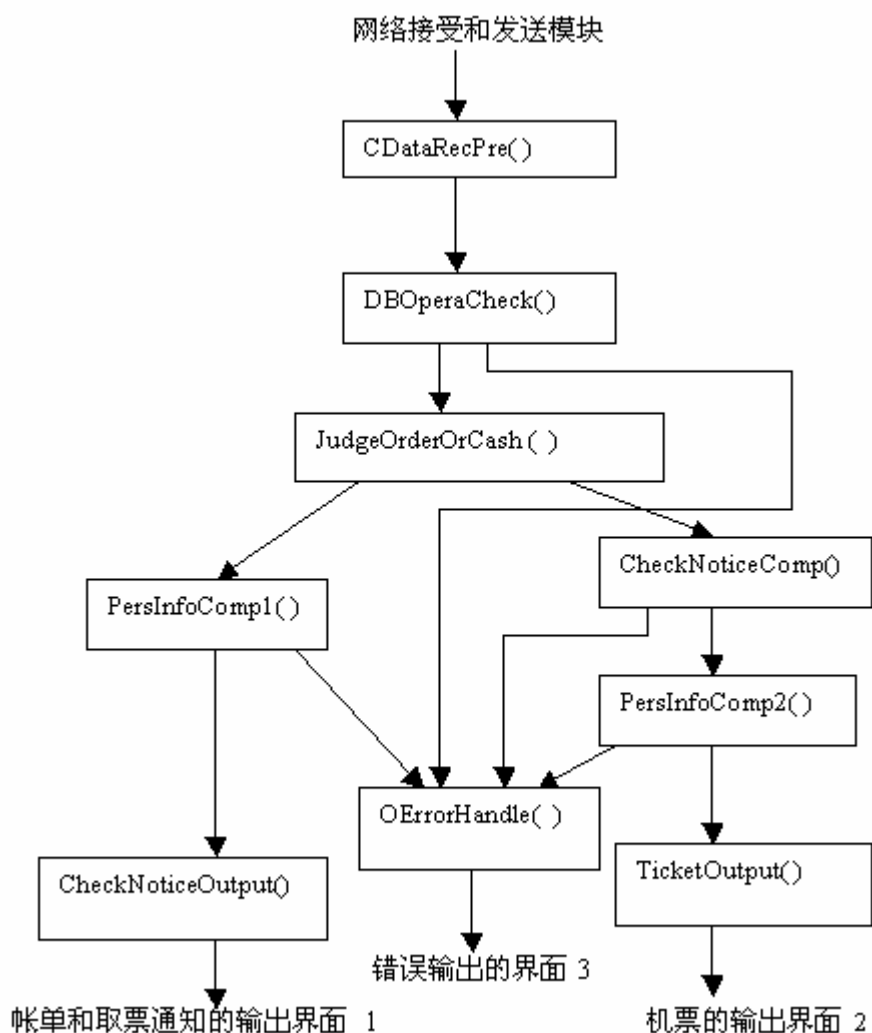
2、2 软件结构:

各模块之间的关系已由概要设计给出。

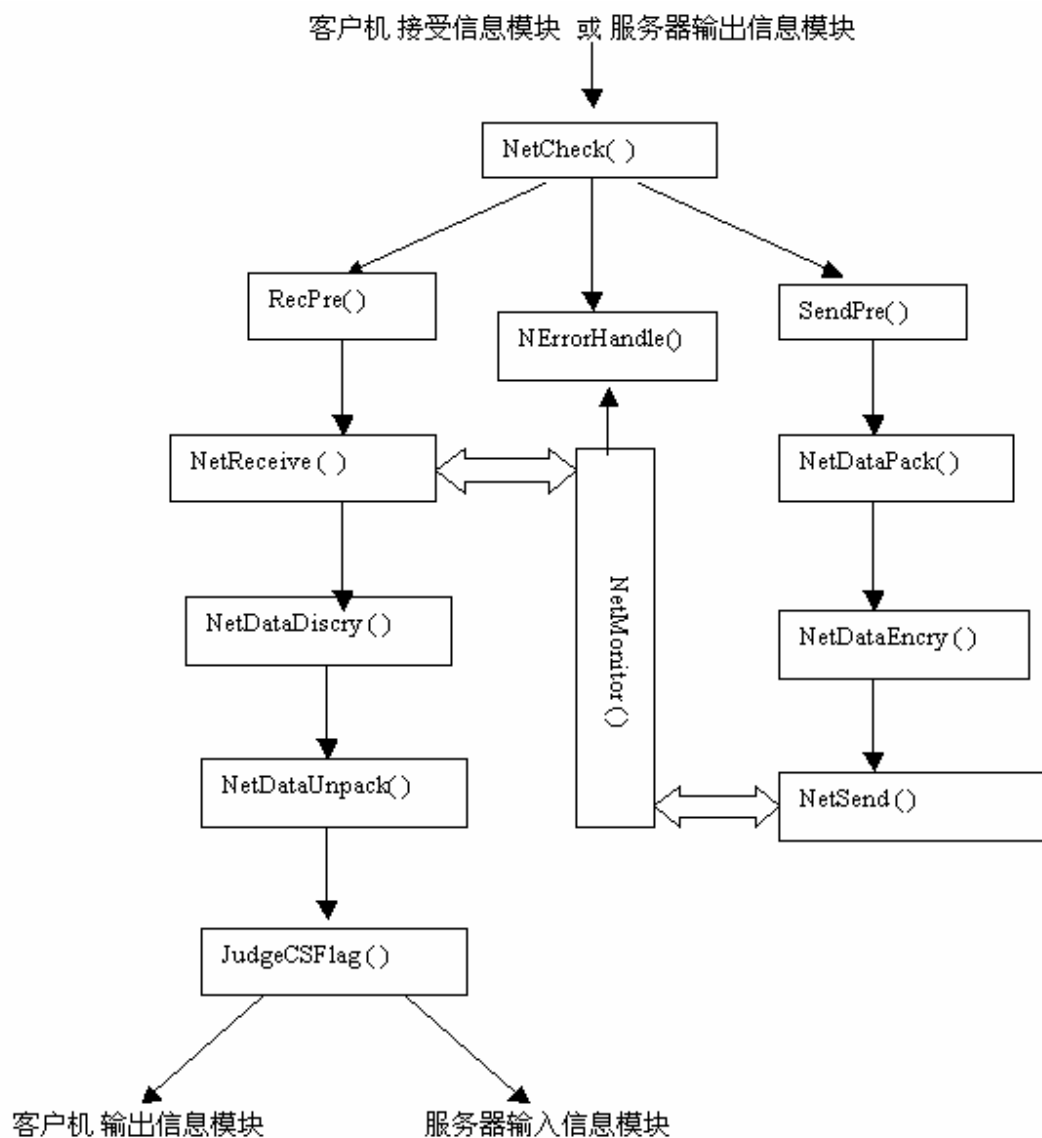
1、客户机接受信息模块结构图:



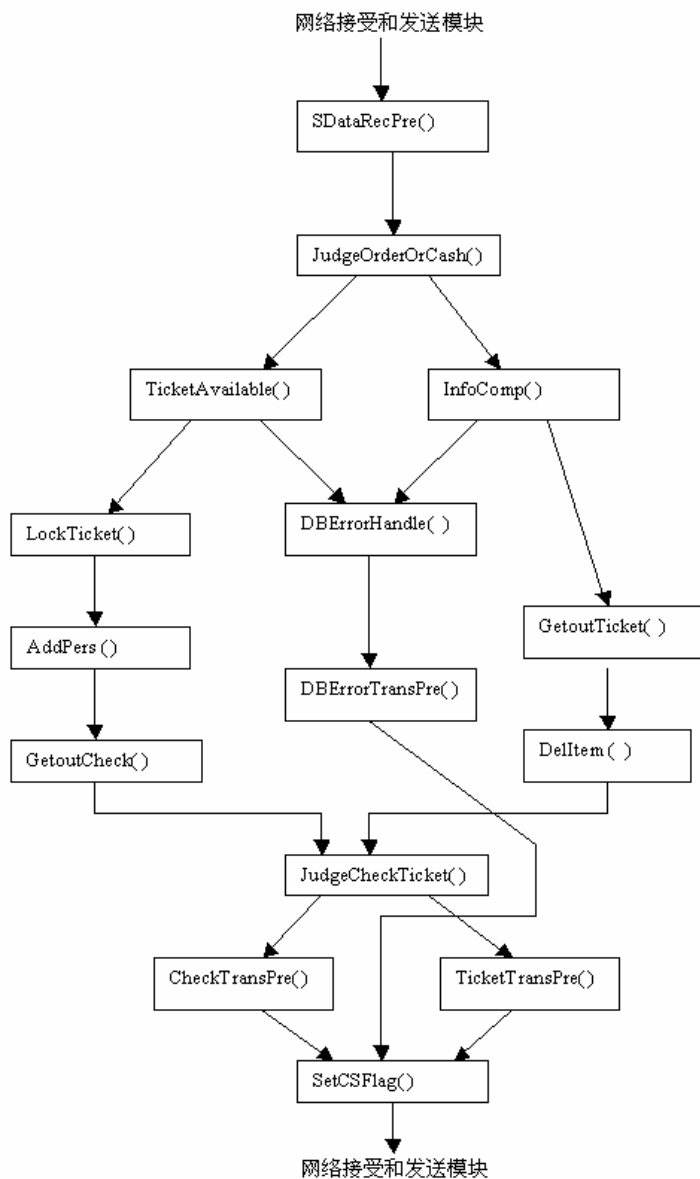
2.客户机输出信息模块:



3.网络接受和发送模块结构:



4.服务器模块:





3、程序描述:

3、1 客户机接受信息模块:

1、PersInfoExam 过程:

对在旅客信息界面中输入的各项信息进行初步检验。若发现错误，令 PErrorAppear=T, 判断错误类型，并将相应的 错误类型 PErrorType 或 PErrorRank 作为参数，转入 IErrorHandle 过程。若未发现错误，转入 PersInfoInput 过程。其中的错误种类有：

1. 数据类型不匹配, PErrorType =T; 否则=F;

姓名	string	旅行目的地	string
性别	string	旅行时间	date
工作单位	string	(年/月/日	yy/mm/dd)
身份证号码	long int	定票/取票	boolean

2、数据超出规定范围 PErrorRank =T; 否则=F;等等

性别只能是‘男’或‘女’；身份证号码按规定必须是 13 位；旅行时间必须在定票的当天过一天以后；等等

2、PersInfoInput 过程:

经检验无误后，将输入界面表单中的数据输入到 Class PersInfo

```
Class PersInfo{           /* 伪码 */
    String name= 姓名  ;
    String sex= 性别;
    String company= 工作单位;
    Long int idcode= 身份证号码;
    Date stime= 旅行时间;
    (syear/smonth/stime=年/月/日)
    String denist= 目的地
    Bool ocflag= 定票/取票
}
```



3.PersInfoTempSave 过程:

将由 PersInfoInput 输入的旅客信息 保存在一个临时文件 PersInfoTemp.txt 中。以备与将来从服务器经网络传输过来的数据校验。 此类 temp 文件在每次软件的完全过程操作后，须删除

4.CheckNoticeExam 过程:

对在帐单和取票通知的信息界面中输入的各项信息进行初步检验。若发现错误，令 CErrorAppear=T,判断错误类型，并将相应的 错误类型 CErrorType 或 CErrorRank 作为参数，转入 IErrorHandle 过程。若未发现错误，转入 CheckNoticeInput 过程。其中的错误种类有:

1.数据结构不匹配,CErrorType=T,否则=F;

帐单号 long int

姓名 string

身份证号码 long int

付款金额 money

航班号 string

取票截止日期 date

目的地 string

2. 数据超出规定范围 CErrorRank =T; 否则=F;

如帐单号不是规定的 15 位;金额为负;取票截止日期已过;等等

5.CheckNoticeInput 过程:

经检验无误后，将输入界面表单中的数据输入到 Class CheckNotice

Class CheckNotic{

Long int cncode=帐单号

String name=姓名

Long int idcode=身份证号码



```

Money      bill =      付款金额

String      planecode=   航班号

Date        dtime=      取票截止日期

String      denist=      目的地

    }
    
```

6.CheckNoticeSave 过程:

将由 CheckNoticeInput 输入的旅客信息 保存在一个临时文件 CheckNoticeTemp.txt 中。以备与将来从服务器经网络传输过来的数据校验。 此类 temp 文件在每次软件的完全过程操作后，须删除。

7.IErrorHandle 过程:

在 PersInfoExam 或 CheckNoticeExam 过程中若发现错误,就转入本过程,执行相应的处理,并输出相应的出错信息。

```

If (PErrorType==T)
    输出 “您的个人信息可能输错了位置。请重试。”
    
```

```

If (PErrorRank==T)
    输出 “您的个人信息不适应规定范围。请重试。”
    
```

再转入旅客信息输入的界面中。

```

If (CErrorType==T)
    输出 “您的帐单信息可能输错了位置。请重试。”
    
```

```

If (CErrorRank==T)
    输出 “您的帐单信息不适应规定范围。请重试。”
    
```

再转入帐单和取票通知信息的输入界面。

8.JudgeOrderOrCash 过程:

根据 PersInfoInput 输入的 ocflag 判断本次操作是定票或取票，相应的设置标志，相应的转入 OrderTransPre 或 CashTransPre 过程。

9.OrderTransPre 过程:



定票要求传输前,作好各方面(硬件、软件)的准备。如准备好要传输的定票信息,包括旅客信息,客户机信息等等。客户机信息包括唯一序列号 `c_id`,并设置等待标志 `waitflag = T` 和等待开始时间 `waitstime`,等等。转入网络模块后,此过程仍处于等待状态,并一直记时,当 `waitflag == T` 且系统时间超过 `waitstime` 比如十分钟后,结束操作,输出操作超时的出错信息。当客户机接受到数据库的信息,客户机输入模块中的 `DataRecPre` 过程会置 `waitflag = F`,这样本过程就会结束。

10. CashTranPre 过程:

取票要求传输前,作好各方面(硬件、软件)的准备。如准备好要传输的取票信息,包括帐单和取票通知的信息,客户机信息等等。客户机信息包括唯一序列号 `c_id`,并设置等待标志 `waitflag = T` 和等待开始时间 `waitstime`,等等。转入网络模块后,此过程仍处于等待状态,并一直记时,当 `waitflag == T` 且系统时间超过 `waitstime` 比如十分钟后,结束操作,输出操作超时的出错信息。当客户机接受到数据库的信息,客户机输入模块中的 `DataRecPre` 过程会置 `waitflag = F`,这样本过程就会结束。

11.SetCSFlag 过程:

设置客户机/服务器标志 `CSFlag = T`(表示将要进行的传输是从客户机到服务器)。以备网络模块中的 `JudgeCSFlag` 过程使用。

3.2 客户机输出信息模块:

1.CDataRecPre 过程:

作好各方面(硬件、软件)的准备,以便接受网络传输来的数据。如设置服务器的空闲与否状态 `s_idle = F`,继承正在工作联系的客户机序列号 `c_id`。完成后转入下一个过程 `DBOperaCheck`。

2.DBOperaCheck 过程:

检验网络传输过来的数据。若是出错信息,转入 `OErrorHandle` 过程。否则转入 `JudgeOrderOrCheck` 过程。

3.JudgeOrderOrCheck 过程:



根据网络传输来的数据中的 ocflag 判断本次操作是定票或取票，相应的设置标志，相应的转入 PersInfoComp 或 CheckNoticeComp 过程。

4.PersInfoComp1 过程:

将数据库中的旅客信息与临时文件 PersInfoTemp.txt 中的旅客信息进行比较，如果吻合，就转入 CheckNoticeOutput 过程。否则出错，转入 OErrorHandle 过程。

5.CheckNoticeOutput 过程:

将帐单和取票通知输出到界面 1 中。注意格式。

```
Long int  cncode =帐单号;

String name= 姓名 ;

String sex= 性别;
String company= 工作单位;
Long int idcode= 身份证号码;
Date stime= 旅行时间;
(syear/smonth/stime=年/月/日)
String denist= 目的地
Bool ocflag= 定票/取票
```

6.CheckNoticeComp 过程:

将数据库中的帐单和取票通知信息与临时文件 CheckNoticeTemp.txt 中的旅客信息进行比较，如果吻合，就转入 PersInfoComp 过程。否则出错，转入 OErrorHandle 过程。

7. PersInfoComp2 过程:

将数据库中的旅客信息与临时文件 PersInfoTemp.txt 中的旅客信息进行比较，如果吻合，就转入 TicketOutput 过程。否则出错，转入 OErrorHandle 过程。

8.TicketOutput 过程:

将机票输出到界面 2。注意格式。

```
Long int planeCode=航班号;

Date/time stime=起飞时间;
```



```
String name=姓名;  
Long int idcode=身份证号码;  
Int seatcode = 座位号;  
Money price= 票价;
```

9. OErrorHandle 过程:

处理各个过程产生的错误, 输出相应的出错信息到输出界面 3。由 DBErrorCheck 过程转入的, 是数据库操作失效, 如: 所需机票已被定光; 数据库中无对应的帐单或旅客信息; 等等。由 PersInfoComp 过程转入的, 是数据库处理有误, 返回的旅客信息无法与原先输入的旅客信息吻合。由 CheckNoticeComp 过程转入的, 是数据库处理有误, 返回的帐单和取票通知信息无法与原先输入的帐单和取票通知信息吻合。

3.3 网络接受和发送模块结构:

1.NetCheck 过程:

检查网路的工作状况。发送一些检验消息, 接受一些反馈信息, 查询要建立通路的终端是否有效, 若有效就作好响应的准备, 转入 RecPre 和 SendPre 过程; 若无效就转入 NErrorHandle 过程。

如服务器要传输信息给某客户机, 可先发检验消息, 包括由服务器发送的 c_id 可检验相应的客户机是否仍处于等待状态, 客户机再将 waitflag 反馈给服务器; 客户机要传输信息给服务器, 若服务器的 s_idle=T, 由客户机发送的 c_id 的值可赋予服务器中的 c_id, 再将 s_idle 反馈给客户机。

2. RecPre 过程:

经过 NetCheck 过程检验网路后, 作好接受数据的准备。若是服务器→客户机, 应该只要相应 c_id 的客户机做准备, 激活客户机输出模块。若是客户机→服务器, 由于服务器始终处于活动状态, 无须 激活。

3. NetReceive 过程:



接受网络数据包。在分包接受的过程中，始终保持与 NetMonitor 过程联系，接受和反馈网络数据传输状况的信息，根据这些信息继续接受数据包，或做相应的调整。

NetDataDiscry 过程：

将由网络传输来的数据依据一定的算法解密。

NetDataUnpack 过程：

将经解密后的数据（被压缩过的）依据一定的算法解压。

4.SendPre 过程：

经过 NetCheck 过程检验网路后，作好发送数据的准备。若是服务器→客户机，应该只要相应 c_id 的客户机做准备，激活客户机输出模块。若是客户机→服务器，由于服务器始终处于活动状态，无须 激活。

5.NetDataPack 过程：

将由服务器模块传过来的数据依据一定的算法压缩，打包。

6.NetDataEncry 过程：

将准备送网络发送的数据依据一定的算法加密。

7.NetSend 过程：

发送经过压缩打包和加密的数据包。在分包发送的过程中，始终保持与 NetMonitor 过程联系，接受和反馈网络数据传输状况的信息，根据这些信息继续发送数据包，或做相应的调整。

8.NetMonitor 过程：

从服务器与某客户机建立网络联系开始，到数据传输完成，联系断开为止，NetMonitor 过程始终监视着网络状况。并将状况信息随时传给 NetReceive 和 NetSend 过程，同时从这两个过程接受数据接受和发送的状况信息。如果以上的状况信息反映出错误，就转入 NErrorHandle 过程处理。

9. NErrorHandle 过程：



网络模块的错误处理过程。主要的错误种类有：1。无法建立服务器与某客户机建立网络联系，由 NetCheck 过程转来；2。联系建立后，网络数据传输出现错误，由 NetMonitor 过程转来。

10.JudgeCSFlag 过程:

本过程根据由客户机输入模块中的 SetCSFlag 过程或服务器模块中的 SetCSFlag 过程来确定网络模块完成后应转入哪一个模块，客户机输出模块或服务器模块。

3.4 服务器模块:

1. SDataRecPre 过程:

作好服务器各方面（硬件、软件）的准备，以便接受网络传输来的数据。如设置服务器的空闲与否状态 s_idle =F,继承正在工作联系的客户机序列号 c_id 。完成后转入下一个过程 JudgeOrderOrCash 。

2. JudgeOrderOrCash 过程:

根据网络输入的数据中的 ocflag 判断本次操作是定票或取票，相应的设置标志，相应的转入 TicketAvailable 或 InfoComp 过程。

3. TicketAvailable 过程:

根据旅客信息中的旅行时间(日期)查询数据库,若 Ticket 表中在此日期仍有票剩余(未被定票或购票),就转入 LockTicket 过程;若无余票,就转入 DBErrorHandle 过程。

4. LockTicket 过程:

根据旅客信息中的旅行时间(日期),在 Ticket 表中,把相应数量的起飞日期等于此日期的机票锁住,作为已定的机票。

5. AddPers 过程:

把已定票的旅客信息添加到 PerInfo 表中。

姓名	性别	身 份 证 号码	工作单位	旅 行目的地	旅行时间
----	----	-------------	------	-----------	------



6. GetoutCheck 过程:

把查询后许可的定票信息添加到 Check 表中,同时准备把此信息向网络输出.

帐 单 号	姓名	身份证号码	金额	航班号	截止日期	目的地

7.InfoComp 过程:

检查输入的旅客信息、帐单和取票通知信息在数据库的 PersInfo 和 Check 表中是否有对应的项。如果有，就转入 DelItem 过程；如果无，就转入 DBErrorHandle 过程。

8. GetoutTicket 过程:

在 Ticket 表中取出并删除对应的被锁住的机票信息，准备把此信息向网络输出.

9. DelItem 过程:

在数据库的 PersInfo 和 Check 表中，删除已取票的旅客信息、帐单和取票通知信息。

10JudgeCheckTicket 过程:

根据数据库输出的数据 判断本次输出是帐单和取票通知或机票，相应的设置标志，相应的转入 CheckTransPre 或 TicketTransPre 过程.

11.CheckTransPre 过程:



帐单和取票通知 传输前,作好各方面(硬件、软件)的准备。如准备好要传输信息,包括帐单信息,要联系的客户机信息等等。客户机信息包括唯一序列号 c_id。

12.TicketTransPre 过程:

机票 传输前,作好各方面(硬件、软件)的准备。如准备好要传输信息,包括机票信息,要联系的客户机信息等等。客户机信息包括唯一序列号 c_id。

13.SetCSFlag 过程:

设置客户机/服务器标志 CSFlag=F(表示将要进行的传输是从服务器到客户机)。以备网络模块中的 JudgeCSFlag 过程使用。

14.DBErrorHandle 过程:

判断数据库操作中出现的错误,并处理和返回到客户机上。由 TicketAvailable 过程转入的错误,是 Ticket 表中在相应日期无票剩余(全被定票或购票,或者飞机已起飞);由 Infocomp 过程转入的错误,是输入的旅客信息、帐单和取票通知信息在数据库的 PersInfo 和 Check 表中没有对应的项,说明取票信息和定票信息不符。

15.DBErrorTransPre 过程:

错误信息 传输前,作好各方面(硬件、软件)的准备。如准备好要传输信息,包括错误类别信息,要联系的客户机信息等等。客户机信息包括唯一序列号 c_id。接着转入网络接受和发送模块。

软件测试和调试

1. 测试计划

1.1 编写目的

在开发大型软件的漫长过程中,面对极其错综复杂的问题,人的主观认识不可能完全符合客观现实,与工程密切相关的各类人员之间的通信和配合也不可能完美无缺。因此,在软件生命周期的每个阶段都不可避免地会产生差错。尤其对于机票预订系统这类会影响人们生活、财产的工程软件,必须尽量减少差错,以免造成严重的损失。测试是“为了发现程序中的错误而执行程序的过程”。测试的目的就是在软件投入生产性运行之



前，尽可能多的发现软件中的错误。目前软件测试仍然是保证软件质量的关键步骤，它是对软件规格说明、设计和编码的最后复审，也是必不可少的关键步骤。

4.2 测试用例

4.2.1 输入

PersInfoExam 过程:

在 PersInfoExam 过程中，首先对用户的输入信息进行测试。客户机上的输入信息为旅客资料或帐单号，还包括一个定票/领票选项。输出为打印帐单或机票，和确认或出错信息。

在输入的测试数据中可分为有效输入类、无效输入类。

其中的错误种类有：

无效输入类

1. 数据类型不匹配, PErrorType =T; 否则=F;

姓名	string	旅行目的地	string
性别	string	旅行时间	date
工作单位	string	(年/月/日	yy/mm/dd)
身份证号码	long int	定票/取票	boolean

2. 数据超出规定范围 PErrorRank =T; 否则=F;等等

性别只能是‘男’或‘女’；身份证号码按规定必须是 13 位

旅

行时间必须在定票的当天过一天以后；等等

有效输入类

1. 当用户的输入只有旅客资料或帐单号时，则

CheckNoticeExam 过程:

在 CheckNoticeExam 过程中，对在帐单和取票通知的信息界面中输入的各项信息进行初步检验。

无效输入类：

1.数据结构不匹配,CerrorType=T,否则=F;



帐单号 long int
姓名 string
身份证号码 long int
付款金额 money
航班号 string
取票截止日期 date
目的地 string

2. 数据超出规定范围 CErrorRank =T; 否则=F;

如帐单号不是规定的 15 位;金额为负;取票截止日期已过;等等

PersInfoInput 过程:

将输入界面表单中的数据输入到 Class PersInfo

```
Class PersInfo{          /* 伪码 */
    String name= 姓名 ;
    String sex= 性别;
    String company= 工作单位;
    Long int idcode= 身份证号码;
    Date stime= 旅行时间;
                (syear/smonth/stime=年/月/日)
    String denist= 目的地
    Bool ocflag= 定票/取票
}
```

CheckNoticeInput 过程:

经检验无误后, 将输入界面表单中的数据输入到 Class CheckNotice

```
Class CheckNotic{
```



```

Long int  cncode=帐单号
String   name=姓名
Long int  idcode=身份证号码
Money    bill =    付款金额

String    planecode=  航班号

Date      dtime=    取票截止日期

String    denist=    目的地
}

```

4.2.2 输出

PersInfoExam 过程:

对在旅客信息界面中输入的各项信息进行初步检验。若发现错误, 令 PErrorAppear=T, 判断错误类型, 并将相应的错误类型 PErrorType 或 PerrorRank 作为参数, 转入 IErrorHandle 过程。若未发现错误, 转入 PersInfoInput 过程。

PersInfoInput 过程:

无

CheckNoticeExam 过程:

对在帐单和取票通知的信息界面中输入的各项信息进行初步检验。若发现错误, 令 CErrorAppear=T, 判断错误类型, 并将相应的 错误类型 CErrorType 或 CErrorRank 作为参数, 转入 IErrorHandle 过程。若未发现错误, 转入 CheckNoticeInput 过程。

ErrorHandle 过程:

在 PersInfoExam 或 CheckNoticeExam 过程中若发现错误,就转入本过程,执行 相应的处理,并输出相应的出错信息。

If (PErrorType==T)

输出 “您的个人信息可能输错了位置。请重试。”



If (PErrorRank==T)

输出“您的个人信息不适应规定范围。请重试。”

再转入旅客信息输入的界面中。

If (CErrorType==T)

输出“您的帐单信息可能输错了位置。请重试。”

If (CErrorRank==T)

输出“您的帐单信息不适应规定范围。请重试。”

再转入帐单和取票通知信息的输入界面。

DBOperaCheck 过程:

检验网络传输过来的数据。若是出错信息，转入 OErrorHandle 过程。否则转入 JudgeOrderOrCheck 过程。

CheckNoticeOutput 过程:

将帐单和取票通知输出到界面 1 中。注意格式。

Long int cncode=帐单号;

String name= 姓名 ;

String sex= 性别;

String company= 工作单位;

Long int idcode= 身份证号码;

Date stime= 旅行时间;

(syear/smonth/stime=年/月/日)

String denist= 目的地

Bool ocflag= 定票/取票

TicketOutput 过程:

将机票输出到界面 2。注意格式。

Long int planecode=航班号;

Date/time stime=起飞时间;



```
String name=姓名;  
Long int idcode=身份证号码;  
Int seatcode = 座位号;  
Money price= 票价;
```

4.2.3 步骤及操作

在测试过程中，首先需要对各子单元过程进行测试。各子单元过程的测试必须先由程序设计员调试并编译通过后才能进行。在各子单元过程测试完毕后，再对各模块（包括各子单元过程之间的接口）进行测试，处理好各模块之间的接口，最后对系统进行测试和维护。

其操作过程如下：

1. 在客户机接受信息模块过程中，先对各子单元过程分别进行测试，然后根据白盒法按照详细设计说明书中的流程图对其进行跟踪测试。
2. 同样，在客户机输出信息模块、网络接受和发送模块结构和服务模块（包括数据库）过程中先对各子单元过程分别进行测试，然后根据白盒法按照详细设计说明书中的流程图对其进行跟踪测试。
3. 然后，根据各模块之间的各种关系，对其接口进行测试。
4. 在系统测试中，要注意对各种意外情况（列如断电、硬盘损坏等）加以处理，对数据库要注意其安全性、可靠性、健壮性、效率。网络传输更要注意其安全性。

测试项目

客户机接受信息模块测试

客户机接受用户输入的各种数据（包括旅客资料或帐单号，还包括一个定票/领票选项）然后经网络传送给服务器。

客户机输出信息模块测试

客户机输出为打印帐单或机票，和确认或出错信息。

网络接受和发送模块结构测试



接受由服务器程序经网络传送到客户机的数据包，它是程序与网络的接口。经解码后发送数据给服务器数据库。

服务器模块（包括数据库）测试

测试数据库的安全性，可靠性，健壮性，效率。

各模块之间的接口测试

对各模块之间的接口进行测试。

系统测试

用黑盒法对系统进行各类功能的测试。

1. 软件需求测试结论

在经过对各模块的测试后，已经能够证实该软件各方面的能力都可以。只是在网络传输方面还有待加强。

4. 评介

4.1 软件能力

经测试证实该软件在各方面的综合能力都可以。

4.2 缺陷和限制

网络传输的保密性，安全性以及数据库的安全性还存在一定的缺陷。有可能会由于传输过程中数据的丢失造成软件运行的错误。

4.3 建议

建议在网络传输方面加强其保密性和安全性。

4. 测试结论

该软件在各方面的综合能力都可以，通过！

附录 4 Rose2003 软件简介

Rose 是用 UML 快速开发应用程序的工具之一，支持用例图、活动图、协作图、状态图、组件图和布局图，通过正向和逆向转出工程代码的特性，可以支持 C++、Java、Visual Basic 和 XML DTD 等代码生成和逆向转出工程代码。以 Rose2003 为基础介绍采用 Rose 进行分析和设计的过程。

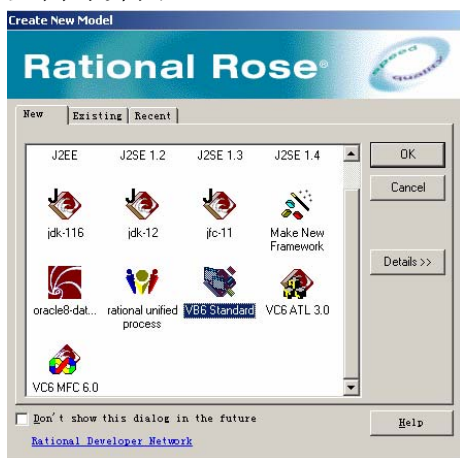
1 使用 Rose2003 设计软件的思路

采用 Rose2003 进行软件设计的基本思路是：

- (1) 启动 Rose2003;
- (2) 选择目标项目 (project) 的向导 (wizards);
- (3) 创建项目;
- (4) 建立各种框图;
- (5) 生成代码的框架。

2 使用 Rose2003 创建模型

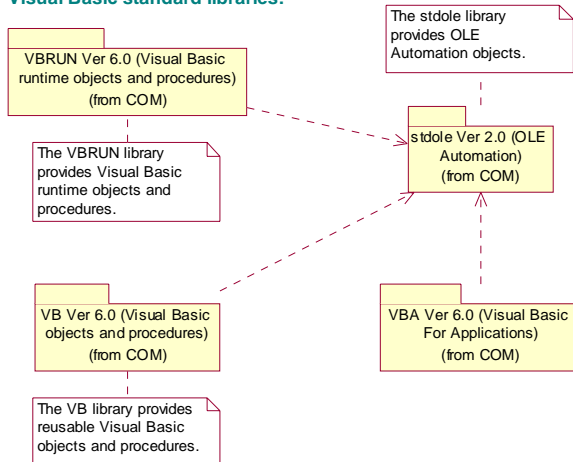
可以通过下面介绍的过程建立一个模型。启动 Rose 后，得到图所示的模型向导界面。



选择模型类型

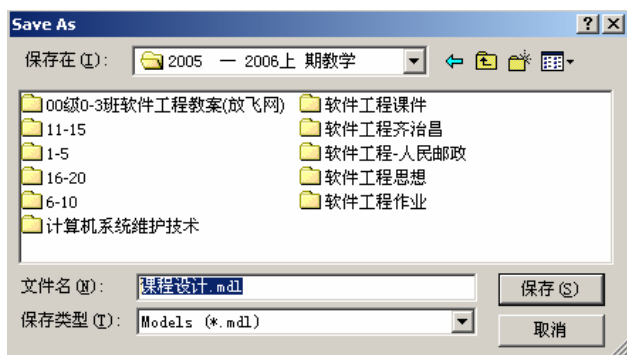
选择模型最终实现的工具，比如选择：“VB6 Standard”，按下“OK”，进入设计画面(见图)。如果不想选用向导进行设计，直接点击“Cancel”也可进入设计画面。

Visual Basic standard libraries:

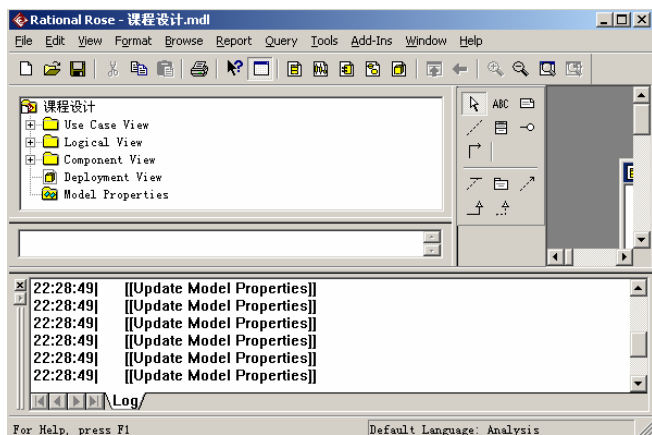


Note: The framework does not contain the methods and properties on the interfaces of these libraries. To import also the methods and properties, right-click on an interface or on the corresponding component in the browser and click Full Import.

右键点击浏览器中的“untitled”，在弹出的快捷菜单中选择“save”，弹出文件保存对话框。

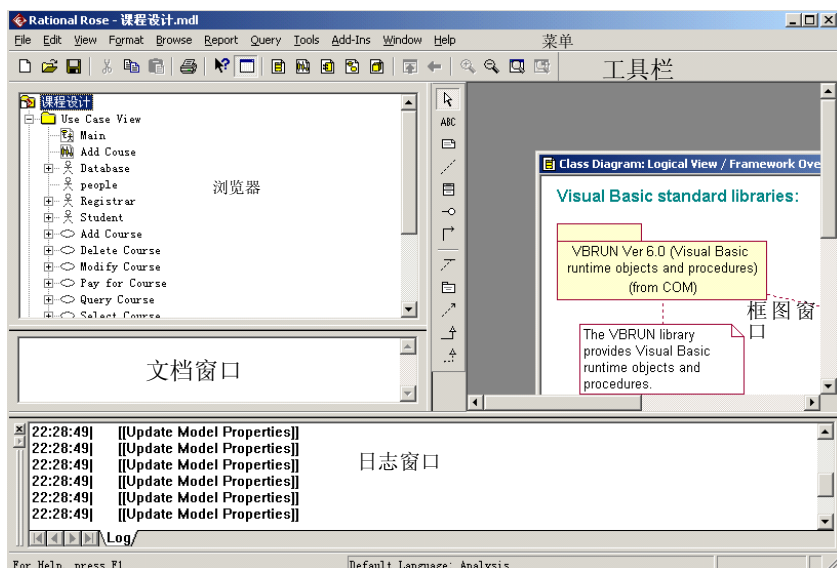


输入“课程设计”，按下“保存”按钮，即建立一个名为“课程设计.mdl”的文件。



3.3 Rose 界面的组成

Rose2003 的界面由 6 个部分组成，分别是菜单、浏览器、文档窗口、框图窗口、日志窗口和工具栏。



Rose2003 界面组成

菜单：集成了系统中几乎所有的操作。包括File、Edit、View、Add. Ins、Window、Help 几大项。选择其中一项，又会弹出相应的子菜单。

浏览器：用于在 Rose 模型中迅速漫游。图所示为浏览器的画面图。

可以看到，UML 的视图包括用例视图(use case view)、逻辑视图(logical view)、组件视图(component view)和布局视图(deployment view)。

文档窗口：用于查看或更新模型元素的文档。

工具栏：用于迅速访问常用命令。

图窗口：用于显示和编辑一个或几个 UML 图。

日志窗口：用于查看错误信息和报告各个命令的结果。

下面分别介绍它们的具体作用。

1. 菜单

菜单集成了 Rose2003 中的所有功能。主要分 File、Edit、View、Format、Bowse、Report、Query、Tools、Add-Ins、Windows、Help 等项。利用菜单，可以方便地进行各种各样的操作。一些菜单项的主要功能如表 1～表 8 所示，其余菜单项略。



表 1 File 子菜单

New	新建文件
Open	打开文件
Save	保存文件
Save as	另存文件
Save Log As	命名保存日志文件
AutoSave Log	自动保存
Clear Log	清除日志
Load Model WorkSpace	载入模型工作空间
Save Model WorkSpace	保存模型工作空间
Save Model WorkSpace As	命名保存模型工作空间
Units	与单元相关的菜单项
Import	载入文件
Export	将模型输出到……文件
Update	更新
Print	打印
Page Setup	页面设置
Edit Path Map	编辑路径映射关系
Exit	退出

表 2 Edit 子菜单

Undo	撤销
Redo	恢复
Cut	剪切
Copy	拷贝
Paste	粘贴
Delete	删除
Select All	全部选择
Delete from Model	从模型中删除
Relocate	重新部署
Find	查找



Reassign 再分配
Compartment 间隔
Change Info 改变信息

表 3 View 子菜单

Toolbars	工具栏
Statusbars	状态栏
Documentation	文档
Browser	浏览器
Zoom to Selection	全部显示选择区域
Zoom In	放大
Zoom out	缩小
Fit in Window	适合于窗口
Undo Fit in Windows	Fit in Window 的相反操作
Page Breaks	页面的间隔
Refresh	刷新
As Booch	显示成 Booch 图的形式
As OMT	显示成 OMT 图的形式
As Unified	显示成 Unified 形式

表 4 Format 子菜单

Font Size	字号大小
Font	字库
Line Color	线的颜色
Fill Color	填充颜色
Use Fill Color	使用填充颜色
Automatic Resize	自动调整大小
Stereotype Display	版型显示
Stereotype Label	版型标签
Show Visibility	显示可见性



Show Compartment Stereotypes	显示分割版型
Show Operation Signature	显示操作签名
Show All Attributes	显示所有属性
Show All Operations	显示所有操作
Show All Columns	显示所有列
Show All Triggers	显示所有触发子
Suppress Attributes	隐匿属性
Suppress Operations	隐匿操作
Suppress Columns	隐匿列
Suppress Triggers	隐匿触发子
AutoSize All	自动更改大小

表 5 Browse 子菜单

Use Case Diagram	用例图
Class Diagram	类图
Module Diagram	模块（组件）图
Process Diagram	进程（配置）图
Interaction Diagram	片断（交互）图
State Machine Diagram	状态机图
Expand	展开
Specification	规格
Top Level	顶层
Referenced Item	参照项
Previous Diagram	前一个图
Create Object Message Diagram	创建对象消息图

表 6 Report 子菜单

Show Usage	显示使用的信息
Show Instances	显示实例信息



Show Access Violations	显示违反存取的信息
Show Participants In UC	显示在 UC 中的参与者
表 7 Query	子菜单

Add Classes	增加类
Add Use Cases	增加用例
Expand Selected Elements	扩展选择的元素
Hide Selected Elements	隐藏选择的元素
Filter Relationships	过滤关系

表 2—8 Tools 子菜单

Create	创建
Check Model	检查模型
Model Properties	模型属性
Options	选项
Open Script	打开描述
New Script	新的描述
ANSI C++	
Ada 83	
Ada 95	
CORBA	
Data Modeler	数据建模
J2EE Deploy	DDL
Java/J2EE	
Oracle8	
Quality Architect	质量架构
Model Integrator	模型集成
Web Publisher	Web 发布
TopLink	
COM	



Version Control 版本控制

Visual Basic

Web Modeler Web 建模

XML-DTD

Class Wizard 类向导

2. 浏览器

利用浏览器，可以实现如下操作：

- 1) 增加模型元素(包括角色、用例、类、组件、图等)
- 2) 查看现有模型元素；
- 3) 查看现有模型元素之的关系。
- 4) 移动模型元素；
- 5) 更名模型元素；
- 6) 将模型元素加进图；
- 7) 将文件或 URL 链接到元素；
- 8) 将元素组成包；
- 9) 访问元素的详细规范；
- 10) 打开图。

浏览器中有 4 个视图：Use Case 视图、Logical 视图、Component 视图和 Deployment 视图，表 9 是每个视图以及其中包含的模型元素或操作。

表 9 Rational Rose 中的视图



Use Case 视图	Open Specification		
	New	Category	类目
		Use Case	用例
		Actors	角色
		Class	类
		Use Case Diagram	用例图
		Class Diagram	类图
		Object Message Diagram	对象消息图
		Message Trace Diagram	消息追踪图
		StateChart Diagram	状态图
		Activity Diagram	活动图
		File	文件
		URL	链接
	Units	Load	装载
		Save	保存
		Save AS	另存
		Unload	卸载
		Control Use Case View	控制用例图
		Uncontrol	解除控制
		Write Protection	写保护
	Data Modeler	New	新建
		Forward Engineer	前向工程
		Transform to Data Model	转换成数据模型
		Transform to Object Model	转换成对象模型
		Compare and sync	比较和同步
	Quality Architect	Generate Unit Test	生成单元测试
Logical 视图	Open Specification		
	New	Class	类
		Class Utility	实用类
		Use Case	用例
		Interface	界面
		Category	类目
	New	Class Diagram	类图
		Use Case Diagram	用例图
		Object Message Diagram	对象消息图
		Message Trace Diagram	消息追踪图
		StateChart Diagram	状态图
		Activity Diagram	活动图
		File	文件
		URL	链接





Logical 视图	Units	Load	装载
		Save	保存
		Save As	另存
		Unload	卸载
		Control Logical View	控制逻辑视图
		Uncontrol	解除控制
		Write Protection	写保护
	Data Modeler	New	新建
		Forward Engineer	前向工程
		Transform to Data Model	转换成数据模型
		Transform to Object Model	转换成对象模型
		Compare and sync	比较和同步
	Quality Architect	Generate Unit Test	生成单元测试
Component 视图	Open Specification		
	New	SubSystem	子系统
		Module	模块
		Module Diagram	模块图
		File	文件
		URL	链接
	Units	Load	装载
		Save	保存
		Save As	另存
		Unload	卸载
		Control Component View	控制组件视图
		Uncontrol	解除控制
		Write Protection	写保护
	Data Modeler	New	新建
		Transform to Data Model	转换成数据模型
		Transform to Object Model	转换成对象模型
		Compare and sync	比较和同步
	Quality Architect	Generate Unit Test	生成单元测试
Deployment	Open		打开
	Print		打印 108
	New	Processor	新建处理器
		Device	新建设备
		File	文件
		URL	链接
	Units	
		control	控制
		Uncontrol	解除控制
		Write Protection	写保护

利用浏览器，可以浏览每个视图中的模型元素、移动和编程模型元素、增加新的元素。通过在浏览器中右键单击元素，可以将文件或 URL 链接到元素、访问元素的详细规范、删除元素和更名元素。

浏览器组成树视图样式，每个模型元素可能包含其他元素。

默认情况下，浏览器出现在屏幕左上角。可以将浏览器移动到另一位置，也可以隐藏浏览器。

要显示或隐藏浏览器，操作如下：

- (1) 右击选择浏览器窗口的空白处；
- (2) 从弹出菜单中选择 Hide，即可显示或隐藏浏览器。

3. 文档窗口

文档窗口用于对模型元素添加说明。例如，可以在文档窗口中书写类的简要说明。

将文档加进类中时，在文档窗口中输入的一切都显示为所产生代码的说明语句，从而不必在今后输入系统代码的说明语句。文档还会在 Rose 产生的报表中出现。

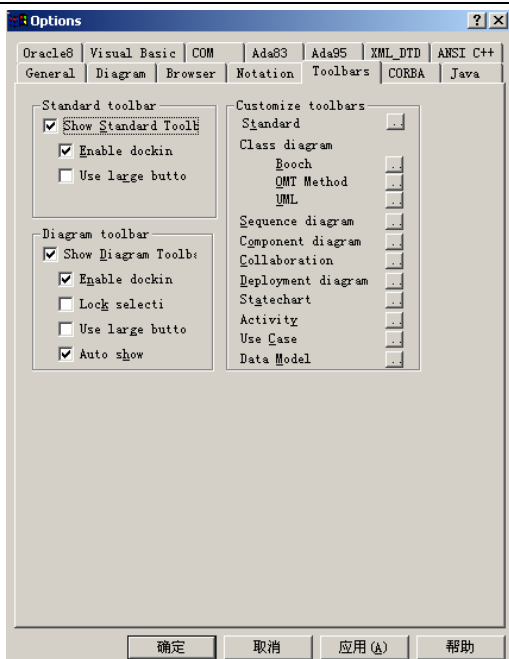
从浏览器或框图中选择不同的元素时，文档窗口自动更新显示所选元素的文档。

4. 工具栏

借用 Rose 工具栏，可以快速访问常用的命令。Rose 中有两个工具栏：标准工具栏(见图 2-18)和图工具栏。标准工具栏包括任何图中都可以使用的选项。图工具栏则随每种 UML 图



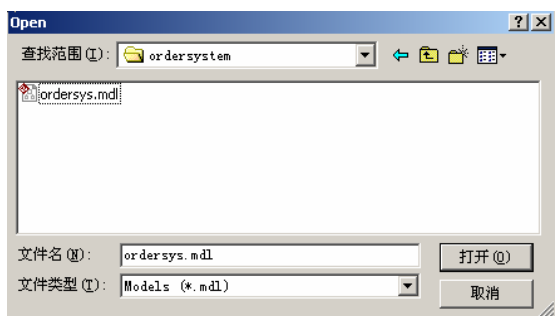
而改变。工具栏的信息可以修改，例如，在 Tools 子菜单中选择“Option...”，则弹出选项窗口(见图)，可以用来设置建模过程中的默认选项值。在选项窗口中点击“Toolbars”选项卡，将“Show Standard Tools”设置为不选，按下“确定”按键，



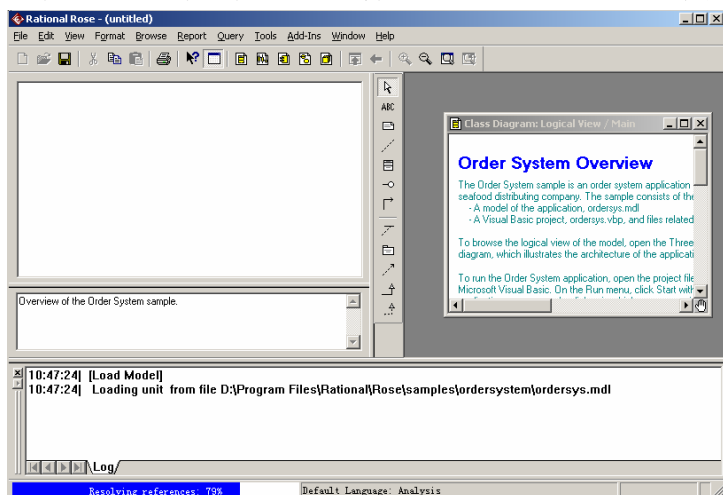
5. 图窗口

在图窗口中，可以浏览模型中的一个或几个 UML 图。改变图中的元素时，Rose 自动更新浏览器。同样，用浏览器改变元素时，Rose 自动更新相应的图。下面以 Rose 自带的例子来熟悉一下图窗口。

点击“File”子菜单中的菜单项“Open”，弹出“打开文件”对话框，在 Rational Rose2003 的安装文件夹中寻找 ordersys.mdl 文件(见图)。



按下“打开”按钮，选择 Use Case View。显示图 2-23 所示的画面。



6. 日志窗口

使用 Rose 模型时，有些信息会在日志窗口中显示。例如，生成代码时，生成的任何错误代码均会在日志窗口中显示。在打开 ordersys.mdl 时，日志窗口中显示如上图所示的信息。

3.4 Rose 中的基本操作

在使用 Rose 进行设计之前，介绍两种基本的操作：

- (1) 删除图元素；
- (2) 增加图元素。



在这里要区分从浏览器中删除元素还是从图中删除元素。当从浏览器中删除元素时，Rose 首先从模型中移去被选择的元素，修改所有出现被移去元素的图，从中删除被移去的元素，并删除元素的详细说明。当在一个图中删除选择的元素时，Rose 从当前图中移去选择元素，但是不修改模型中其他的图。

类似地，可以采用两种方式增加元素：在图中或在浏览器中。



附录5 UML 软件实验部分操作

学校网上选课系统的用例图的设计和实现

1、用例图设计实例

介绍学校的网上选课系统的用例图的设计和实现

2、需求

学校的网上选课系统主要包括如下功能：管理员通过系统管理界面进入，建立本学期要开的各种课程、将课程信息保存在数据库中并可以对课程进行改动和删除。学生通过客户机浏览器，根据学号和密码进入选课界面，在这里学生可以进行三种操作：查询已选课程、选课以及付费。同样，通过业务层，这些操作结果存入数据库中。

3、分析

本系统拟使用 Java 语言通过三层模型实现：数据核心层、业务逻辑层和接入层。其中，数据核心层包括对于数据库的操作；业务逻辑层作为中间层对用户输入进行逻辑处理，再映射到相应的数据层操作；而接入层包括用户界面，包括系统登录界面、管理界面、用户选课界面等。

本系统涉及的用户包括管理员(Registrar)和学生(student)，他们是用例图中的活动者，他们的主要特征相似，都具有姓名和学号等信息，所以可以抽象出“基”活动者 People，而 Registrar 和 Student 则从 People 统一派生。数据库管理系统是另外一个活动者。

4、事件流

下面是系统中出现的一些事件流。

添加课程事件流：

- (1) 管理员选择进入管理界面，用例开始。
- (2) 系统提示输入管理员密码
- (3) 管理员输入密码。
- (4) 系统验证密码。

A1：密码错误



(5) 进入管理界面，系统显示目前所建立的全部课程信息

(6) 管理员选择添加课程。

(7) 系统提示输入新课程信息

(8) 管理员输入信息。

(9) 系统验证是否和已有课程

A2: 有冲突

(10) 系统添加新课程，提示课程添加成功。

(11) 系统重新进入管理主界面，显示所有课程。

(12) 用例结束。

其他事件流：

A1: 密码错误

(1) 系统提示再次输入。

(2) 用户确认。

(3) 三次错误，拒绝再次访问。

(4) 否则进入添加课程事件流第 5 步。

A2: 有冲突

(1) 系统提示冲突，显示冲突课程信，

(2) 用户重新输入。

(3) 继续验证直到无冲突。

(4) 进入添加课程事件流第 10 步。

删除课程事件流和修改课程事件流与此类似。

选课事件流：

(1) 学生进入选课登录界面，用例开始。

(2) 系统提示输入学号和密码。

(3) 学生输入学号和密码。

(4) 系统验证。

A1: 验证失败

(5) 进入选课主界面。

(6) 学生点击选课。

(7) 系统显示所有课程信息。



(8) 学生选择课程。

(9) 系统验证课程是否可选。

A2: 不可选

(10) 系统提示课程选择成功, 提示学生交费。

(11) 用例结束。

错误流:

A1: 验证失败

(1) 系统提示验证失败, 提示重新输入。

(2) 三次失败, 拒绝访问。

(3) 成功, 转选课事件流第 5 步。

A2: 课程不可选

(1) 系统提示课程不可选及原因。

(2) 学生重新选课。

(3) 重新验证直至成功。

(4) 转选课事件流第 10 步。

因为付费方式多样, 所以在此将不讨论付费用例。查询事件流比较简单, 这里也不详细描述。

5、用例图

根据以上的分析, 可以绘制出如图 1 所示的系统用例图。

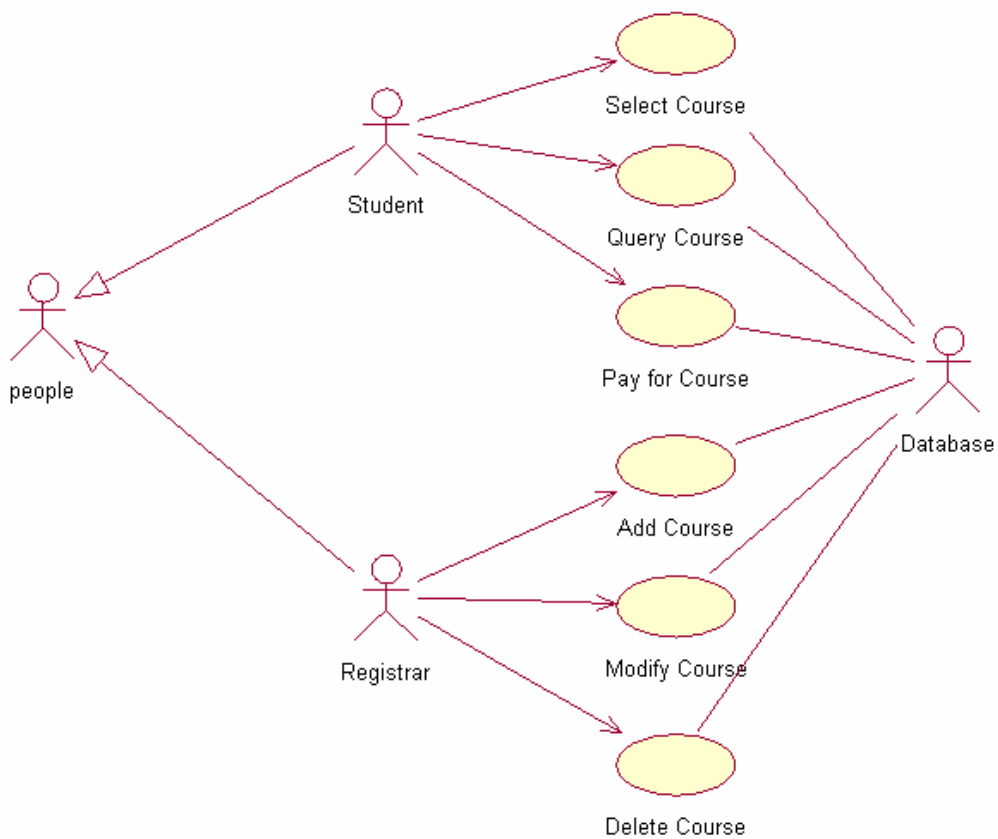


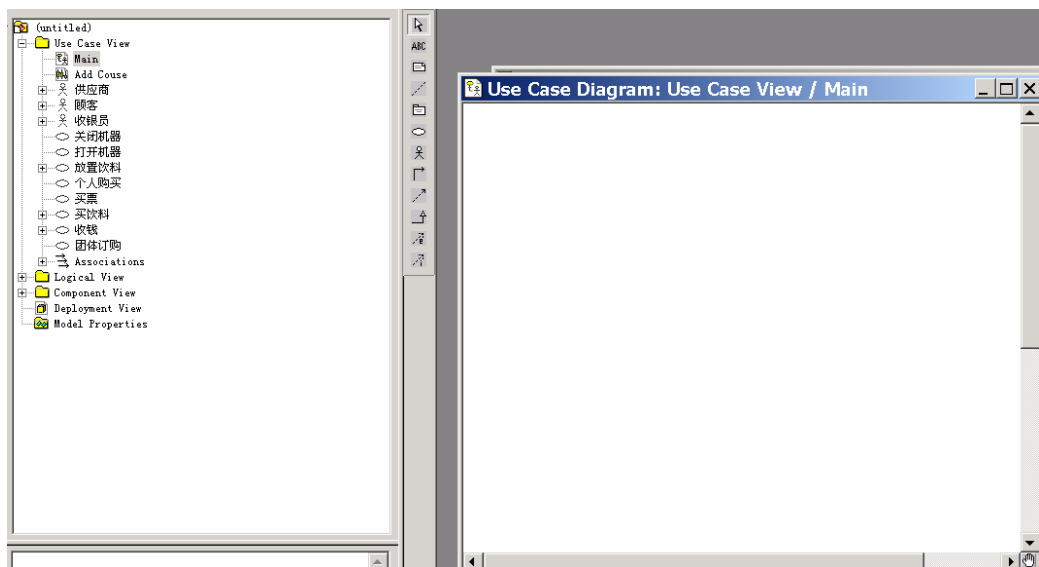
图 1

首先，在 Rose2003 中设计选课系统的用例图，具体步骤如下：

- (1) 点击浏览器中的 use case View 中的 Main，弹出用例图窗口如图 3。此时，与用例图窗口对应的工具栏如图 2 所示。



如图 2



如图 3

- (2) 将光标置于工具栏的 Actor 图标上，按下左键并将光标拖曳到用例图窗口上，放开左键，用例图窗口内出现一个活动的图标（见图 4），其名字为“NewClass”。

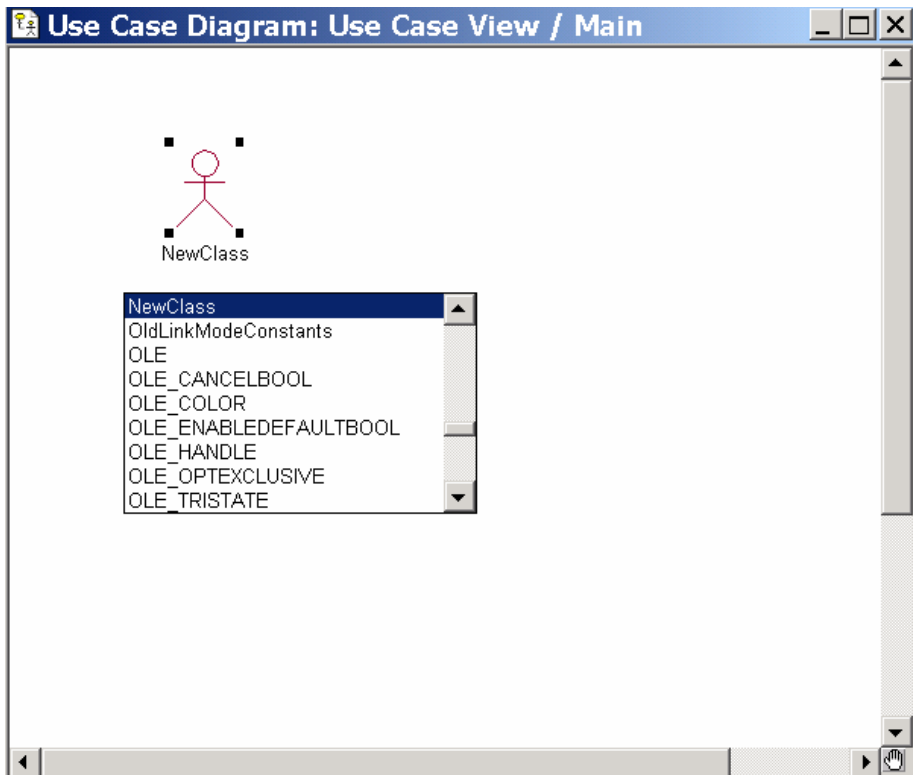


图 4

- (3) 修改元素的名字有如下两种方法：

①在用例图窗口中双击 NewClass 图标，弹出如图 5 所示的窗口。然后再选择“General”选项卡，将 Name 修改为“People”，最后按下“确认”按钮。

②如图 4 所示，在用例图窗口中将光标置于“NewClass”处，直接将其修改为“People”。

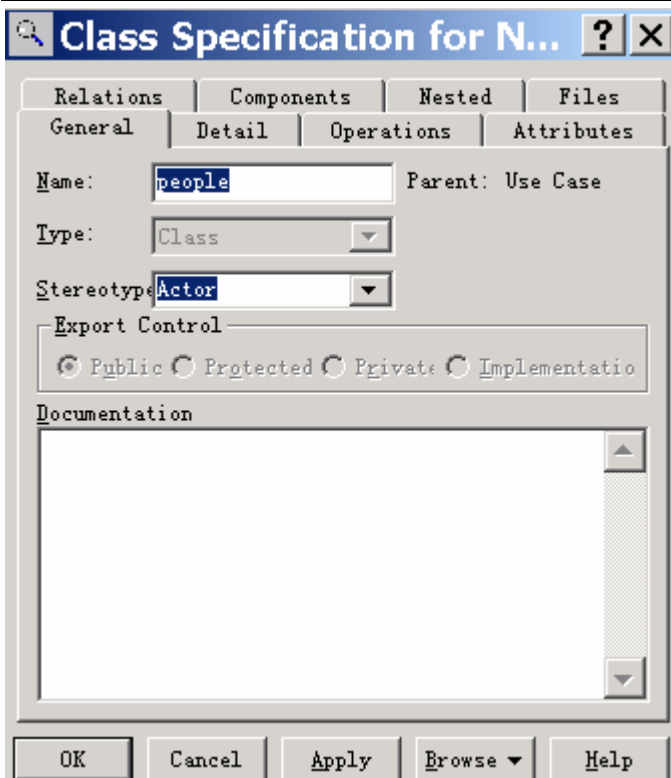


图 5

- (3) 采用同样的方法，在用例图中添加 Registrar 和 Student 图标，点击用例窗口工具栏中的图标“generalization”，在用例图窗口中，将光标从 Student 移动到 People，在 Student 与 People 之间添加概括(generalization)关系(见图 6)。

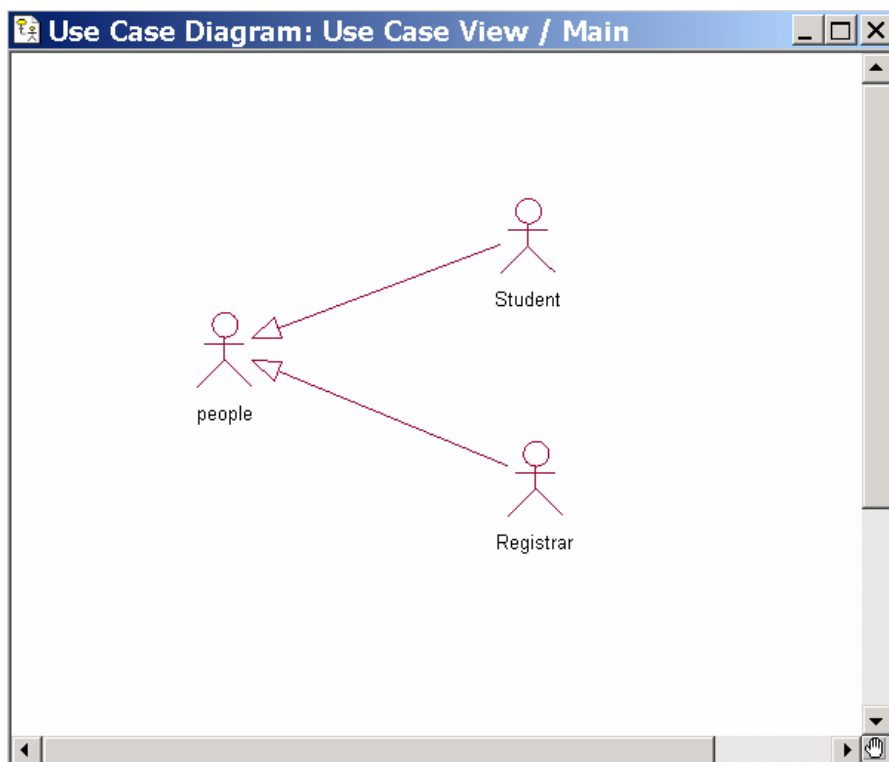


图 6

(4) 同样在 Registrar 和 People 之间添加概括关系。

(5) 点击工具栏中 Use Case 的图标，将光标移到用例图窗口，窗口内显示用例的椭圆图标，采用和 (3) 同样的方法，将图标名称改成“SelectCourse”。添加 Select Course 用例，点击工具栏中的 Undirectional Relation 图标，将光标从 Student 指向 Select Course，在 Student 和 Select Course 之间添加关系(见图 7)。

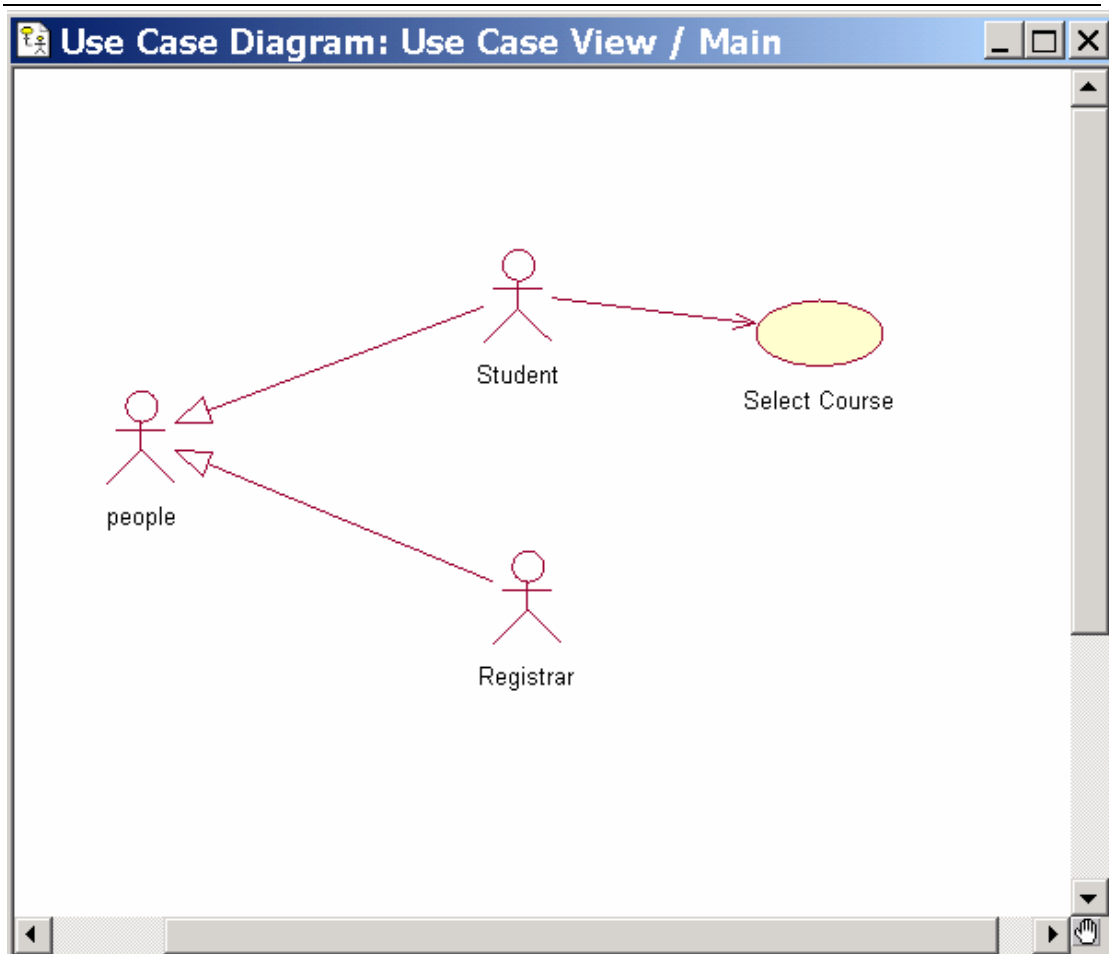


图 7

(9) 重复以上步骤，完成所示的用例图。

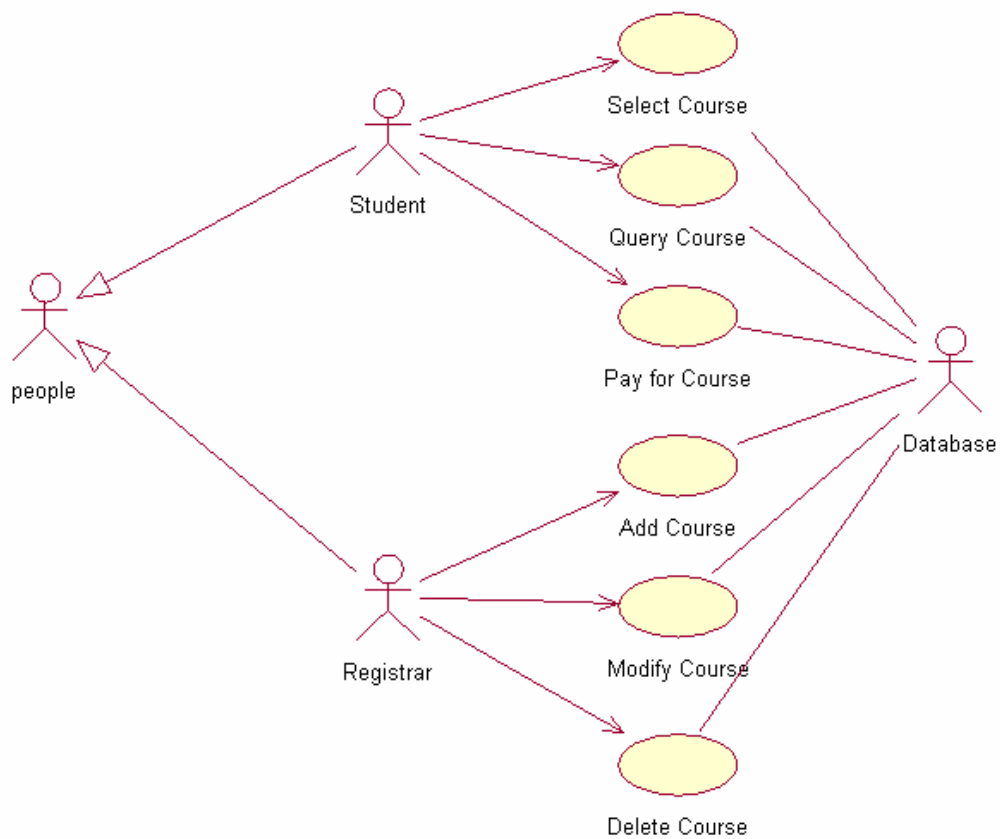


图 8

(10) 然后再指定活动者和用例的属性。在浏览器中选择活动者 People, 按下鼠标右键, 在弹出的菜单中选择 “Open Specification”, 弹出图 9 所示的对话框。

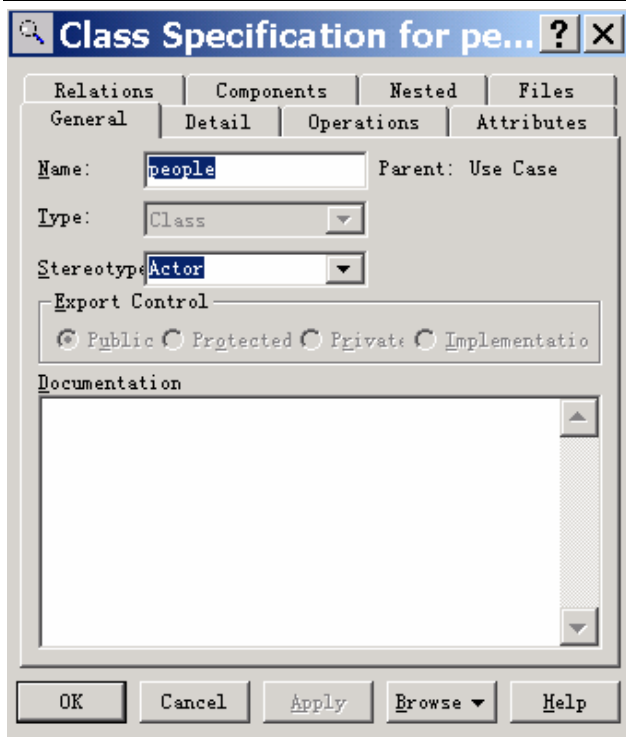


图 9

(11) 从图 9 中可以看到 People 的版型 (Stereotype) 是 Actor (以后还将介绍这个画面)。

(12) 双击 Select Course 用例, 弹出 Use Case Specification 对话框 (见图 10)。

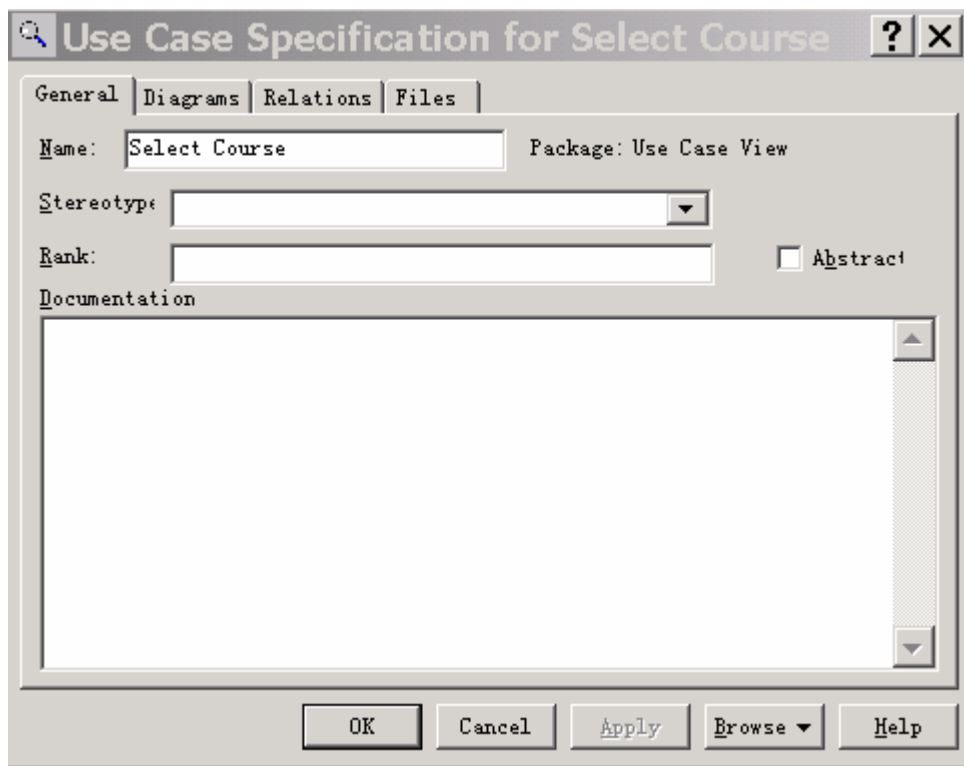


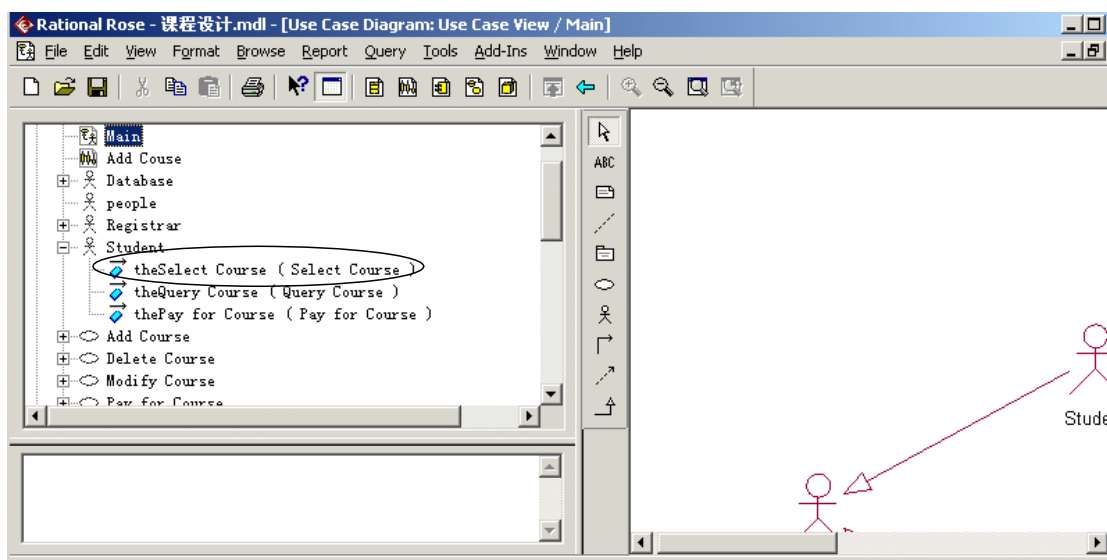
图 10

(13) 在 Documentation 文本框中可以书写该用例的事件流(见图 11)



图 11

(14) 完成以上步骤，可以看到 Rose 在浏览器中自动标出了各种关系。图 12 中使用椭圆将它们标注出来。



6、小结

用例描述用户使用系统时从头到尾的一系列事件，它展现和暗示了系统的需求。UML中使用用例图来组织用例。

本实验介绍了用例图及其所涉及的概念，包括：

- (1) 用例。
- (2) 活动者
- (3) 事件流
- (4) 用例和用例之间、用例与活动者之间、活动者与活动者之间的关系。
- (5) 最后，还介绍了在 Rose 中用例图的创建方法。



Add Course（添加课程）用例的活动图设计和实现

利用文本描述事件流是很有用的，但如果事件流的逻辑复杂且有许多其他事件流，则文本形式可能较难阅读和理解。这时可以使用流程图来描述事件流，在 UML 中这种图称为活动图，它是事件流的另一种建模方式。

因为活动图显示的信息与文本事件流相同，所以可在业务模型中用活动图描述业务过程的工作流。本实验介绍如何创建活动图。

1. 活动图的基本概念和组成

在用例模型中，活动图用来捕捉用例的活动，使用框图的方式显示动作及其结果。活动图着重描述操作 (Operation) 以及用例实例或对象中的活动。

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 (对象状态改变)、何时发生 (动作序列) 以及在何处发生 (泳道)。活动图可以用作下述目的：

- 1) 描述一个操作执行过程中所完成的工作 (动作)，这是活动图最常见的用途。
- 2) 描述对象内部的工作。
- 3) 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。
- 4) 显示用例的实例如何执行动作以及如何改变对象状态。
- 5) 说明一次商务活动中的人 (角色)、工作流组织和对象是如何工作的。

活动图由起始状态 (start state)、终止状态 (end state)、状态转移 (state transition)、决策 (decision)、守护条件、同步棒 (synchronization bar) 和泳道 (swimlane) 组成。

起始状态显式地表示活动图上一个工作流程的开始，用实心圆点来表示 (见图 1)。在

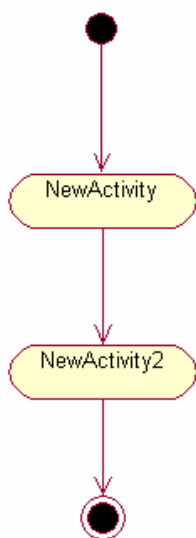


图 1

一个活动图中，只有一个起始状态。终止状态表示了一个活动图的最后和终结状态，一个活动图中可以有 0 个或多个终止状态，终止状态用实心圆点外加一个小圆圈来表示（见图 1）。

活动图中的动作用一个圆角四边形表示，其内部的文本串用来说明采取的动作，动作之间的转移用箭头来表示，称为转移，用带有箭头的实线表示。箭头上可能还带有守护条件、发送短句和动作表达式，守护条件用来约束转移，守护条件为真时转移才可以开始。用菱形符号来表示决策点，决策符号可以有一个或多个进入转移，两个或更多的带有守护条件的发出转移。

可以将一个转移分解成两个或更多的转移，从而导致并发的动作。所有的并行转移在合并之前必须被执行。一条粗黑线表示将转移分解成多个分支，同样用粗黑线来表示分支的合并，粗黑线称为同步棒（见图 2）。

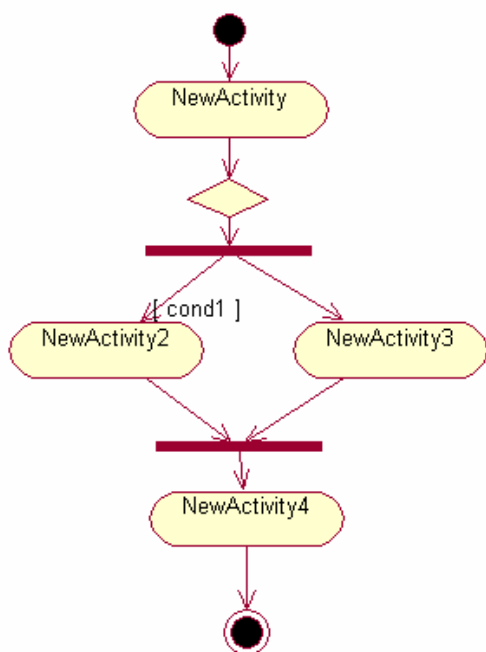


图 2

泳道分割活动图，有助于更好地理解执行活动的场所。

下面采用活动图来描述选课系统中的 Add Course 用例的工作流

2. 实例

本实验主要是通过实际制作选课系统中的 Add Course（添加课程）用例来学习设计和制作活动图。

2. 1 分析

在用例图中分析过，Add Course 用例的事件流为：

- (1) 管理员选择进入管理界面，用例开始。
- (2) 系统提示输入管理员密码。
- (3) 管理员输入密码。
- (4) 系统验证密码。

A1：密码错误



(5) 进入管理界面，系统显示目前所建立的全部课程信息。

(6) 管理员选择添加课程。

(7) 系统提示输入新课程信息。

(8) 管理员输入信息。

(9) 系统验证是否和已有课程冲突。

A2: 有冲突

(10) 系统添加新课程，提示课程添加成功。

(11) 系统重新进入管理主界面，显示所有课程。

(12) 用例结束。

因为管理员密码验证过程可以抽取出来，作为通用的流程，所以这里对 Add Course 事件流稍作修改，将管理员输入课程信息作为起始的活动，内容如下：

(1) 管理员输入信息。

(2) 系统验证是否和已有课程冲突。

A2: 有冲突

(3) 系统添加新课程，提示课程添加成功。

(4) 系统重新进入管理主界面，显示所有课程。

(5) 用例结束。

根据以上描述并进一步细化，画出图 3 所示的 Add Course 用例的活动图。

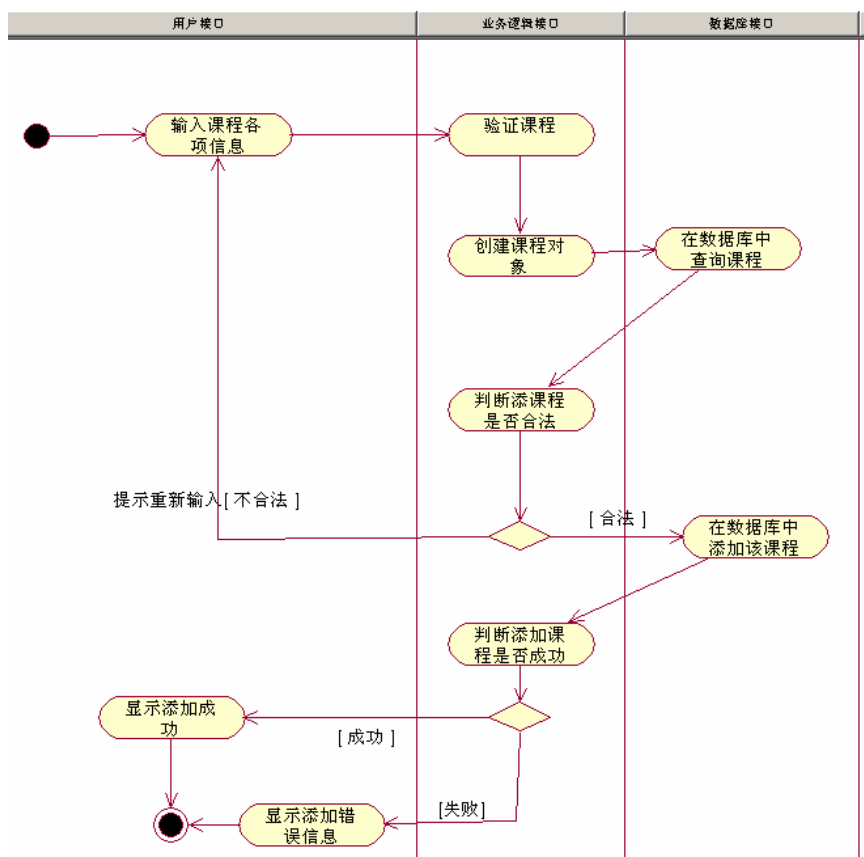


图 3

2. 2 创建 Add Course 活动图

创建 Add Course 活动图的步骤如下：

(1) 在浏览器中，使用鼠标右键点击 Use Case View，在弹出的菜单中选择“New” → “Activity Diagram”，则在 Use Case View 中添加一项“State / Activity Model”，该项产生一个名为“NewDiagram”的活动图，将其改为“Add Course”，

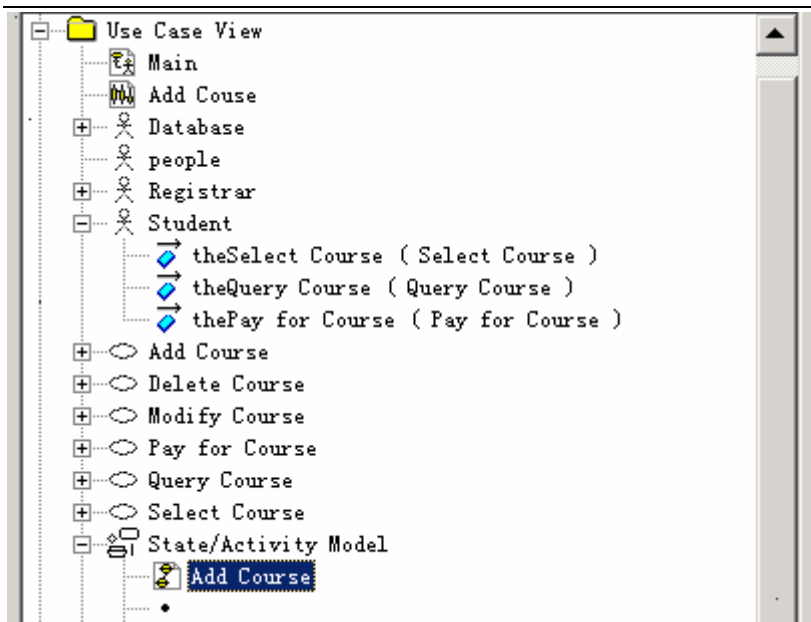


图 4

(2) 使用鼠标双击活动图“Add Course”，弹出如图 5 所示的活动图窗口。

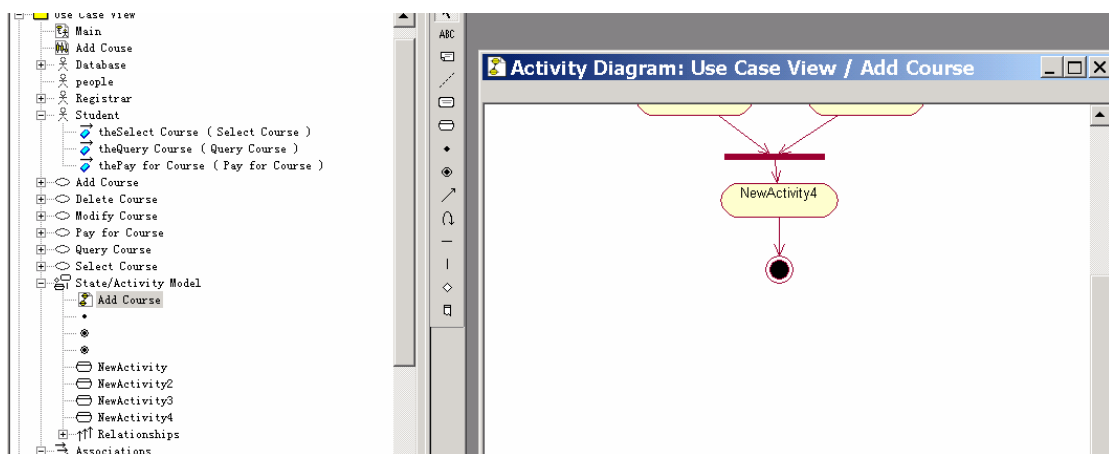


图 5

(4) 点击工具栏的“Swimlanc”工具，在右边活动图窗口点击鼠标左键，即可增加一个新



的泳道“NewSwimlane”，同时在左边的活动图“Add Course”之下，也出现一个泳道“NewSwimlane”标识,将泳道名称修改为“用户接口”，如图6所示。

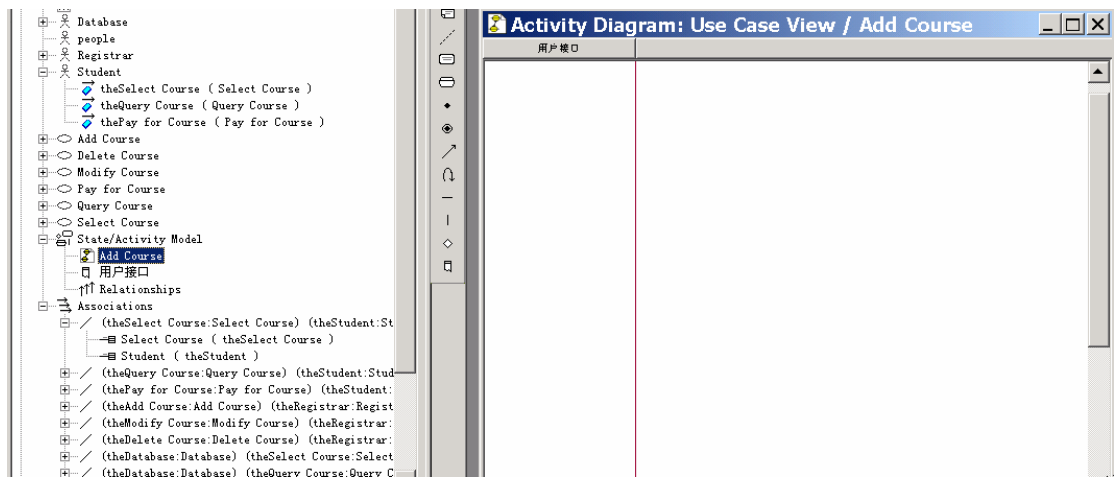


图6

(7) 使用同样的方法增加“业务逻辑接口”和“数据库接口”泳道，结果如图7所示。

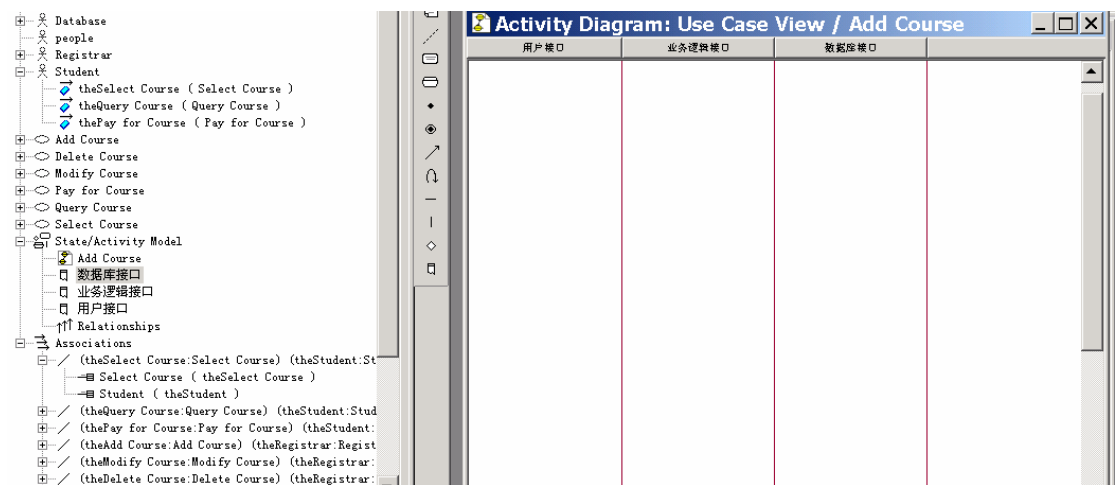


图7

(8) 在工具栏中选择“Start State”图标，放置到“用户接口”泳道内，选择工具栏



中的“Activity”图标，在“用户接口”泳道内增加一个新的活动。将活动的名称修改为“输入课程各项信息”。

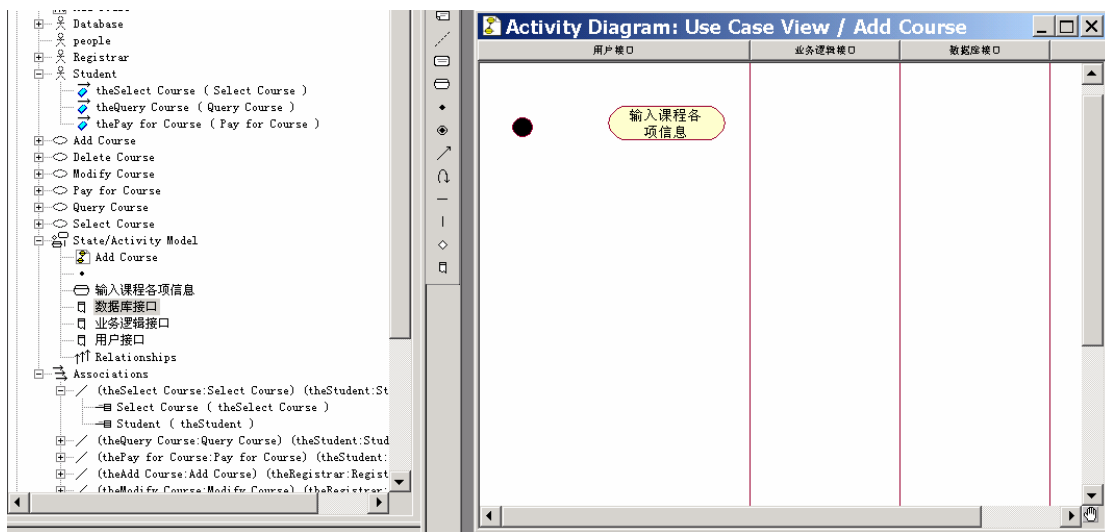


图 8

(12)在工具栏中选择“Transition”图标，在活动图窗口中，将光标从起始状态到“输入课程各项信息”，则从起始状态到“输入课程各项信息”之间添加了一条带箭头的实

这就是转移，输入课程以后，还要判断输入的课程是否合法，即需要进行课程验证。这个工作要提交到“业务逻辑接口”中进行。采用前述的方法，在“业务逻辑接口”泳道中添加“验证课程”活动，并在“输入课程各项信息”和“验证课程”之间添加转移

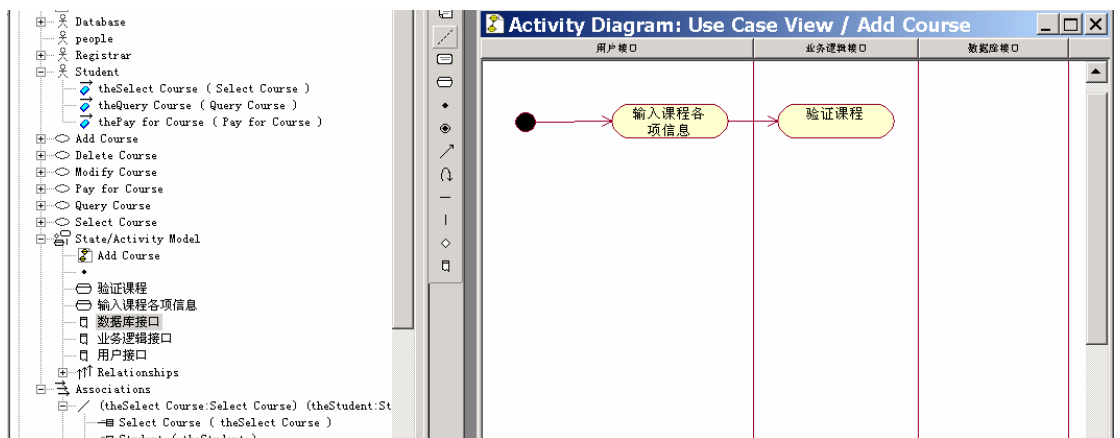


图 9 添加“验证课程”活动和相关转移

(14) 为了进行课程信息的验证，需要在业务逻辑接口中依据已有的课程信息，创建课程对象，提交到数据库中，进行信息的验证。添加“创建课程对象”活动，依据课程对象的信息，在数据库中查找课程信息。为了完成这个功能，在“数据库接口”泳道中添加“在数据库中查询课程”活动，并在“创建课程对象”和“在数据库中查询课程”之间添加转移关系(见图 10)。

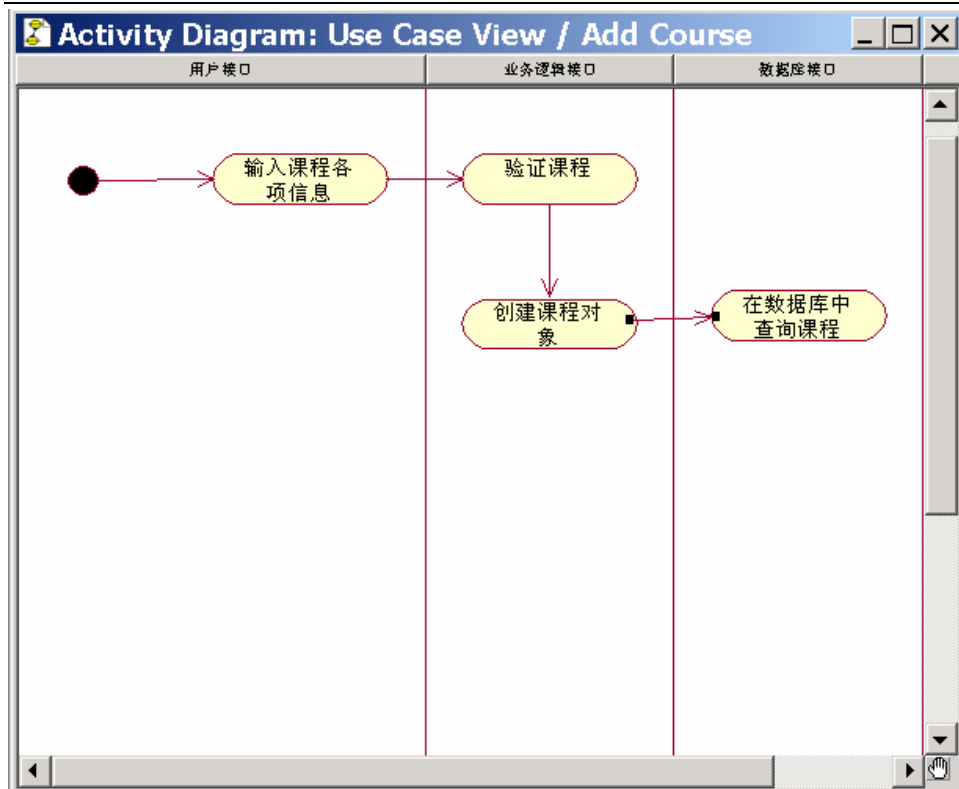


图 10

(16) 查询结果将被返回到业务逻辑接口，由业务逻辑接口判断查询结果的信息，以确认

添加的课程是否合法。被添加的课程有可能是合法的，也有可能是不合法的，所以要增加一个决策(decision)进行判断。将光标移到活动图窗口的工具栏上，选择“Decision”图标，然后将光标移动到活动图窗口的“业务逻辑接口”泳道，按下鼠标键添加一个决策，在“判断添课程是否合法”和决策之间添加转移(见图 11)。

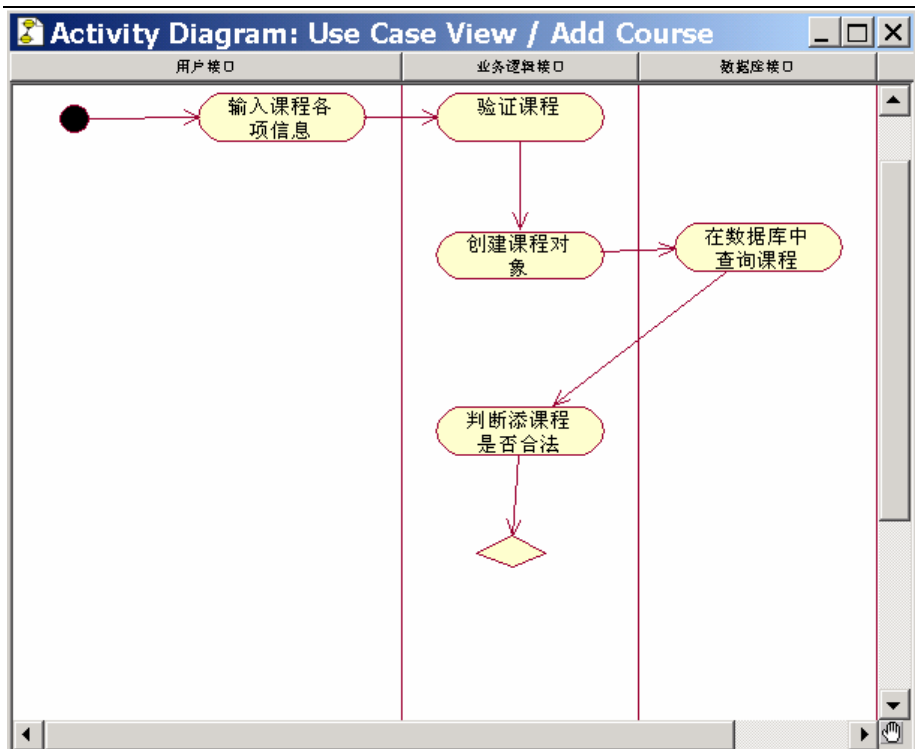


图 11

(18)如果输入的信息合法，则将在数据库中添加输入的课程信息。如果不合法，则提示重新输入课程信息。在“数据库接口”泳道中添加“在数据库中添加该课程”活动。在决策标记和“在数据库中添加该课程”活动之间添加转移。这个转移发生的条件是课程合法，所以，要在转移上二添加守护条件“合法”。添加守护条件的方法是：双击刚才添加的转移，在弹出的对话框中选择“Detail”选项卡，在“Guard condition”栏中添转移条件“合法”（见图 12）。

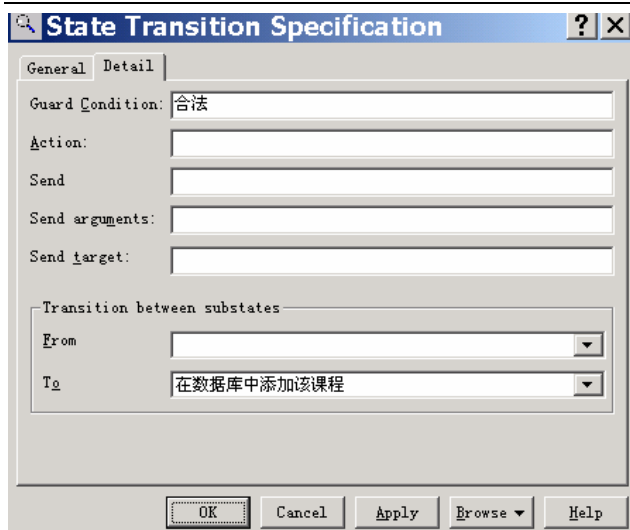


图 12

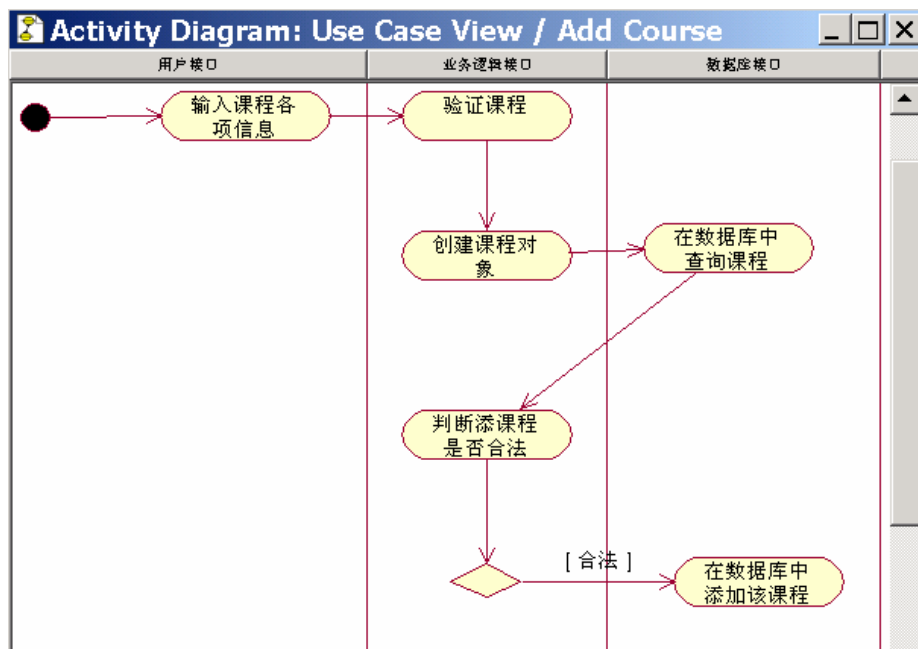


图 13

(20) 在图所示的对话框中按下“OK”按钮，则活动图可以看到，从决策到“在数据库中 添加该课程”的转移上增加了守护条件“合法”。

(21) 如果课程信息不合法，则提示重新输入，重新转入“输入课程各项信息”活动。在决策和“输入课程各项信息”之间添加转移，守护条件是“不合法”（见图 14）。注意到在图中标记的转移上，标有“提示重新输入”的文字，这是在“stateTransition Specification”对话框中设置“Event”项得到的。

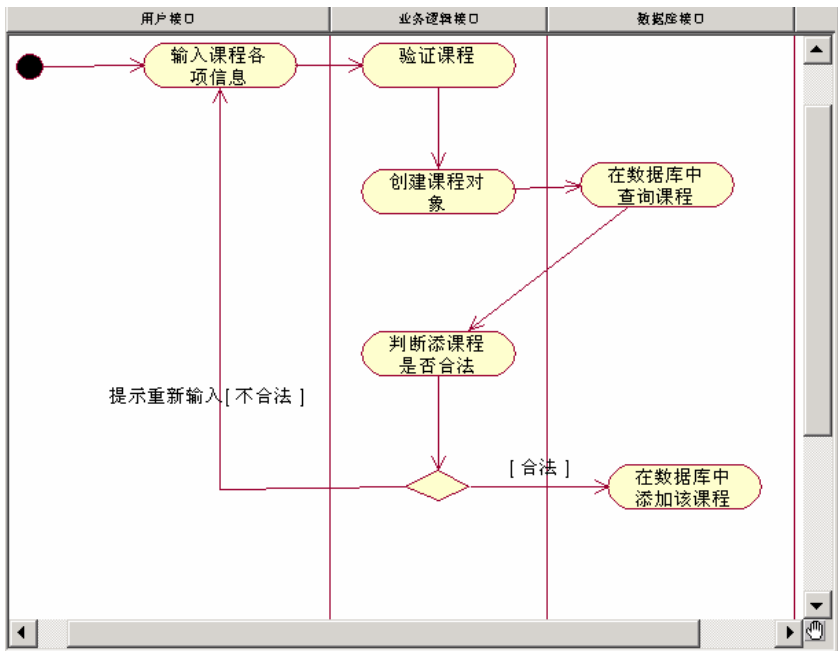


图 14

(23) 如果课程信息合法，则在数据库中添加课程，然后判断添加课程是否成功。增加一个决策，如果添加课程成功，则显示添加成功信息，过程结束（见图 15）。

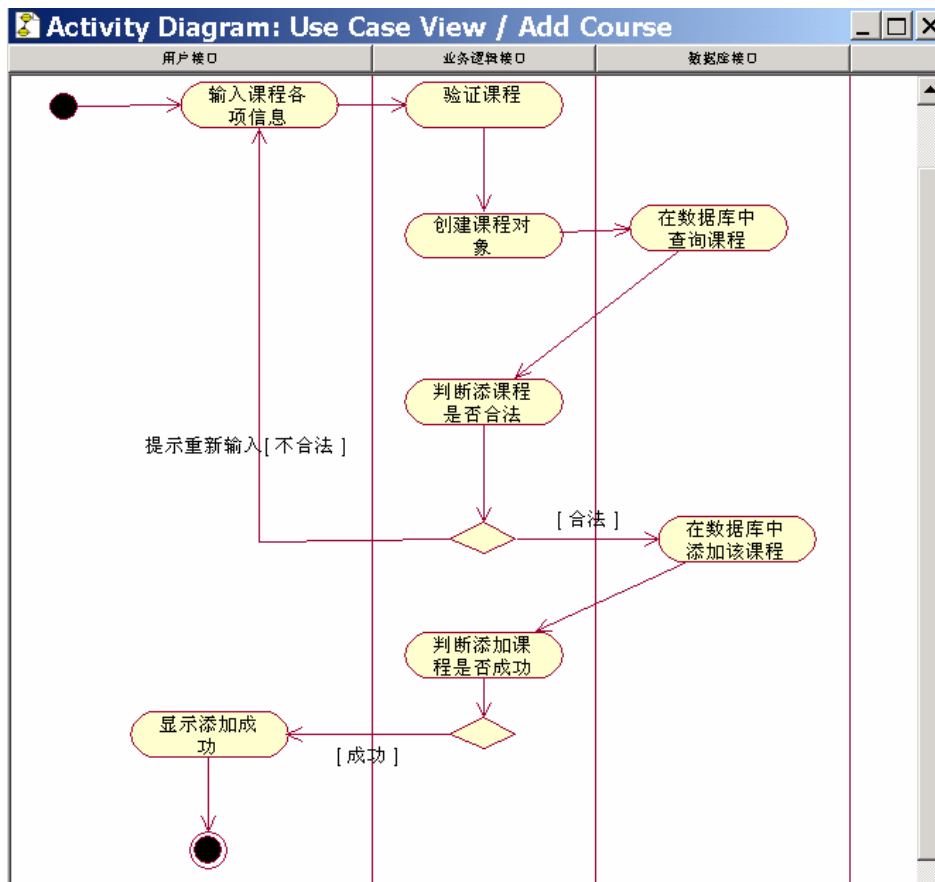


图 15

(24) 如果添加课程失败，则显示添加错误信息，过程结束(见图 16)。到此为止，一个完整的活动图完成了。

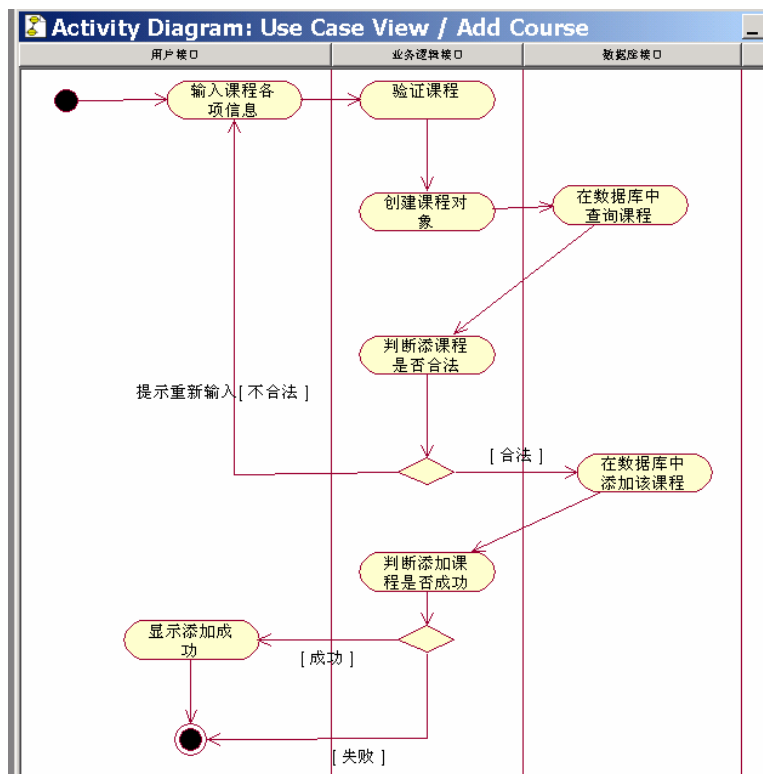


图 16

25) 双击决策，弹出“Decision specification”对话框，点击“Transl ‘tions”选项卡，可以看到与决策相关的转移关系(见图 17)。

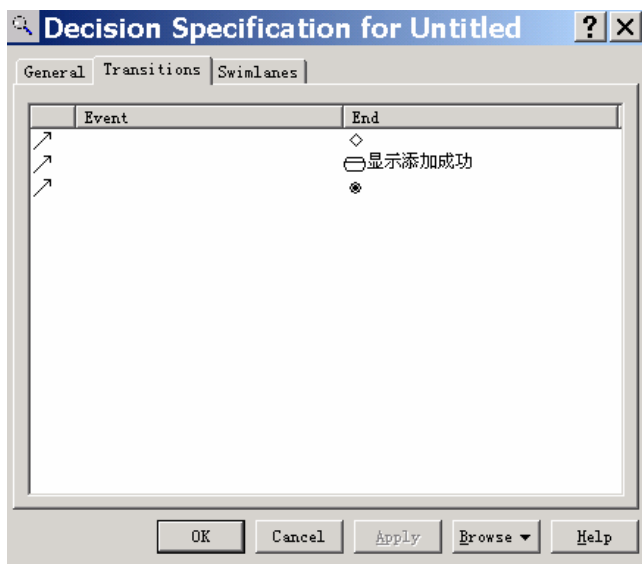


图 17

当然，也可以给决策命名。只要选择“Decision specification”对话框的“General”选项卡，修改其中的“Name”项的内容即可。

(26)完成整个活动图后可以看到，浏览器也发生了相应的变化：浏览器中列出了活动图模型的元素(见图 18)。

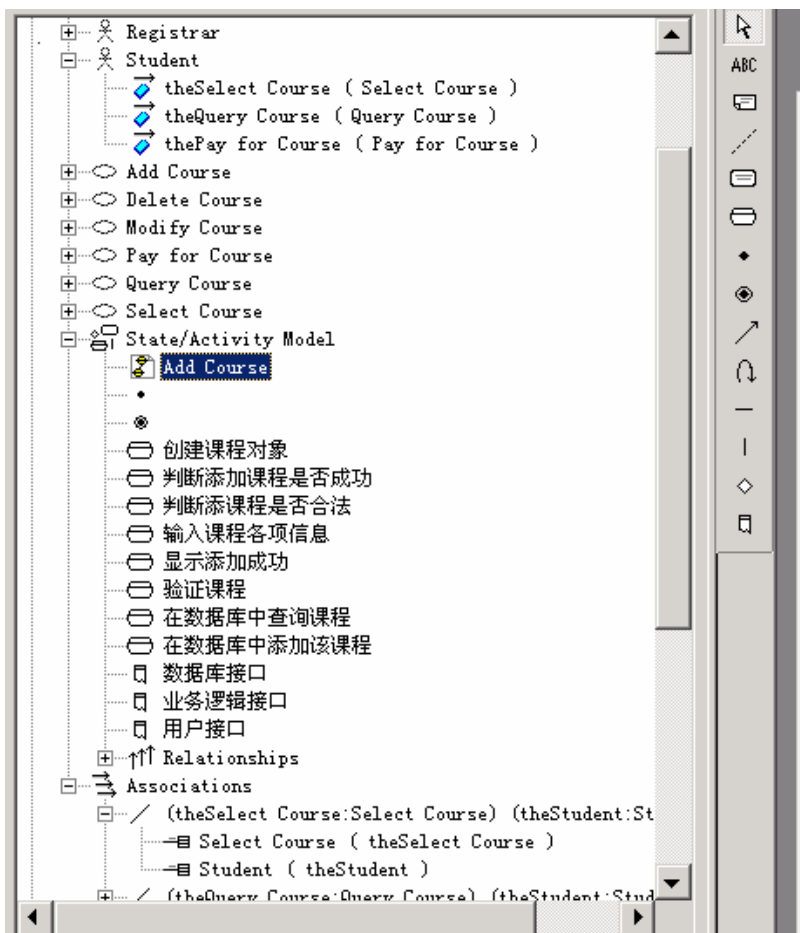


图 18

3 小结

活动图是另一种建模事件流的方式。利用文本描述事件流很有用，但如果事件流逻辑复杂且有許多其他事件流，或客户对文本不像对框图那么熟悉，则文本形式可能较难阅读和理解。

活动图显示与文本事件流相同的信息。活动图是描述工作流的另一种方式，描述采取何



种动作、做什么(对象状态改变)、何时发生(动作序列)以及在何处发生(泳道)。

活动图由起始状态、终止状态、状态转移、决策、守护条件、同步棒和泳道组成。通过本实验的学习,要求掌握以下内容:

- (1)活动图的基本概念和组成。
- (2)使用 Rose 创建活动图。

选课用例(select Course)的序列图和协作图设计和实现。

一旦定义了一个工程的用例,就可以用它们来指导对系统的进一步开发。用例的实现描述了相互影响的对象的集合,这些对象将支持用例所要求的功能。系统用例的实现,是从外部给视图转到内部结构的第一步。

UML 中,用例的实现用交互图(interaction diagram)来指定和说明。交互图通过显示对象之间的关系和对象之间处理的消息来对系统的动态特性建模。

有两种交互图:序列图(sequence diagram)和协作图(collabor diagram)。本实验介绍序列图和协作图及其实现。

1. 实例

本实验主要以选课系统中的选课用例(select Course)为例,来学习序列图和协作图的设计和实现。

2. 分析

为了使问题更简单一些,不考虑学生的登录。假设学生已经成功登录系统,选课的事件流如下:

- (1)学生进入选课主界面。
- (2)学生点击选课。
- (3)系统显示所有课程信息。
- (4)学生选择课程。
- (5)系统验证课程是否可选。
- A1: 课程不可选 .
- (6)系统提示课程选择成功,提示学生交费。



(7)用例结束。

A1 课程不可选

(1)系统提示课程不可选及原因。

(2)学生重新选课。

(3)重新验证直至成功。

(4)转选课事件流第 6 步。

序列图描述的是对象与对象的交互，创建序列图的步骤如下：

(1)寻找对象。

(2)寻找角色。

(3)将消息加进图中。

首先，查找 Select Course 用例的对象。从事件流中发现涉及以下对象：

(1)界面。

(2)课程。

(3)对于业务层的操作，也应该有对象进行处理。

(4)事件流中设计的角色有：学生、数据库。

然后，分析对象、角色之间交互的消息。本用例主要有以下交互：

(1)学生通过界面发送选课命令。

(2)界面向控制对象请求课程信息。

(3)控制对象向数据库发送查询数据消息。

(4)控制对象暂存数据库的查询结果。

(5)界面对象从控制对象中取得所有的课程信息。

(6)在界面上显示所有的课程信息。

(7)界面对象发送命令要求控制对象删除课程信息。

(8)学生选择课程。

(9)界面对象要求学生输入学号。

(10)界面对象向控制对象发送信息，查询该生是否可以选定的课程。

(11)控制对象从数据库中查询关联信息。

(12)控制对象判断是否可以选课。

[13]如果可以选课，则向数据库中添加关联信息。

(14)向界面对象返回信息。



根据以上分析，整理的最终 Select Course 序列图如图 1 所示。

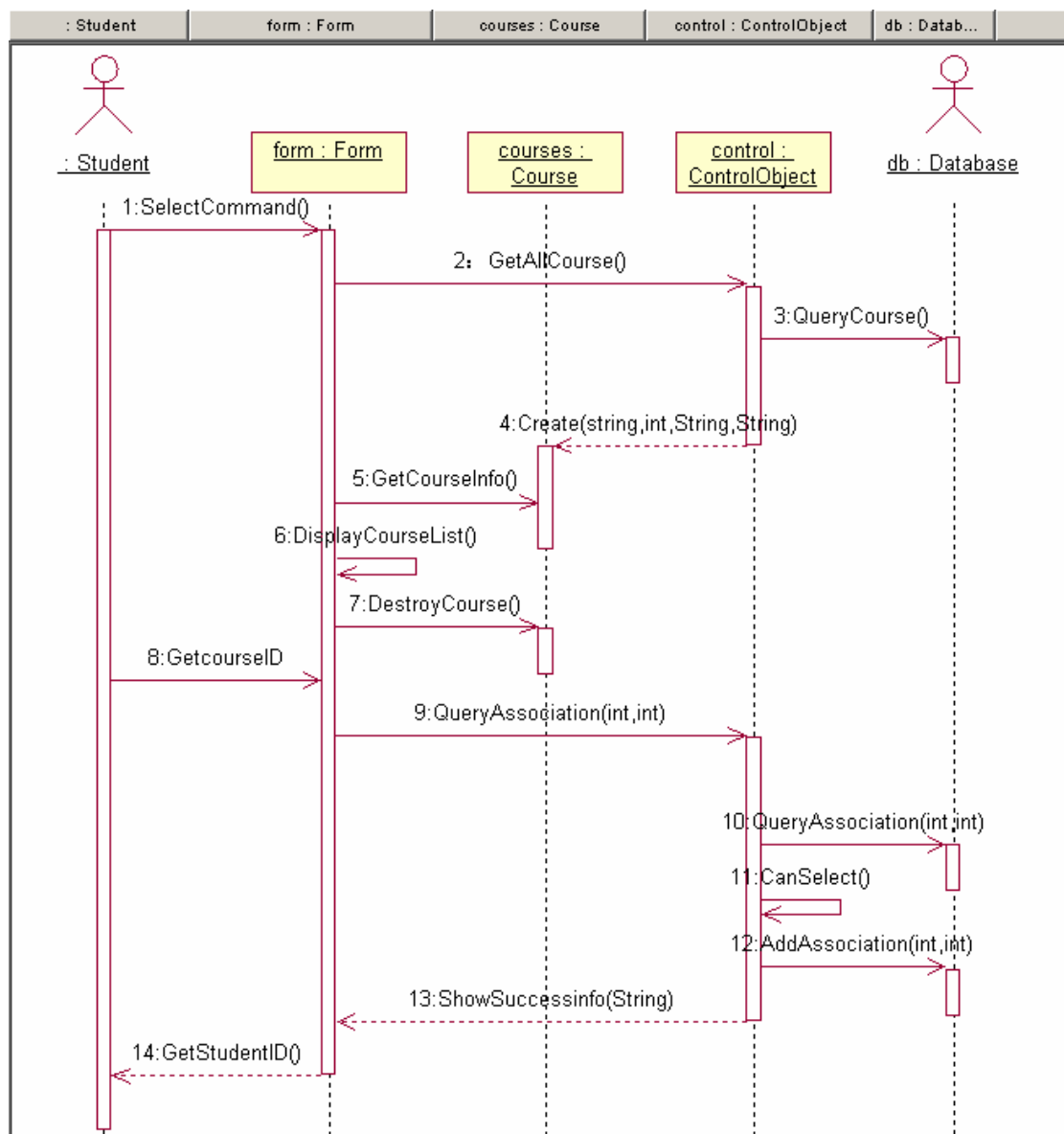


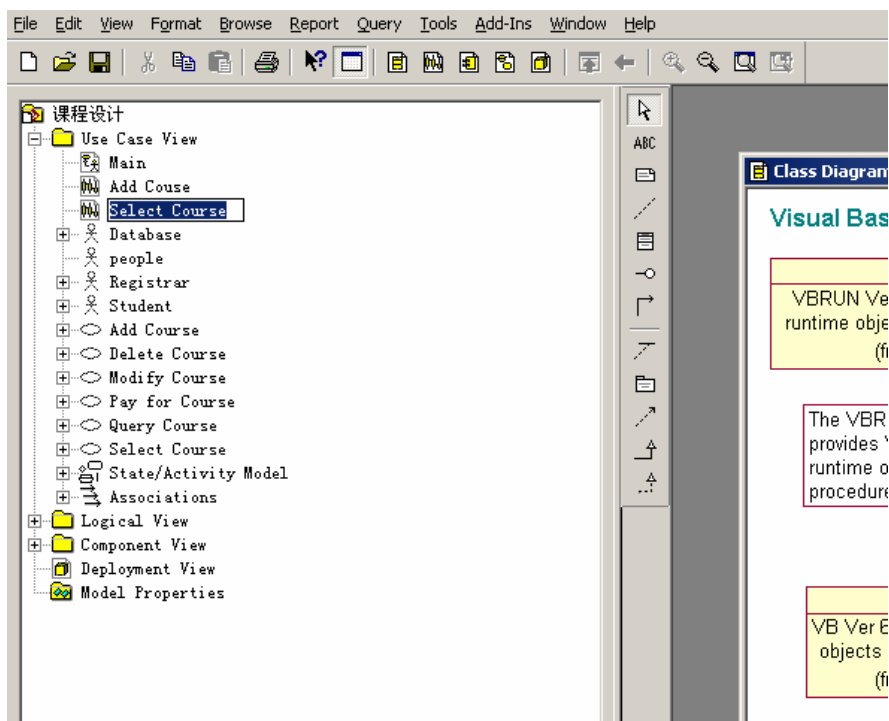


图 1 Select Course 序列图

3. Select Course

创建 Select Course 序列图的具体步骤如下：

(1) 在浏览器中右键点出“Use Case View”，在弹出的菜单中选择“New”，再在下一级菜单中“Sequence Diagram”，则 Use Case View 中显示一个新创建的序列图的图标，名字是 New Diagram，如图 2 所示，将该图的名字改成 Select Course。



(2) 在浏览器中选择一个序列图时，序列图窗口工具变成如图 3 所示的形式。



(3)在浏览器中选择角色 Student，将其从浏览器中拖到序列图中(见图 4)，图窗口中显示角色：Student 和泳道：Student，：Student 对象下有虚线条。

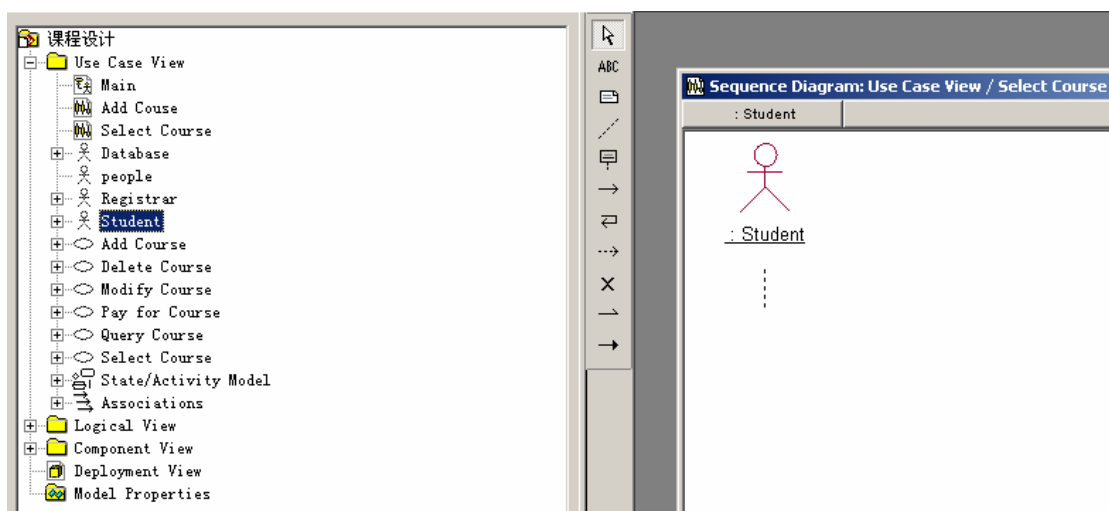


图 4 向序列图窗口中添加角色

(4)选择 Create a Object 具栏按钮，光标变成十字形状，将光标移到图窗口中，按下鼠标左键，则序列图窗口中添加了一个无名对象，窗口的顶部也出现了一个无名的冰道(见图 5)。

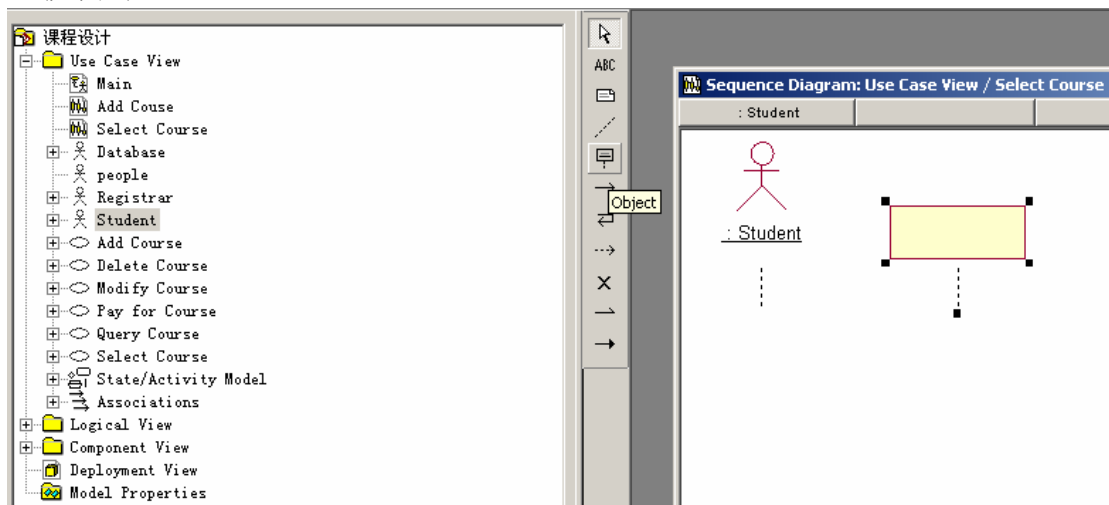


图 5 添加无名对象



(5) 选择新创建的对象，按下鼠标右键，在弹出的菜单中选择“Open specification”，弹出“Object Specification”对话框，选择对象的类，这是个界面对象，这里选择 FormView

结果如图 6 所示。

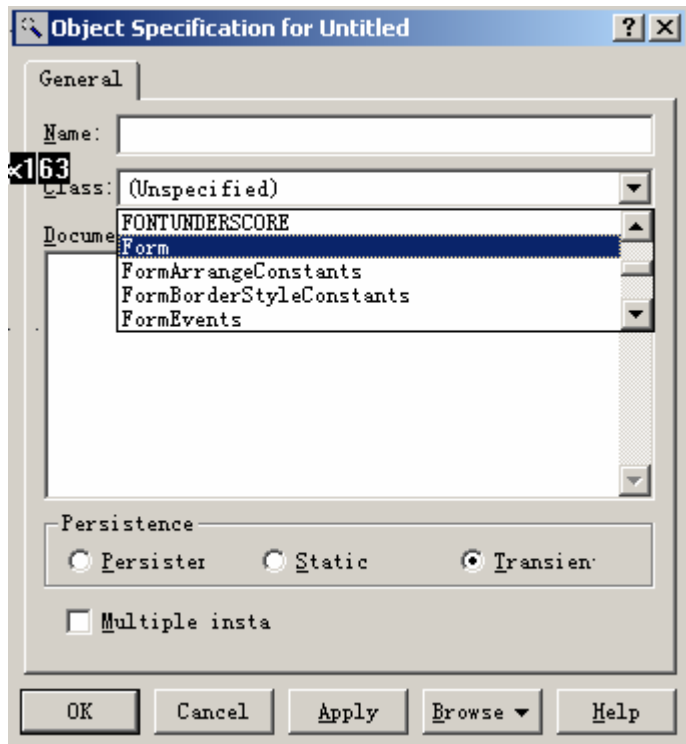
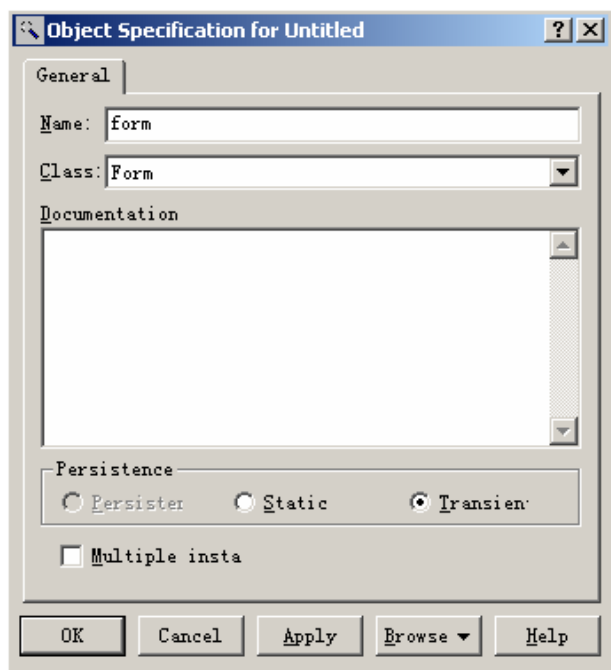


图 6 选择界面对象的类

如图 7 所示，将这个对象命名为 Form。



(7) 选择对象类且输入对象名称后，按下“OK”按钮，序列图中显示已经命名的标有类的对象。

(8) 用同样的方法在序列图中添加对象 courses 和 control，添加活动者 Database。添加后的结果如图 8 所示。

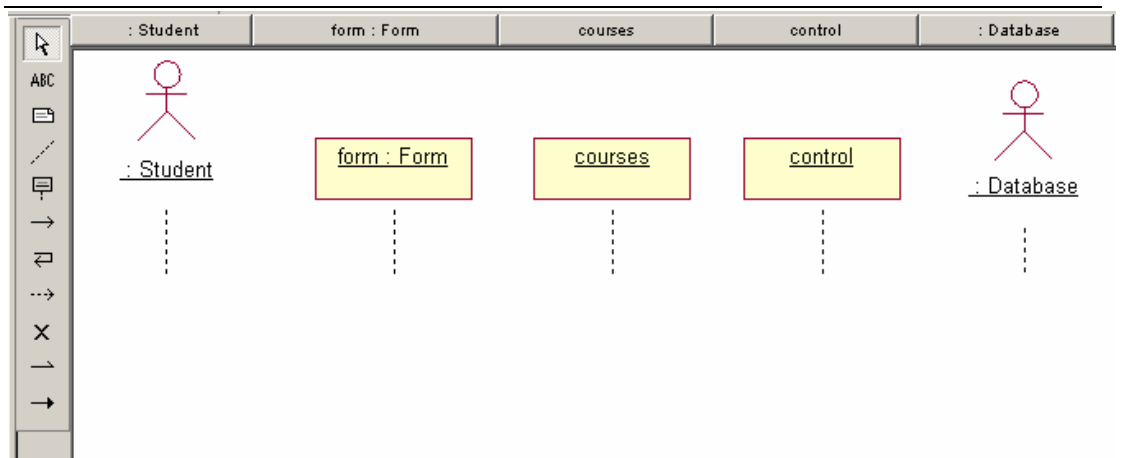
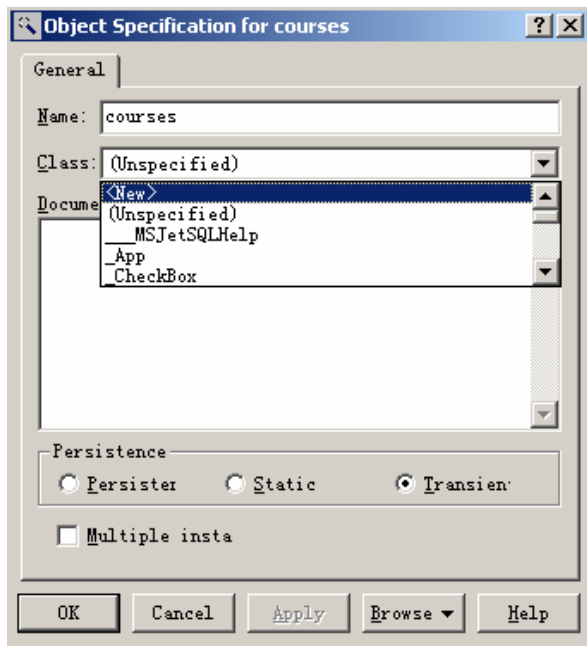


图 8

(9) 注意到图 8 中 courses、control 都没有类, 为了给它们添加类信息, 可以双击 courses 的图标, 在弹出的“Object Specification”对话框的 Class 下拉列表中选择<New>项。

图

图 9 给出操作示意图。



(10) 选择<New>后按下“OK”按钮，弹出如图 10 所示的“Class Specification”对话框，然后在 Name 文本框中输入“Course”。

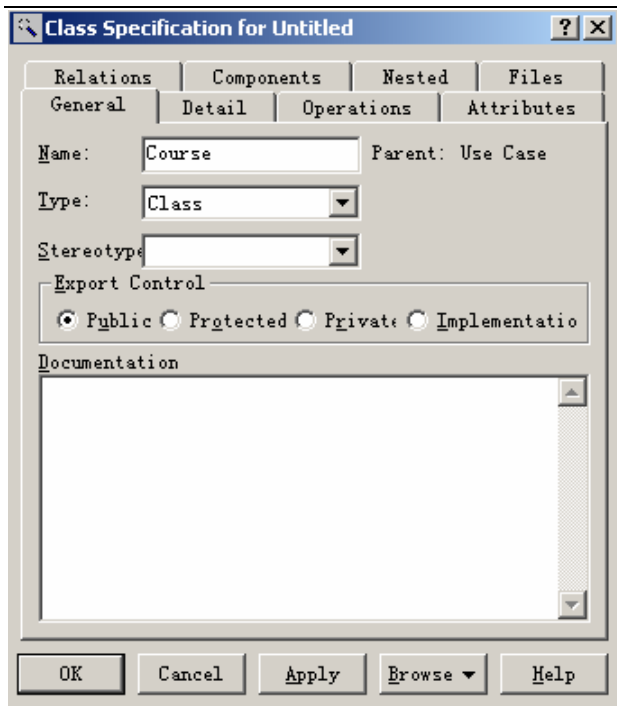
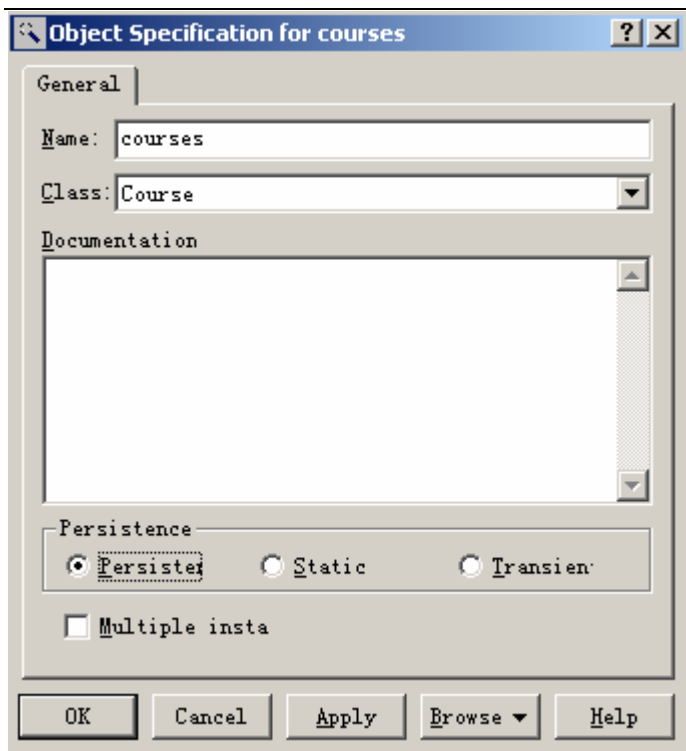


图 10 建立新的类

(11)在对话框中按下“OK”按钮，返回到“Object Specification”对话框。注意到在这个对话框中，courses 的类已经修改成 Course(见图 11)，将 courses 的 Persistence 属性设置成 Persistent。



12) 在图 11 的对话框中按下“OK”按钮，返回序列图窗口（见图 12）。

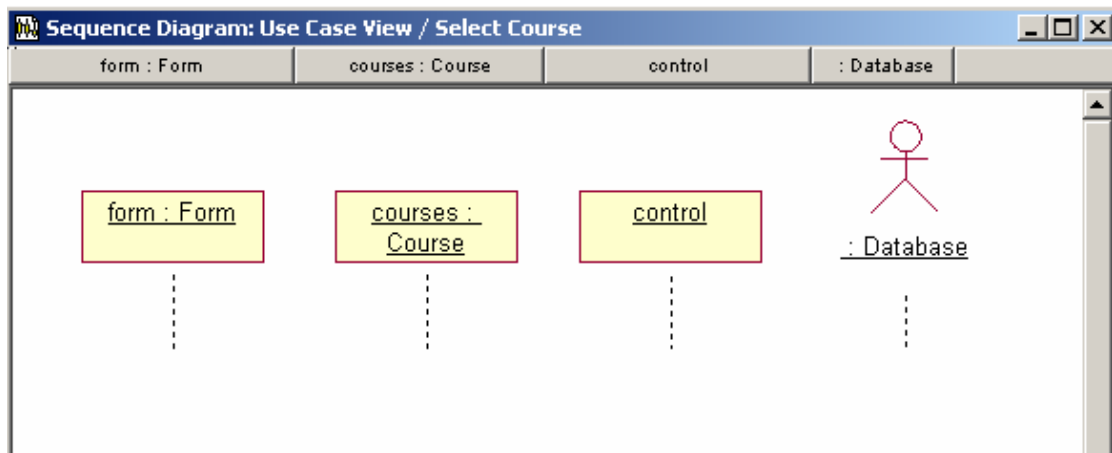


图 12 修改了的 courses 对象

(13)用同样的方法设置 control 对象的类为 ControlObject，并修改数据库角色的名称为

“db”（见图 13）。

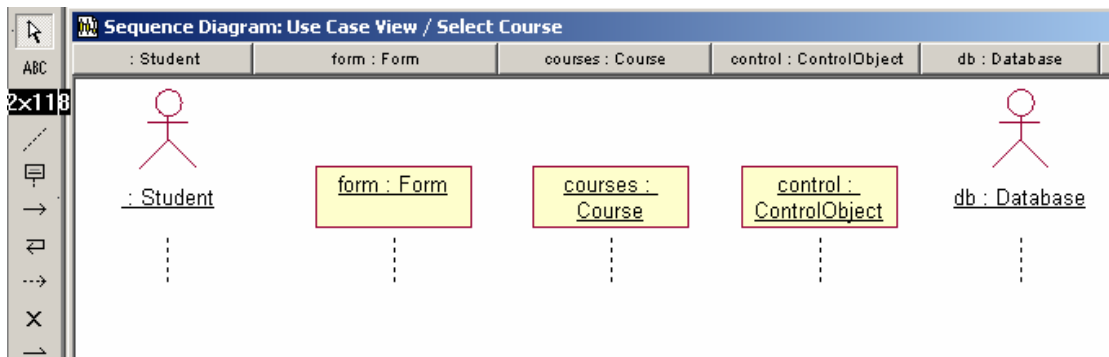
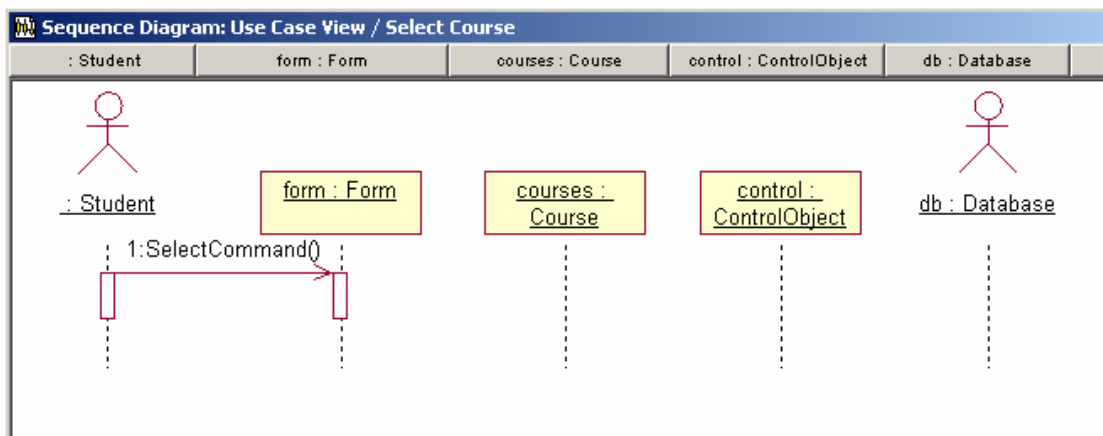
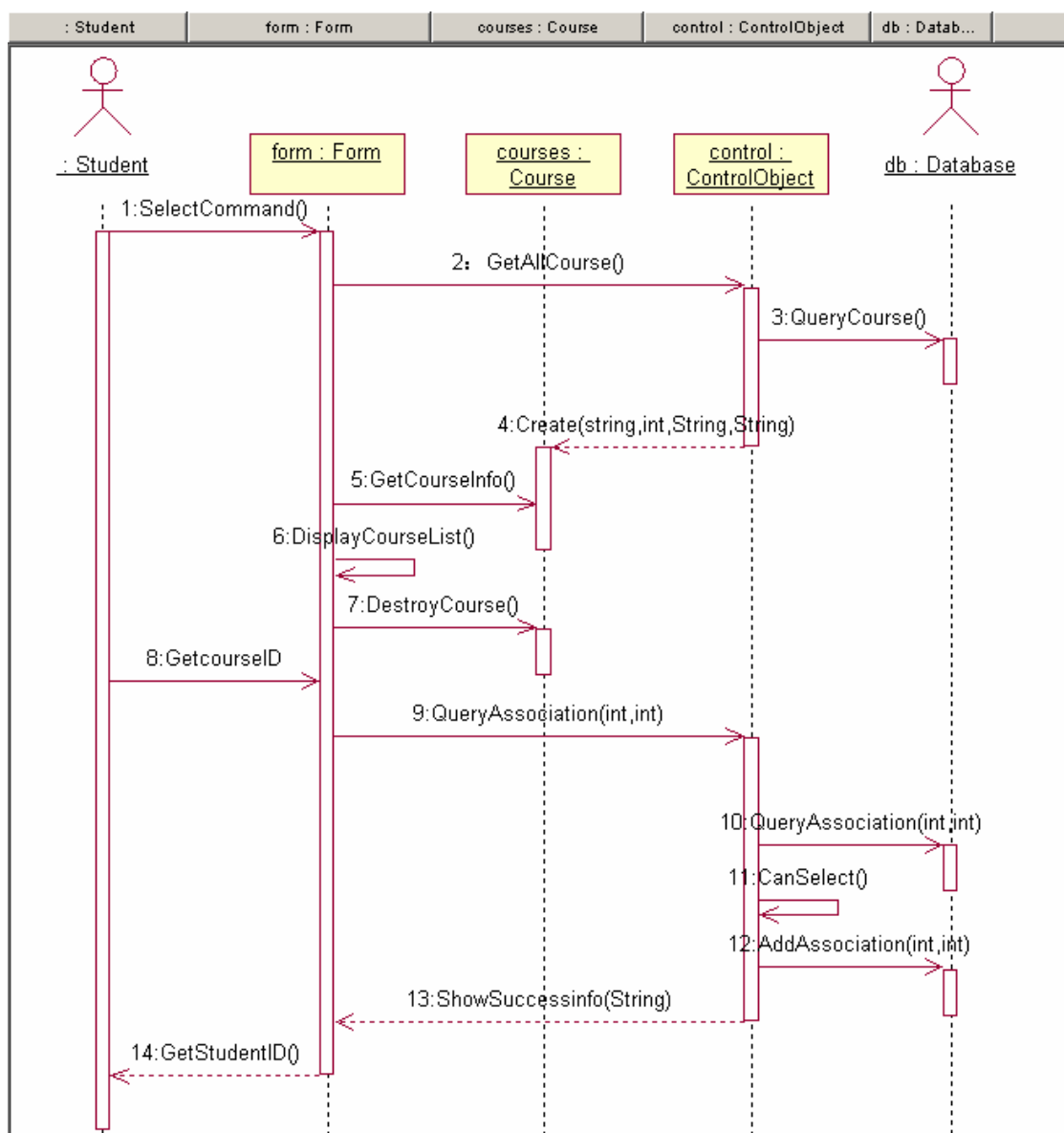


图 13

(14) 下面在序列图中增加消息。选择工具栏中的“Object Message”图标，在序列图中将光标从 stu 指向 Form，释放鼠标右键，则 stu 和 Form 之间添加了一个消息，标有序号 1，在其后添加消息名称 SelectCommand() (见图 14)。



(15) 重复以上过程，完成整个序列图(见图 15)。





这里顺便介绍一下从序列图中删除对象的方法。用鼠标在序列图窗口选择一个对象，在“Edit”菜单中选择“Delete from Model”，则所选择的对象被删除。

4 Add Course 序列图的创建

先介绍 Add Course 序列图。

Add Course 用例的事件流如下：

- (1) 管理员选择进入管理界面，用例开始。
- (2) 系统提示输入管理员密码。
- (3) 管理员输入密码。
- (4) 系统验证密码。
- A1: 密码错误
- (5) 进入管理界面，系统显示目前所建立的全部课程信息。
- (6) 管理员选择添加课程。
- (7) 系统提示输入新课程信息
- (8) 管理员输入信息。
- (9) 系统验证是否和已有课程冲突。

A2: 有冲突

- (10) 系统添加新课程，提示课程添加成功。
- (11) 系统重新进入管理主界面，显示所有课程。
- (12) 用例结束。

首先，查找 Add Course 用例的对象。从事件流中发现涉及以下对象：

- (1) 界面。
- (2) 课程。
- (3) 对于业务层的操作，也应该有对象进行处理。
- (4) 事件流中设计的角色有：管理员，数据库。

然后，分析对象、角色之间交互的消息。本用例主要有以下交互：

- (1) 管理员进入管理界面，选择添加课程功能。
- (2) 界面提示用户输入课程信息。
- (3) 界面对象创建一个课程对象。
- (4) 通过控制对象来对课程信息进行合法性检查。



- (5) 控制对象向课程对象返回结果。
- (6) 控制对象向数据库查询课程相关信息。
- (7) 控制对象对查询结果进行判断。
- (8) 控制对象向数据库中插入数据。
- (9) 在界面上显示结果。
- (10) 控制对象撤销建立的课程对象。

根据以上分析，整理的 Add Course 序列图如图 16 所示。

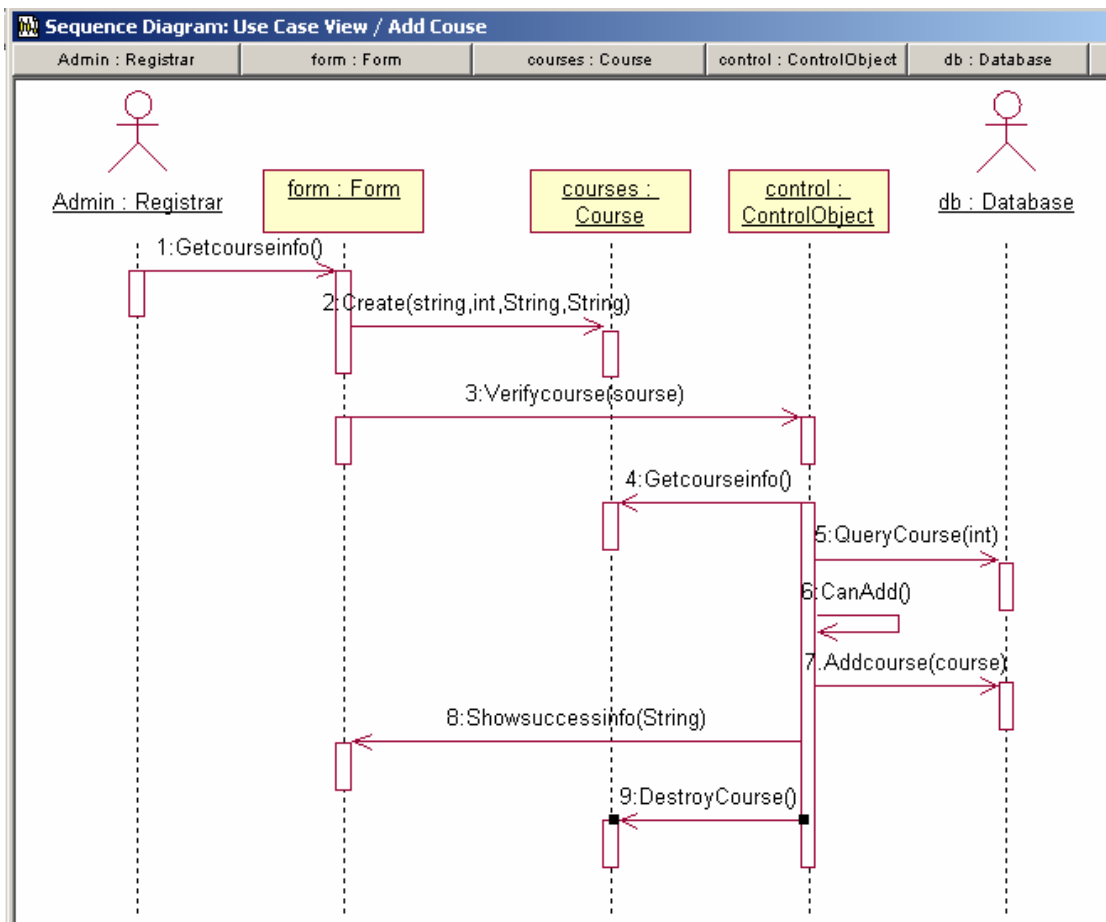




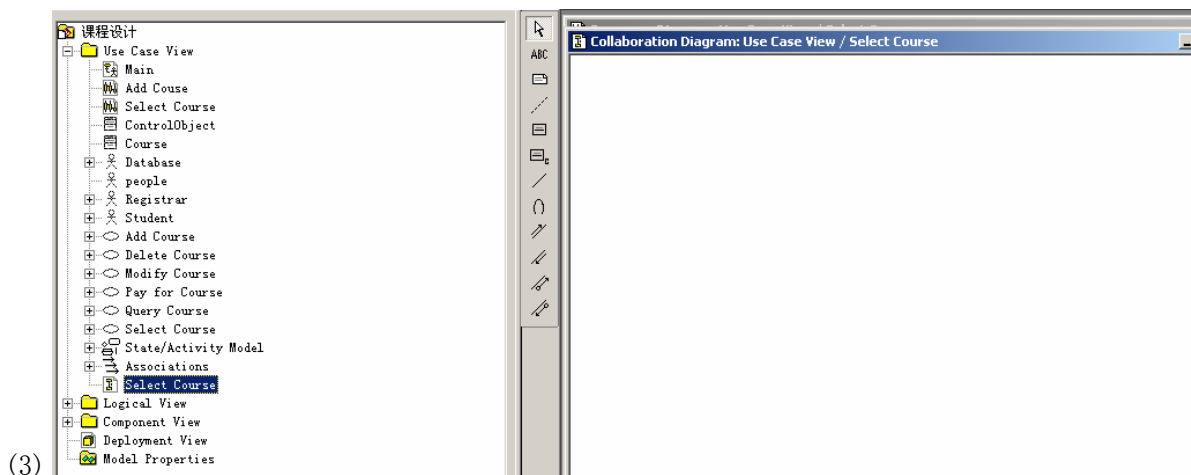
图 16 Add Course 序列图

5. Select Course 协作图的创建

下面介绍 Select Coursef 协作图的创建，具体步骤如下：

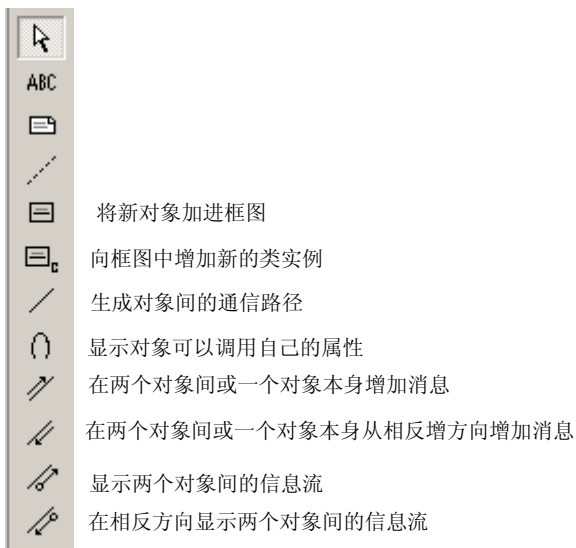
- (1) 在浏览器的“Use Case View”中的用例 Select Course 上按下鼠标右键，在弹出的菜单中选择“New”，然后再在弹出的菜单中选择“Collaboration Diagram”，创建一个新的协作图，将协作图的名字改成 Select Course(见图 17)。

(2)

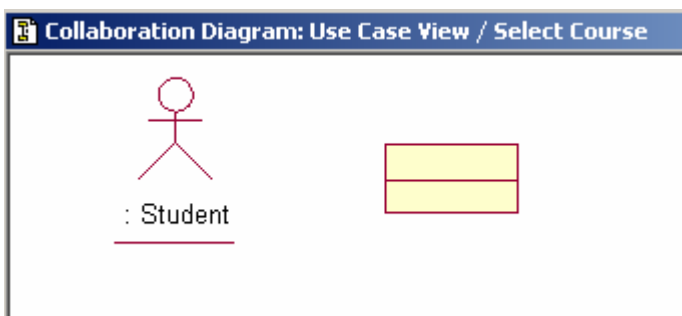


(3)

- (2) 选择协作图 Select Course，协作图窗口工具栏如图 18 所示。



- (4) Select Course 协作图涉及以下对象：学生、界面、控制对象、数据库对象、课程对象。下面操作在图中添加对象：在 Use Case View 中选择 Student 角色，将其拖动到协作图窗口；再选择工具栏图标 “Object”，如图 19 所示，在协作图窗口中按下鼠标左键，添加一个对象



- (5)
- (4) 选择对象并按下鼠标右键，在弹出的菜单中选择 “Open Specification”，然后在弹出的对话框中设置对象的属性，如图 20 所示。

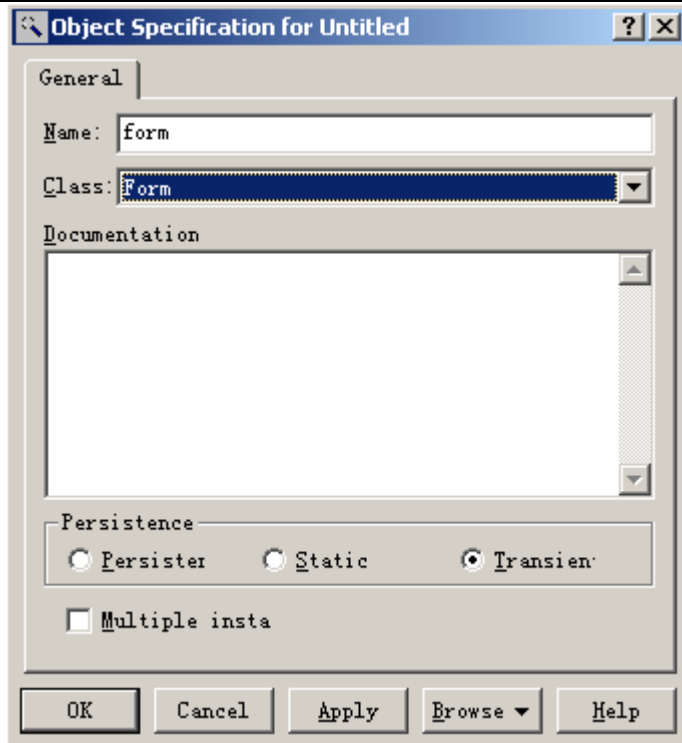


图 20 设置对象的属性

(5) 采用同样的方法添加对象 control、courses 和 db，结果如图 21 所示。

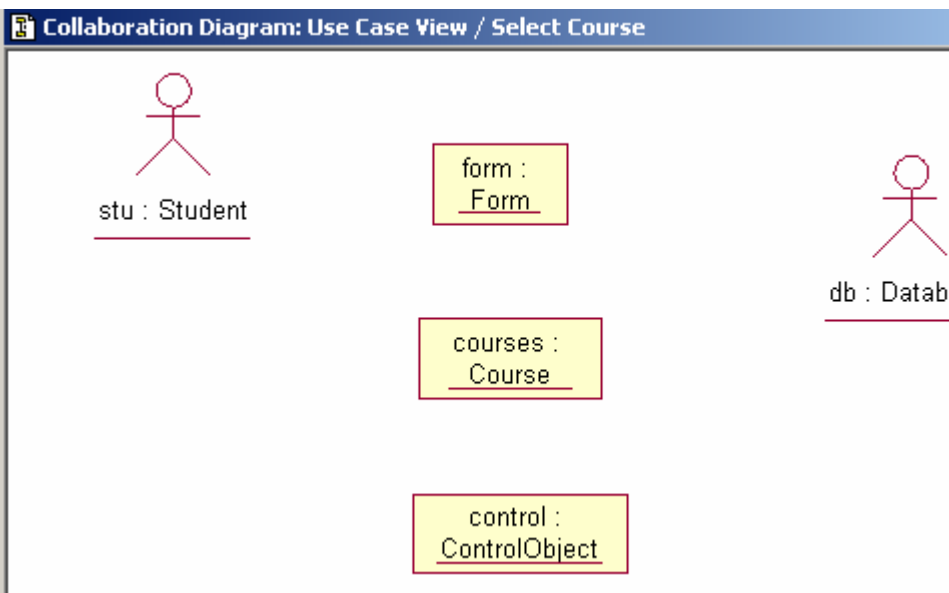


图 21 添加其他对象

(6) 对象之间的交互如下：

- 学生通过界面发送选课命令。
- 界面向控制对象请求课程信息。
- 控制对象向数据库发送查询数据消息。
- 控制对象暂存数据库的查询结果。
- 界面对象从控制对象中取得所有的课程信息。
- 在界面上显示所有的课程信息。
- 界面对象发送命令要求控制对象删除课程信息。
- 学生选择课程。
- 界面对象要求学生输入学号。
- 界面对象向控制对象发送信息，查询该生是否可以选定的课程。
- 控制对象从数据库中查询关联信息。
- 控制对象判断是否可以选课。
- 如果可以选课，则向数据库中添加关联信息。

- 向界面对象返回信息。

采用以下方法在对象之间添加消息：

选择“Object Link”工具栏图标，将光标移到协作图窗口，由 stu 指向 form，建立 stu 到 form 的连接，结果如图 22 所示。

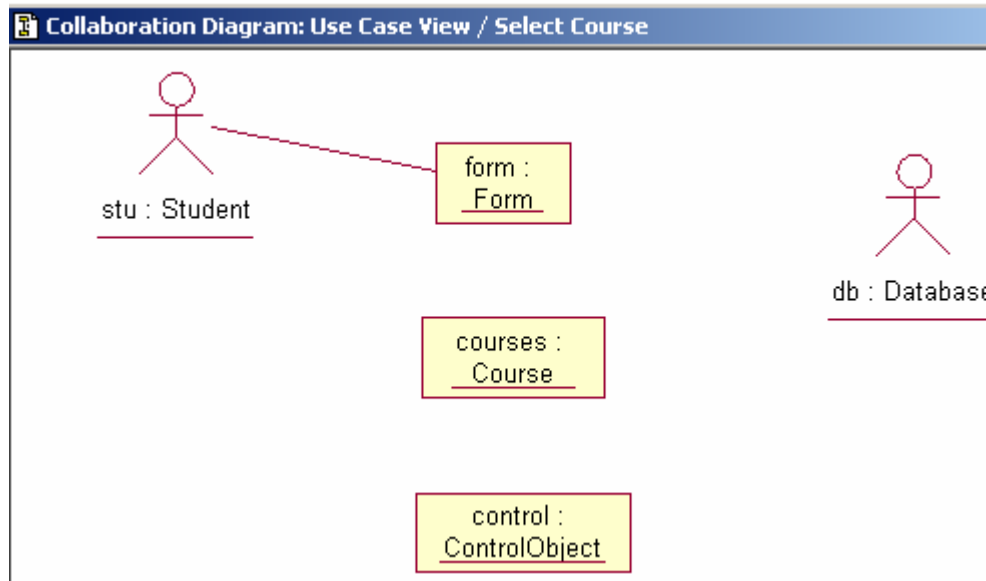


图 22 建立连接

(7) 选择工具栏中的“Link Message”图标，点击刚才添加的连接，则添加一条消息，如图 23 所示。

Collaboration Diagram: Use Case View / Select Course

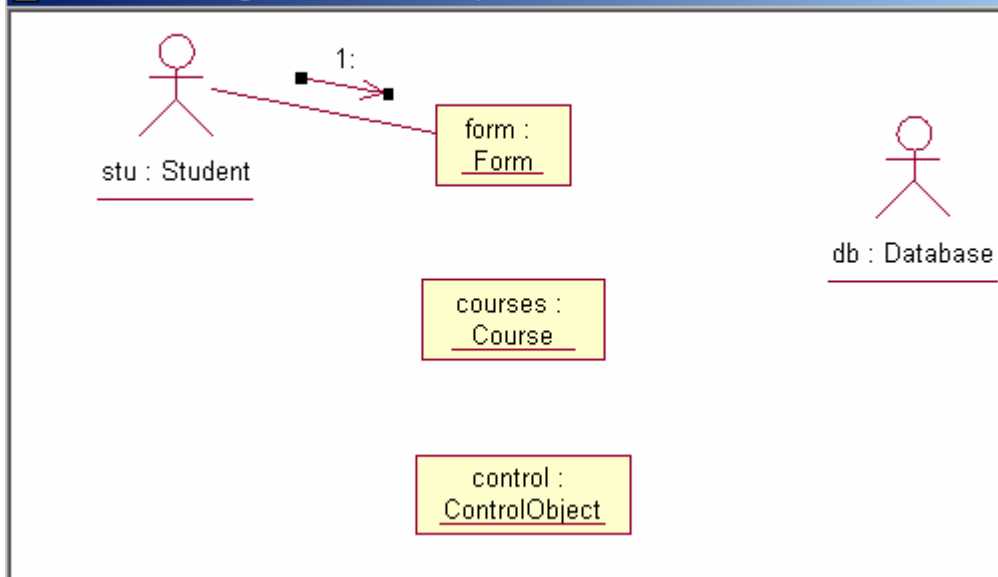
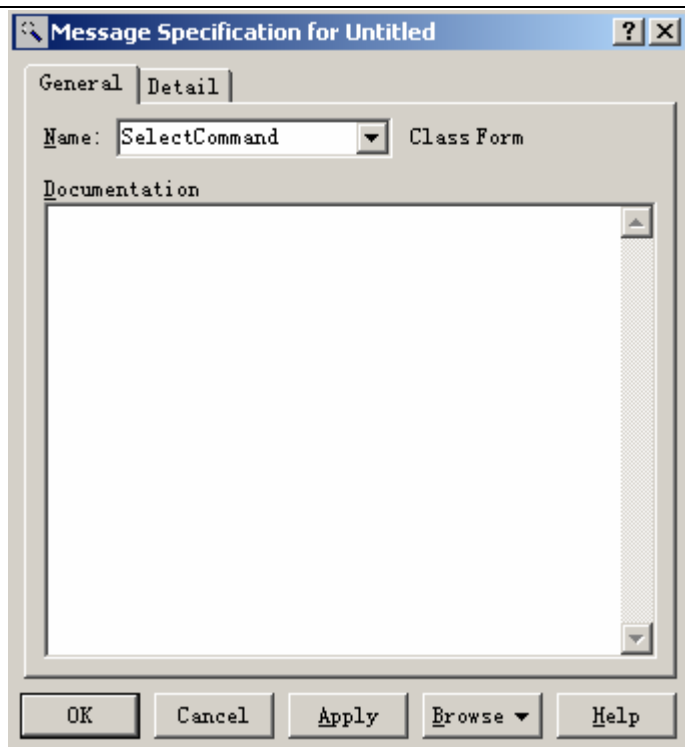


图 23 添加消息

(8) 从步骤(6)中得知学生发送到界面的第一条消息是“学生通过界面发送选课命令”，记作 SelectCommand，下面将其设置到消息上去。在协作图窗口上选择 1:，点击鼠标右键，在弹出的菜单中选择“Open Specification”，弹出图 24 所示的对话框，在对话框中输入消息的名字。按下“OK”按钮，得到图 25 所示的形式。



Collaboration Diagram: Use Case View / Select Course

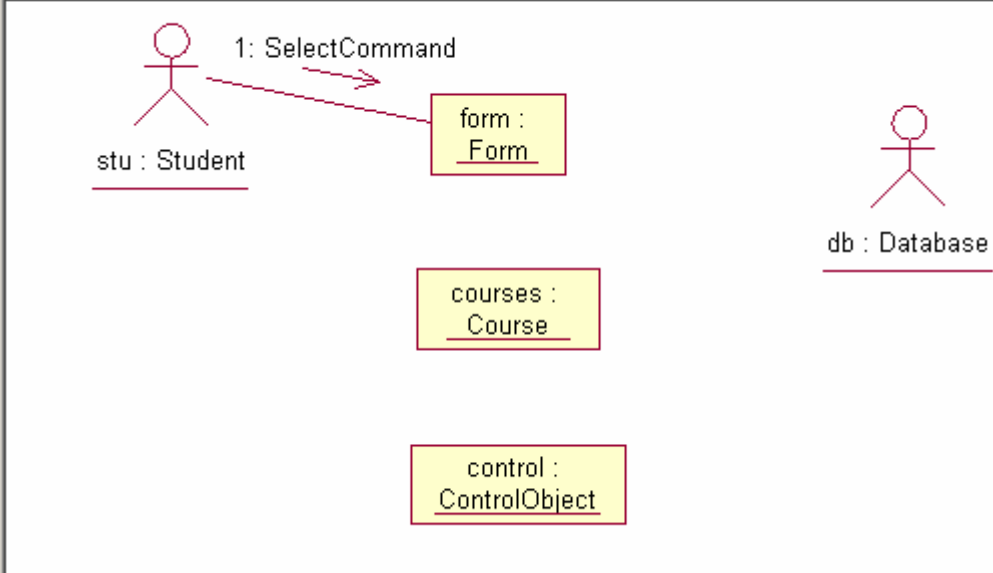
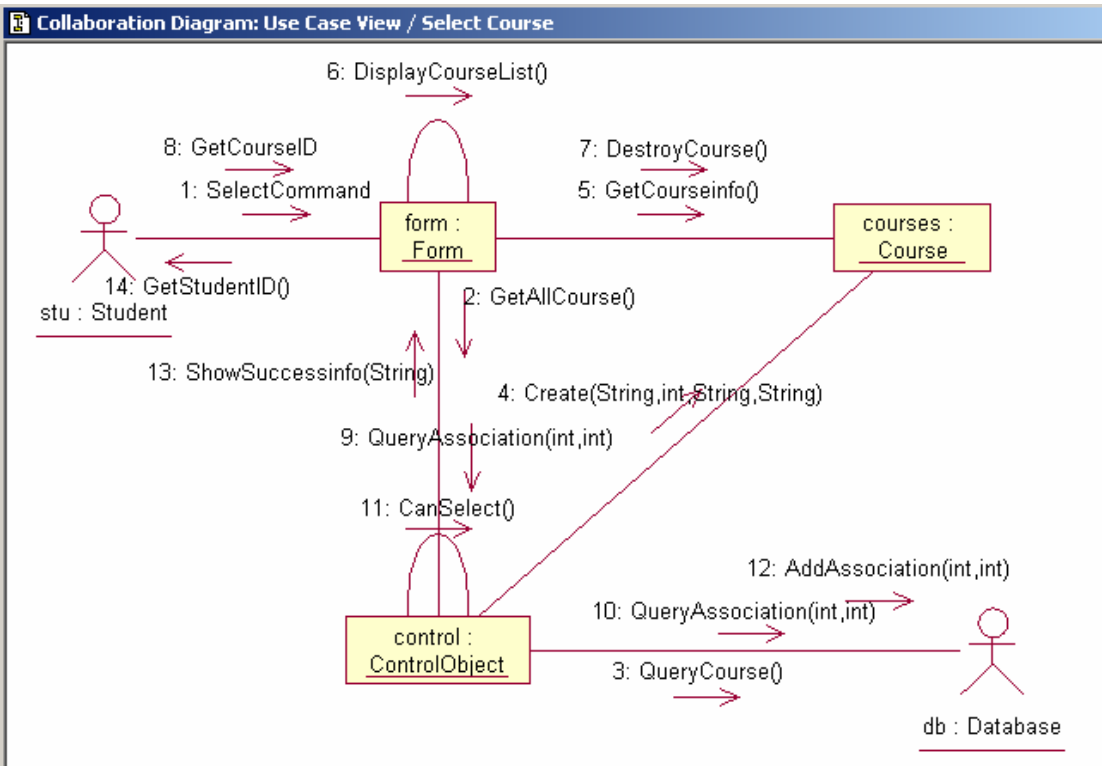


图 25 设置了属性的消息

(9) 采用以上方法添加对象、连接和消息，设置消息的属性，最终得到如图 26 所示的协作图。



如图 26 Select Course 协作图

6. 小结

本实验做了序列图和协作图，它们描述了用例的初始实现。在实际设计时，只要做成其中的一种图，然后选择图，按下“F5”键，即可生成另一种图。

在 UML 中，用例的实现用交互图来指定和说明。交互图通过显示对象之间的关系和对象之间处理的消息来建模系统的动态特性。有两种交互图，它们是序列图和协作图。本实验介绍了序列图和协作图的基本概念、组成以及在 Rose 中的实现。



选课系统的类图设计和实现

1 类的创建

本实验先分析选课系统中的类，然后作出它们的类图。

1.1 选课系统中的类

分析实验六中的所有序列图，除了角色之外，有以下名词：

- (1) 课程。
- (2) 界面。
- (3) 控制对象。

这 3 个名词可以抽象出 3 个类：课程类、界面类和控制对象类。

首先看课程类(Course)。课程类应该有如下属性：

- (1) 课程名称(name)
- (2) 开课教室(classroom)
- (3) 课程号(courseID)
- (4) 授课教师(teacher)
- (5) 选课的学生(students)
- (6) 开课起始时间(firstlessondate)
- (7) 允许选课的学生人数(maxstudents)
- (8) 设置课程名称(setname)
- (9) 设置开课教室(setclassroom)
- (10) 设置课程号(setcourseID)
- (11) 设置授课教师信息(setteacher)
- (12) 设置开课起始时间(setfirstlessondate)
- (13) 设置允许选课的学生人数(setmaxstudents)
- (14) 查询课程名称(getname)
- (15) 查询开课教室(getclassroom)
- (16) 查询课程号(getcourseID)
- (17) 查询授课教师信息(getteacher)
- (18) 查询开课起始时间(getfirstlessondate)



(19) 查询允许选课的学生人数(getmaxstudents)

关于界面类和控制类，因为涉及到后续的内容，所以本实验暂时不考虑它们。

1.2 类图的创建

在下面介绍在 Rose 中创建 Course 类的过程。

(1)选择浏览器中的“Logical View”，按 F 鼠标右键，在弹出的菜单中选择“New”，再在下一级菜单中选择“Class”，创建一个新的类，然后将这个类的名字改成“Course”，结果如图 1 所示。

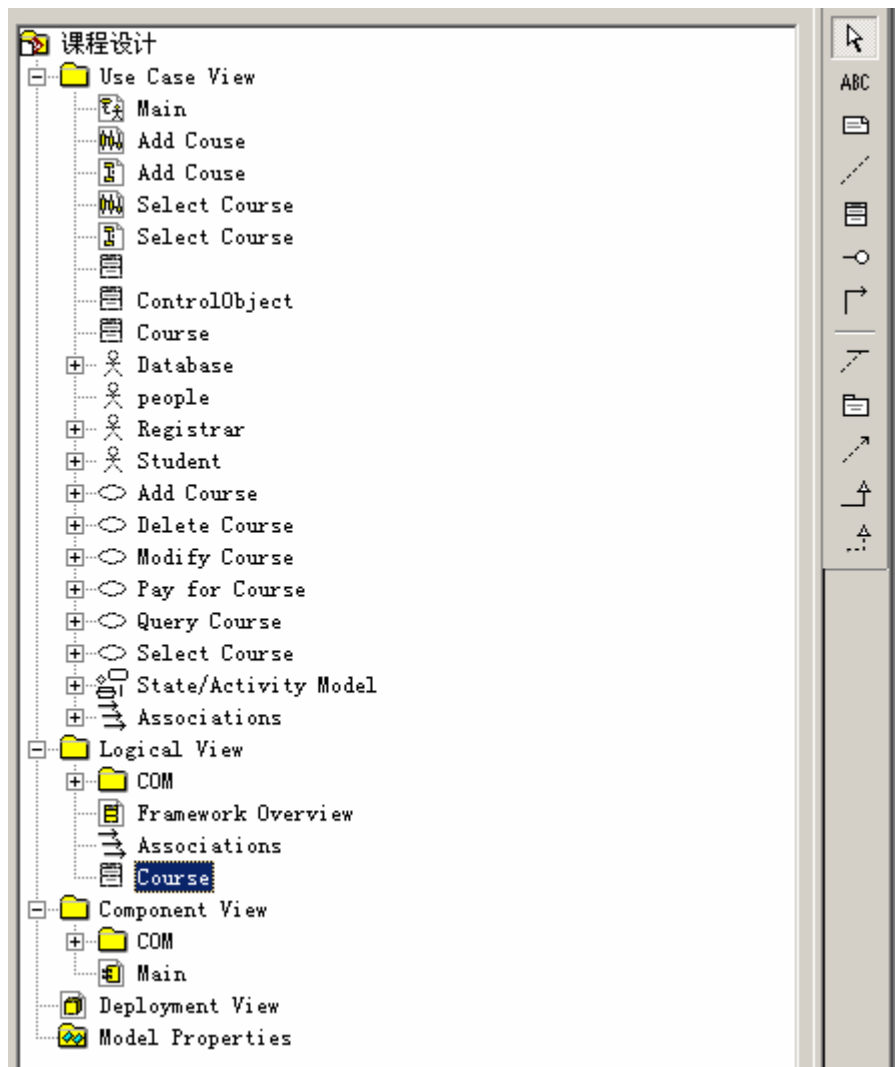
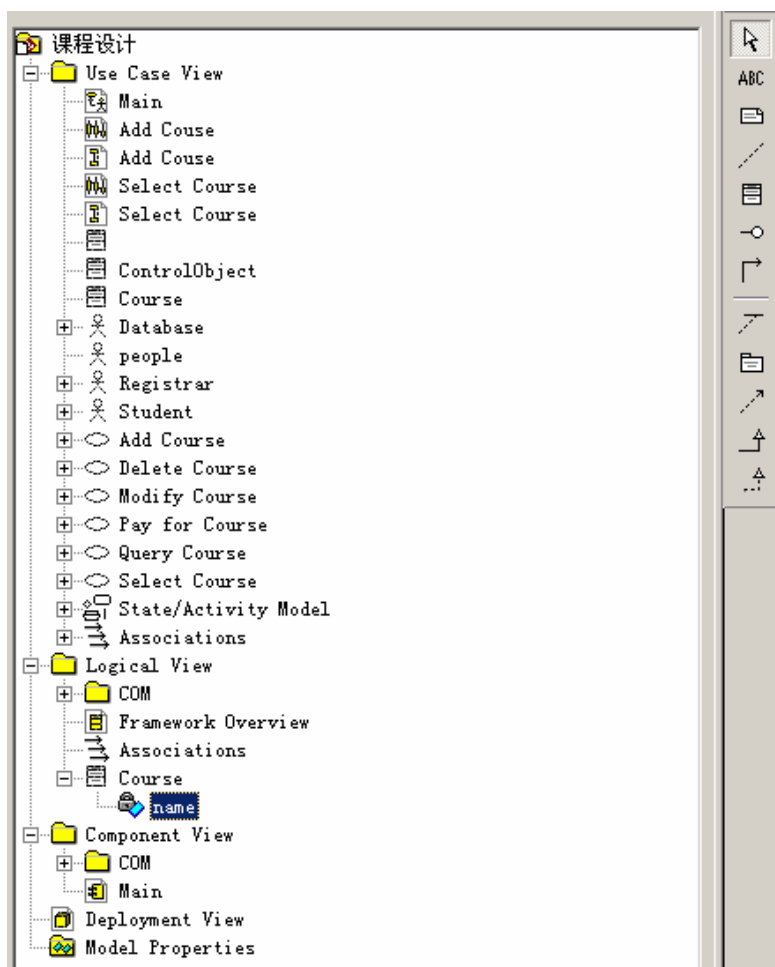
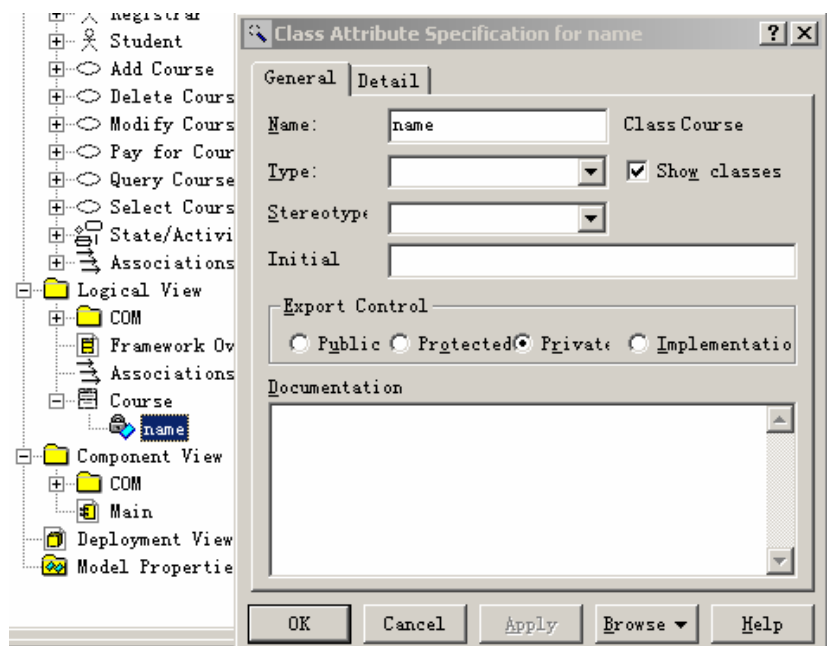


图1 创建 Course 类的示意图

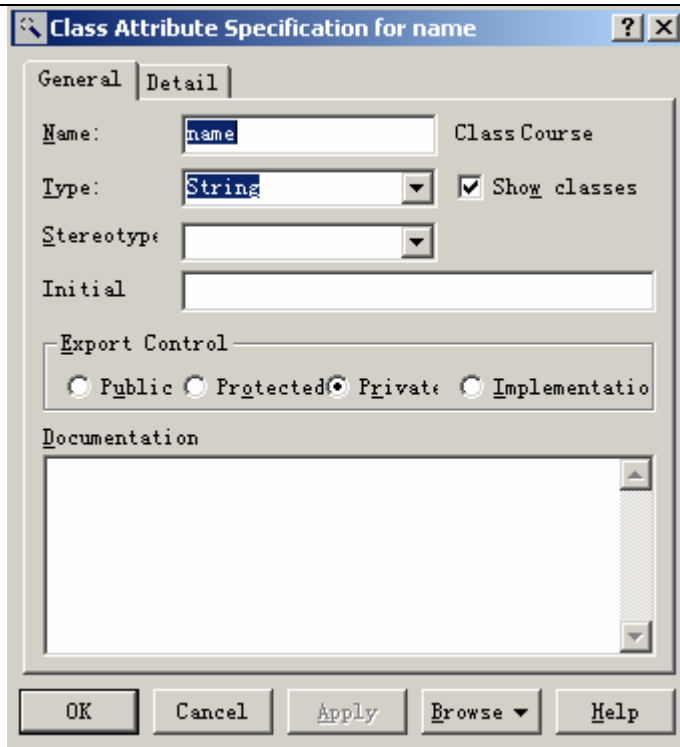
(2) 添加属性。在浏览器中选择类“course”，按下鼠标右键，在弹出的菜单中选择“New”，再选择“Attribute”，则一个新的属性被添加，将属性名字改成“Name”，如果如图2所示。



(3)选择 name, 按下鼠标右键, 选择“open Specification”, 则弹出如图 3 所示的“Class Attribution Specification”对话框。在这个对话框中有两个选项卡, 一个用来设置属性的固有特性, 比如类型 (Type)、版型 (Stereotype)、初始值 (Initial)、存取控制 (Export Control) 等, 一个用来进一步指定属性是静态 (Static) 的还是继承 (Derived) 的等。



(4)图 4 所示为设置属性的类型 (Type) 示意图。同样可以设置初始值 (Initial) 和存取控制 (Export Control) 等。



(5)图 5 给出进一步设置属性的示意图。图中将“Containment”设置为“By Value”属性的“Containment”特征表示属性如何存放在类中。By value 表示属性放在类中，By reference 表示属性放在类外，类指向这个属性。Unspecified 表示还没有指定控制类型，应在生成代码之前指定 By value 或 By reference。

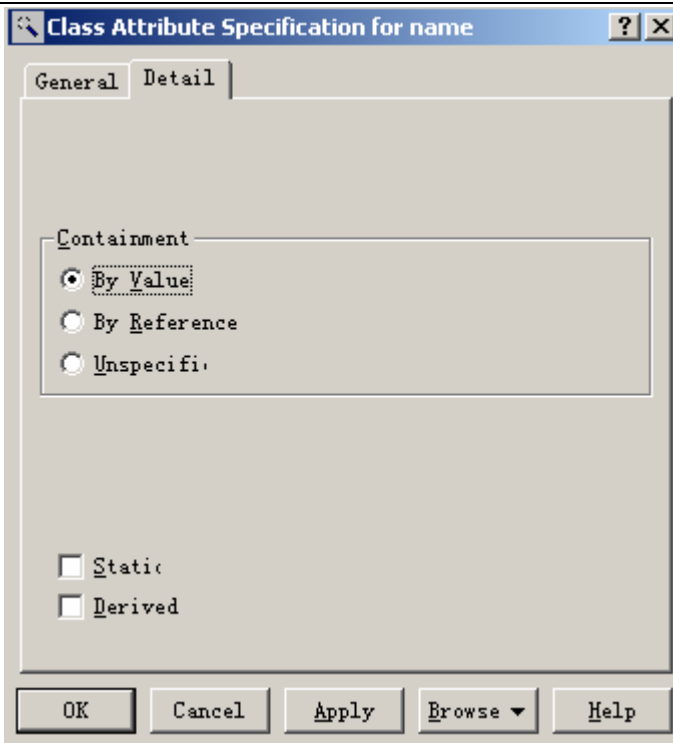


图 5

(6) 如果要删除属性，右键选择属性，在弹出的菜单中选择“Delete”即可。

(7) 添加操作。在浏览器中选择类“Course”，按下鼠标右键，在弹出的菜单中选择“New”，再在 F 一级菜单中选择“Operation”，则添加一个新的操作。图 6 是将添加操作的名字改为“GetName”的示意图。

(8) 选择 GetName，按下鼠标右键，选择“Open Specification”，在弹出的对话框中可以设置操作的固有特性。如图 7 所示。

(9) 在 Specification 对话框中，操作或者属性都有存取控制的选项，操作的存取控制的选项默认值是公有的，属性的存取控制的选项默认值是私有的。不同的存取控制采用不同的标记表示(如图 8 所示)。本图只是说明不同的控制标记，后面的图将按照实际要求设计。



(10) 重复以上步骤，完成 Course 类的设计。图 9 是设计结果图

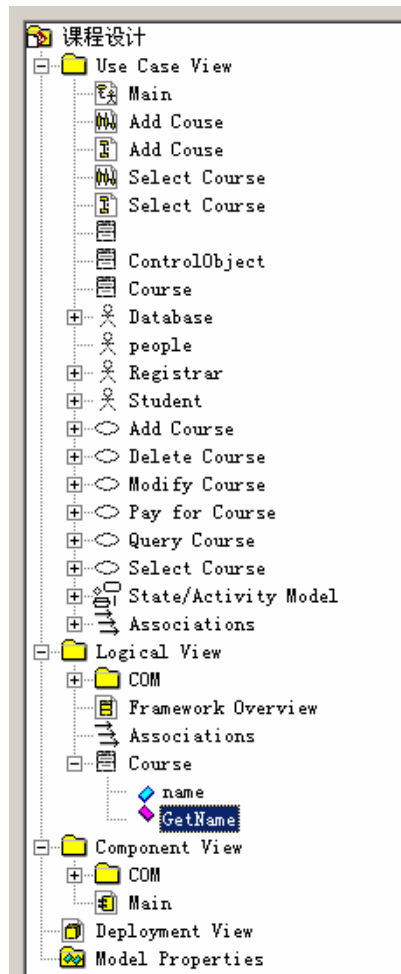


图 6 添加操作示意图

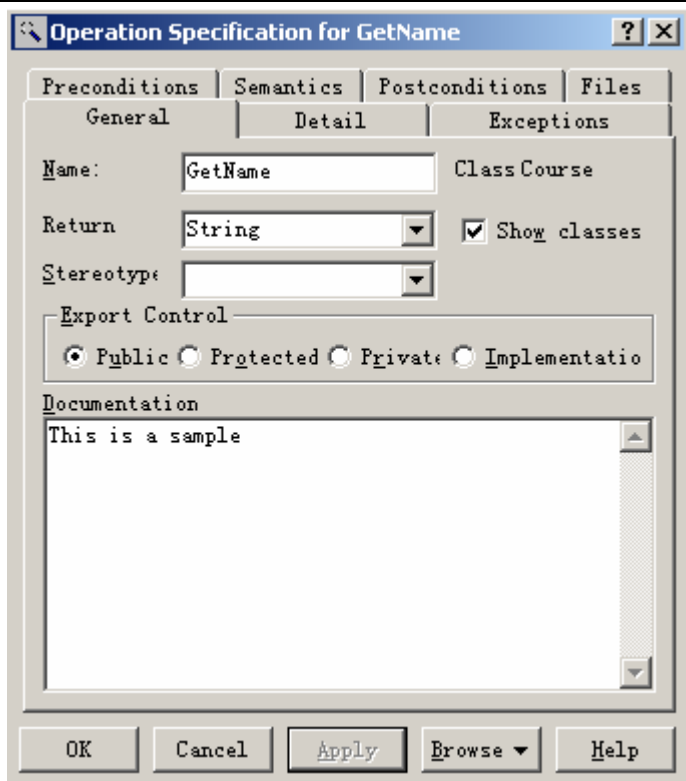


图 7 设置操作的特性

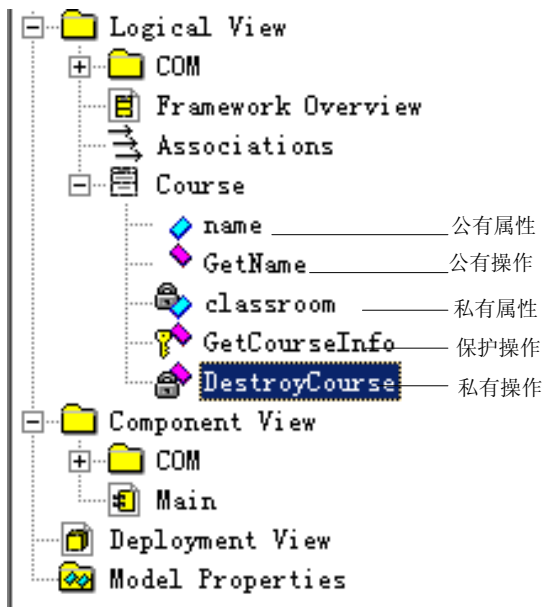


图 8 属性和操作的不同的存取控制

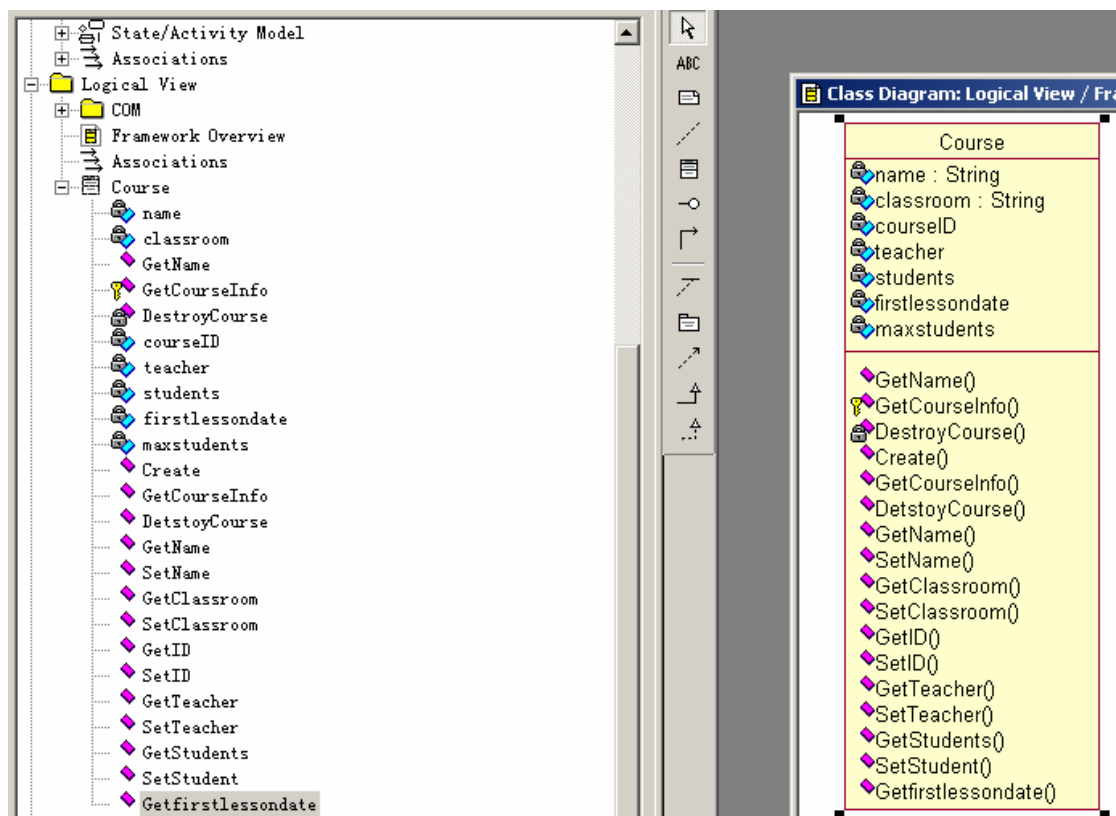


图9 完整的 Course 类

1.3 小结

类能实现对象的封装，是面向对象的重要特征。通过第一部分的学习，应该能够了解类、对象、属性、操作的概念，可以从一类单一的对象中抽象出类的定义，并依据实际要求，设计类的属性和操作。

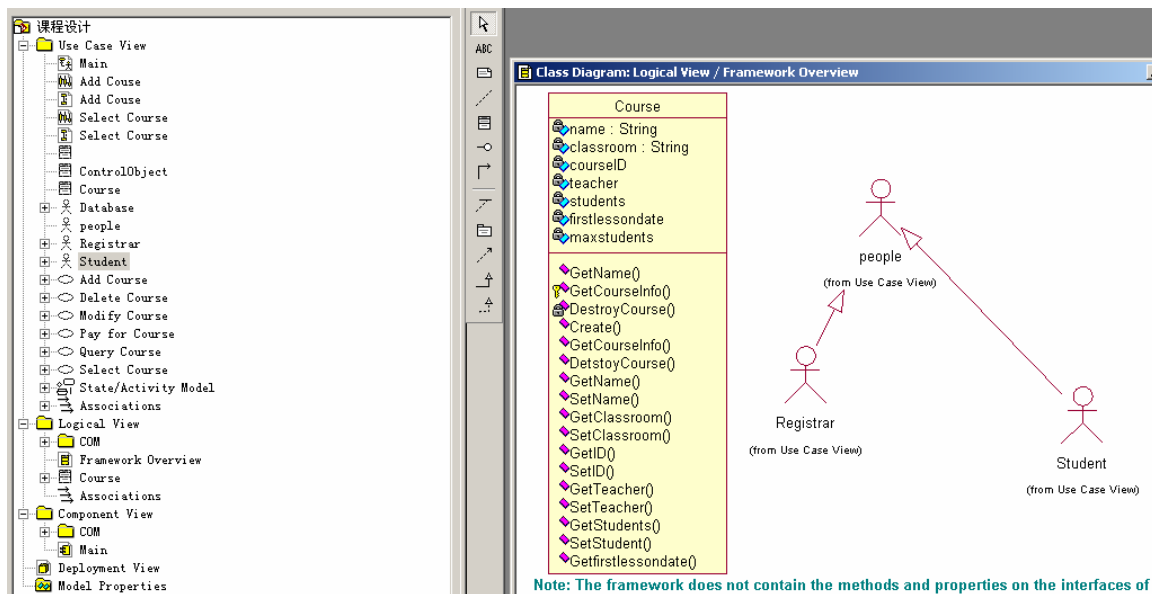
2 类的继承

在选课系统中，涉及到的用户包括管理员 (Registrar) 和学生 (Student)，其主要特征相似，所以可以建立统一基类 People，而 Registrar 和 Student 则由 People 派生。

2.1 在角色或类之间添加继承关系很简单，下面给出实现过程。



- (1) 双击浏览器中 Logical View: 的 Main 图标，弹出类图窗口，从 Use Case View 下将 People、Registrar 和 Student 拖到窗口中。图 10



(2) 选择类图工具栏的“Generalization”图标，在类图窗中，按下鼠标左键，将光标从角色 Student 移动到 People，如图 10 所示，在二者之间出现一个表示继承关系的箭头标志。同理添加 Registrar 与 People 之间的继承关系。(若用例图中已画则会自动生成)

(3) 如图 11 所示，给 People 增加一私有属性 name，这时 Registrar 和 Student 都没有设置属性。在类图窗口中双击 Student 的图标，弹出“Class Specification”对话框，点击 Attributes 选项卡，注意这时 Show inherit 被选择，而 Attributes 列表中没有任何属性。从图中可以看到，父类中的私有属性没有被继承。

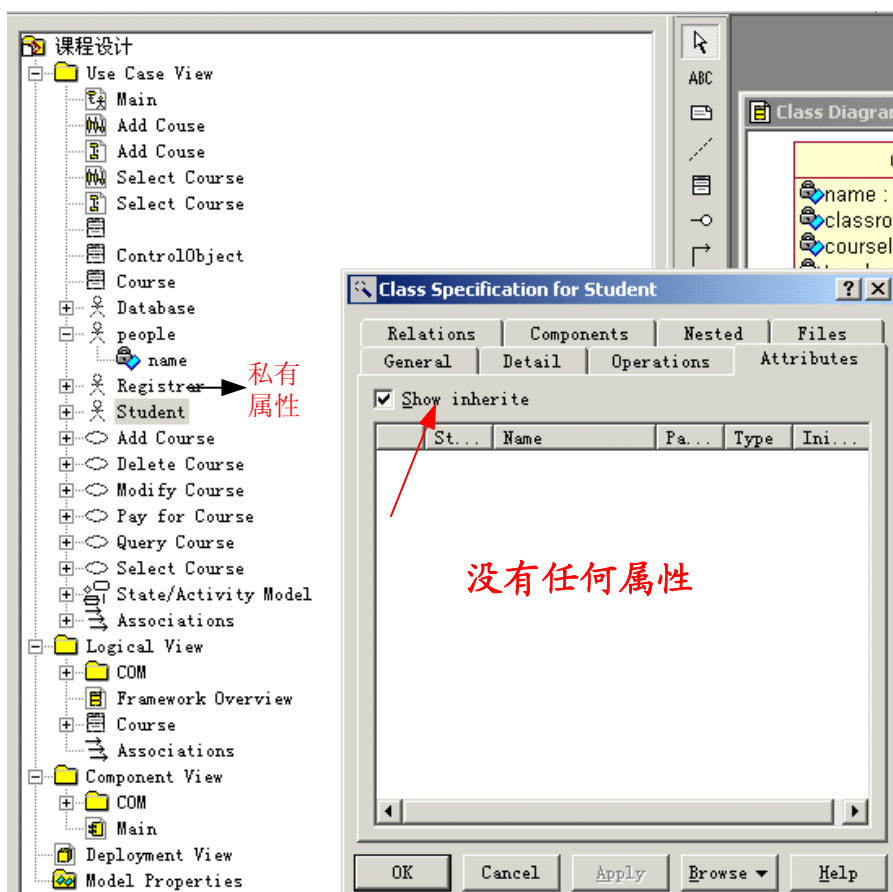


图 11

- (1) 将 People 的 name 属性设置成保护属性，操作同(3)，可以看到 Show inherit 被选择时，Attributes 列表中有一个属性 name（见图 12），父类中的保护属性被子类继承。

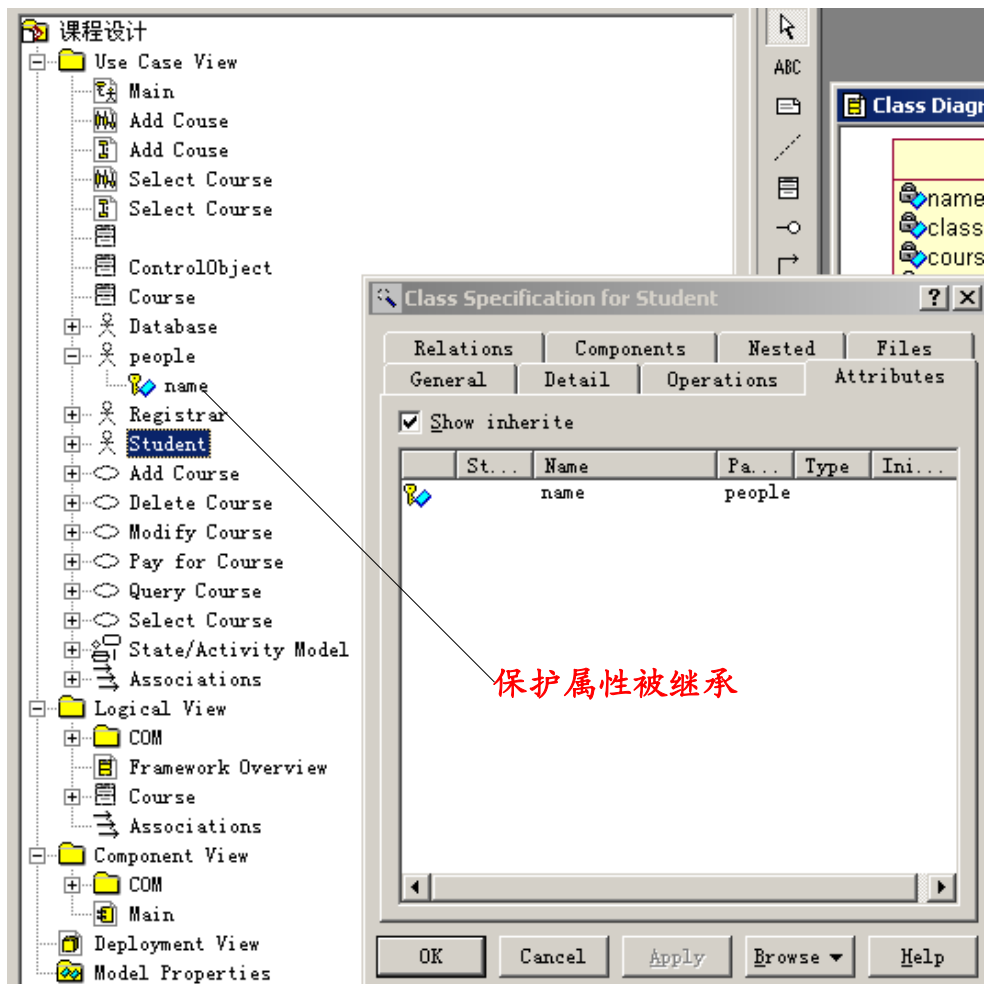


图 12 保护属性的被继承

(5) 将 People 的 name 属性设置成公有属性，操作同(3)，可以看到 Show inherit Attributes 列表中有一个属性 name(见图 13)，父类中的公有属性被子类继承。

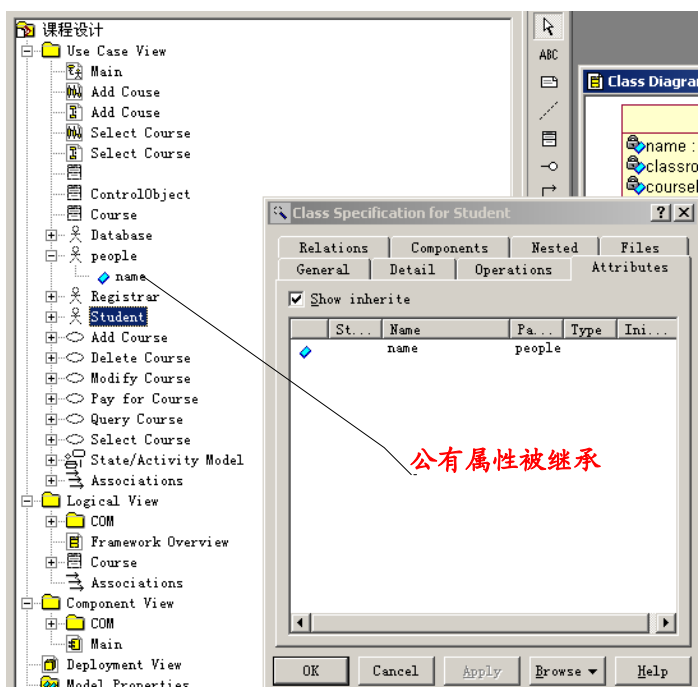


图 13 公有属性继承性示意图

2. 2 小结

继承是面向对象的重要特征之一，通过继承，可以创建事务的层次分类。可以首先定义

一个具有广泛意义的类，然后从它进行派生，添加一些具有特定特征的类。

当一个类是另外一个类的特殊情形时，可以考虑引入继承关系。

父类的属性和操作，只有当存取控制是公有和保护型时，才能被继承。私有属性和操作不能被子类继承。

3. 类的关联

可以使用下列的指导方针列出暂时性的关联关系：

- (1) 动词与介词词组。
- (2) 两个或两个以上的类有相依赖的关系，就可能有关联。



(3) 隐含的关系。

(4) 一般的常识。

然后依据以下的规则去除假性的关联：

(1) 如果有一类别被去除，则与之相关的关联也应去除。

(2) 现在不考虑与外界领域的关联或是实现时才会产生的关联。

(3) 关联是说明结构的属性而不是在说明事件。

3.1 选课系统的关联关系分析

在选课系统中设计了以下类：

1) 界面 FormObject。

2) 控制对象 ControlObject。

3) 课程 Course。

设计了以下角色：

1) 学生 Student。

2) 管理员 Registrar。

3) 学生和管理员的父类 People。

4) 数据库 Database。

这些类和角色之间的关系如下：

1) 角色 Student 和 Registrar 从 People 派生。

2) 学生、管理员在与系统交互时，都有一个界面与之对应。

3) 一个界面可能和课程相关(0-多门)。

4) 控制对象负责课程的处理，处理结果在界面上显示。

5) 控制对象完成对数据库的操作。

6) 界面请求控制对象的服务。

依据以上分析，画出图 8-6 所示的简略的类关系图。

进一步分析如下：

(1) 一个用户界面对象和 0 个或多个学生相关。

(2) 一个用户界面对象和 0 个或多个管理员相关。

(3) 一个用户界面对象和 0 个或多个课程信息相关。

(4) 一个控制对象和 0 个或多个课程信息相关。

(5) 一个用户界面对象和一个控制对象相关。



(6) 一个控制对象和一个用户界面对象相关。

更加详细的类图如图 14 所示，类的属性和操作请参见上面。

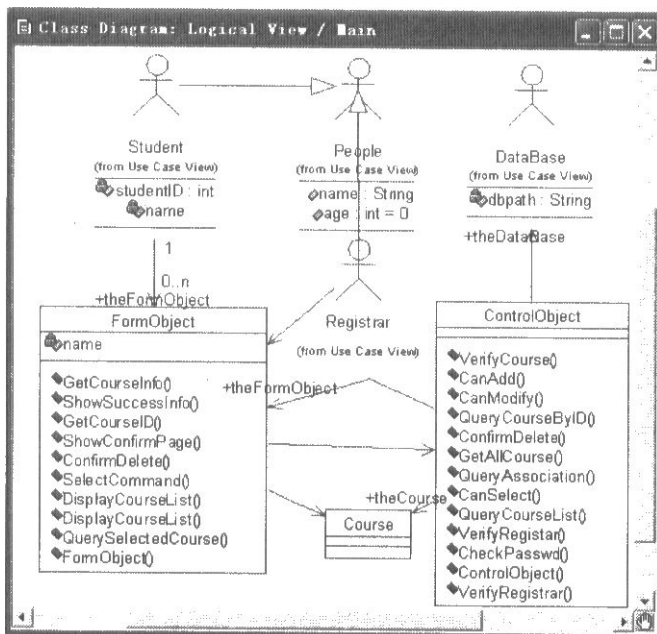


图 14 详细的类图

3.2 选课系统中的类、属性和操作

总结一下选课系统中定义的类、类的属性和操作。到目前为止，选课系统中自定义的类共有 7 个，如表 1 所示。

表 1 选课系统中定义的类

NO	类 名	说 明
1	People	Student 和 Registrar 的基类
2	Student	学生类
3	Registrar	管理员类
4	FormObject	界面类

* 为了使问题简化，这里将所有的界面都排看成是 FormObject 的对

象。在实际的系统中，要根据实际情况进行修改

5	Course	课程类
6	ControlObject	控制对象类
7	DataBase	数据库类

仔细分析系统的需求和前面的设计，可以得到每个类的属性和操作。下面对它们分别予以说明。

(1) 类 People 定义的属性和操作如图 15 所示，说明见表 2。

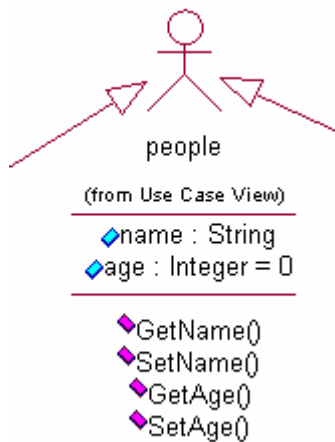


图 15

表 2

名 称	类 别 (型)	描 述
name	String	姓名
age	int, 初始值为 0	年龄
GetName ()		查询姓名
SetName ()		设置姓名
GetAge ()		查询年龄

SetAge()

设置年龄

(2) 类 Student 的属性和操作如图 16 所示，说明见表 3。

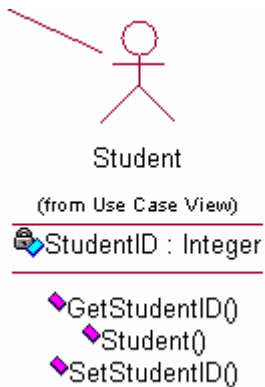


图 16

表 3 类 Student 的属性和操作

名 称	类 别（型）	描 述
StudentID	int	学号
Student()		初始化（构造函数）
GetStudentID()		查询对象学号
SetStudentID()		设置对象学号

因为 student 是从 People 继承的“Class Specification”对话框中，选择“Show inherit”，属性和操作显示如图 17 所示，从中可以看到它从 People 中继承的属性和操作。

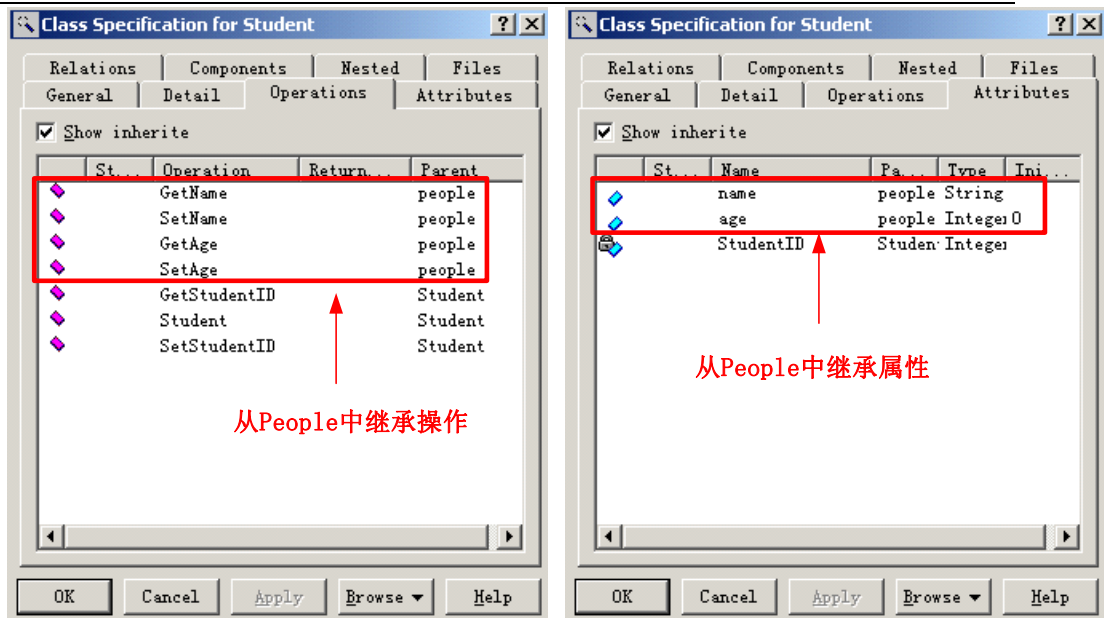


图 17

(3) 类 Registrar 的属性和操作如图 18 所示，说明见表 4。

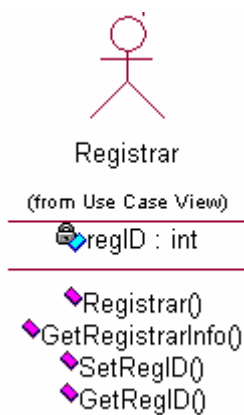


图 18

表 4 类 Registrar 的属性和操作

名 称	类 别 (型)	描 述
regID	int	管理员编号
Registrar		初始化 (构造函数)
GetRegistrarInfo()		查询管理员信息
SetRegID()		设置管理员信息
GetRegID()		查询管理员编号

因为 Registrar 是从 People 继承的“Class Specification”对话框中，选择“Show inherit”，属性和操作显示如图 19 所示，从中可以看到它从 People 中继承的属性和操作。

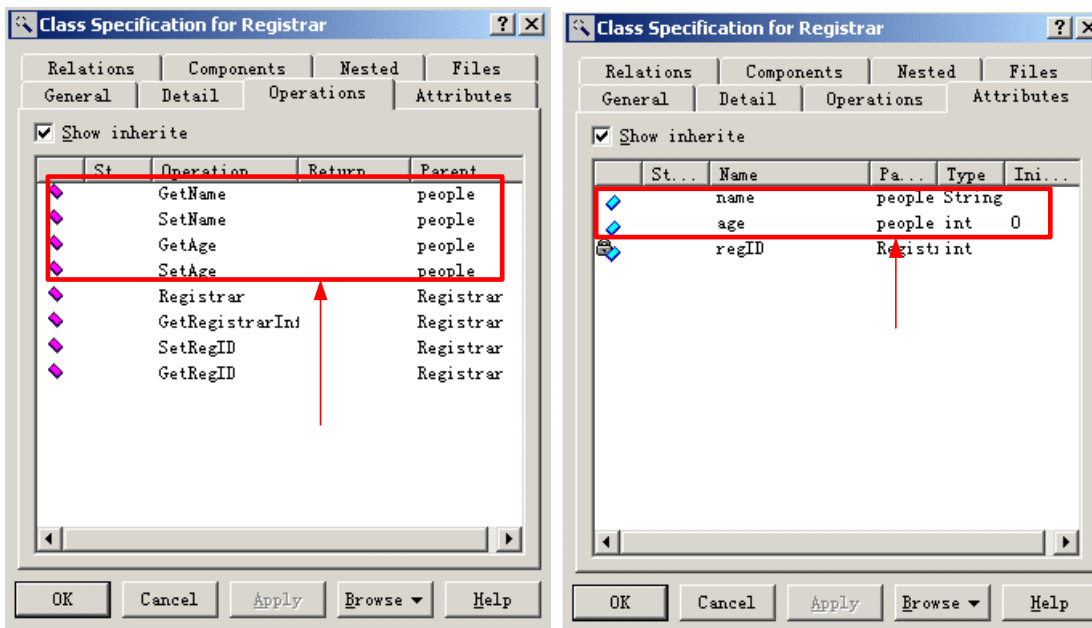


图 19 Registrar 所有属性和操作

(4)类 DataBase 的属性和操作如图 20 所示，说明见表 5。

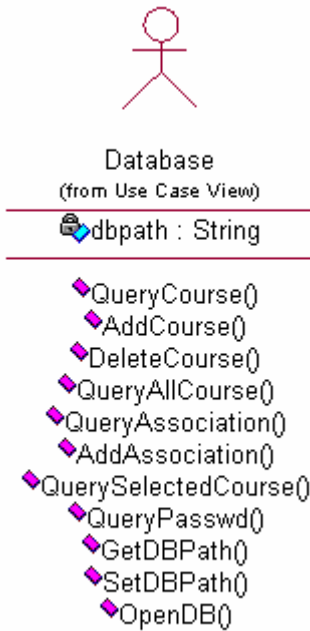


图 20

表 5 类 DataBase 的属性和操作

名 称	类 别（型）	描 述
dbDath	string	连接的数据库名称（位置）
QueryCourse()		查询课程
AddCourse()		添加课程
DeleteCourse()		删除课程
QueryAllCourse()		查询所有的课程
QueryAssociation()		查询关联
AddAssociation()		添加关联
QuerySelectedCourse()		查询选择的课程信息
QueryPasswd()		查询密码
GetDBPath()		取得数据库名称



SetDBPath()

设置数据库名称

OpenDB()

打开数据库

(2) 类 ControlObject 的属性和操作如图 21 所示，说明见表 6。

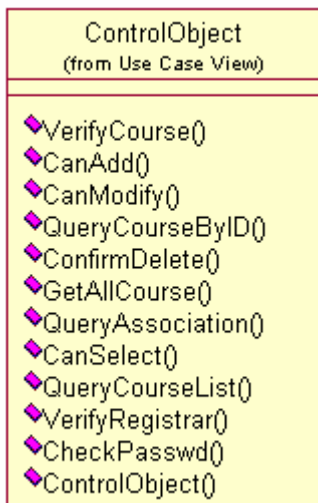


图 21

表 6 类 ControlObject 的属性和操作

名 称	类 别（型）	描 述
VerifyCourse()		验证课程的合法性
CanAdd()		判断能不能添加指定的课程
CanModify()		判断能不能修改指定的课程
QueryCourseByID()		通过 ID 查询课程
ConfirmDelete()		确认删除
GetAllCourse()		查询所有课程
QueryAssociation()		查询关联
CanSelect()		判断是不是能选择
QueryCourseList()		查询课程列表
VerifyRegistrar()		验证管理员的合法性



CheckPasswd()

检查密码

ControlObject()

构造函数

(3) 类的 Course 的属性和操作如图 22 所示，说明见表 7。

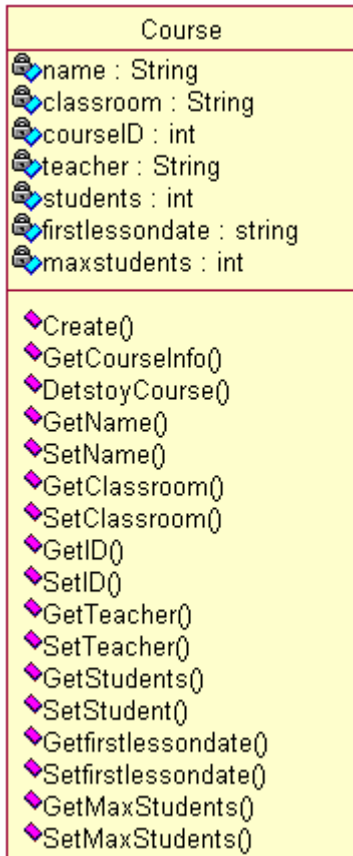


图 22

表 7 类 Course 的属性和操作

名 称	类 别(型)	描 述
name	string	课程名称
classroom	string	开课教室
courseID	int	课程编号



teacher	string	授课教师
students	int	选课学生
firstlessondate	string	开课起始时间
maxstudents	int	允许选课的最大学生数
Create()		构造函数
GetCourseInfo()		查询课程信息
DestroyCourse()		对象注销时的操作
GetName()		查询课程名称
SetName()		设置课程名称
GetClassroom()		查询教室
SetClassroom()		设置授课教室
GetID()		查询课程编号
SetID()		设置课程编号
GetTeacher()		查询信息
SetTeacher()		设置教师信息
GetStudents()		查询选课学生信息
SetStudents()		设置选课学生信息
Getfirstlessondate()		查询开课起始时间
Setfirstlessondate()		设置开课起始时间
Getmaxstudents()		查询选课的最大学生数
Setmaxstudents()		设置选课的最大学生数

(7)FormObject 的属性和操作如图 23 所示，说明见表 8—8。

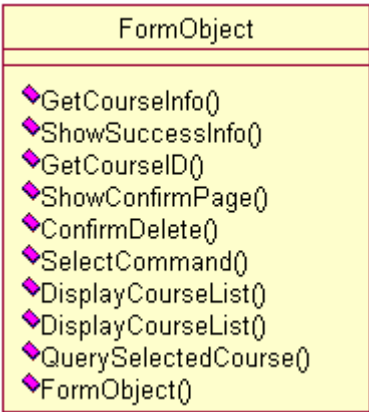


图 23

表 8 类 FormObject 的属性和操作

名 称	类 别(型)	描 述
GetCourseInfo()		查询课程信息
ShowSuccessInfo()		显示选课成功信息
GetCourseID()		查询课程编号
ShowCOnfirmPage()		显示确认页
COnfirmDelete()		确认删除
SelectCommand()		选择命令
DisplayCourseList()		显示课程列表
DisplayCourseList()		显示课程列表
QuerySelectedCourse()		查询选择的课程
FormObject()		构造函数

DisplayCourseList()是重载函数

3.3 关联关系的创建

前面介绍了类的创建，现在介绍类图的创建。操作步骤如下：

(1)在“课程设计”模型的 LogicalVieW 中，选择“Main”类图，将上节中定义的类型或角色从浏览器中拖至“Main”类图窗口，详情见图 24。

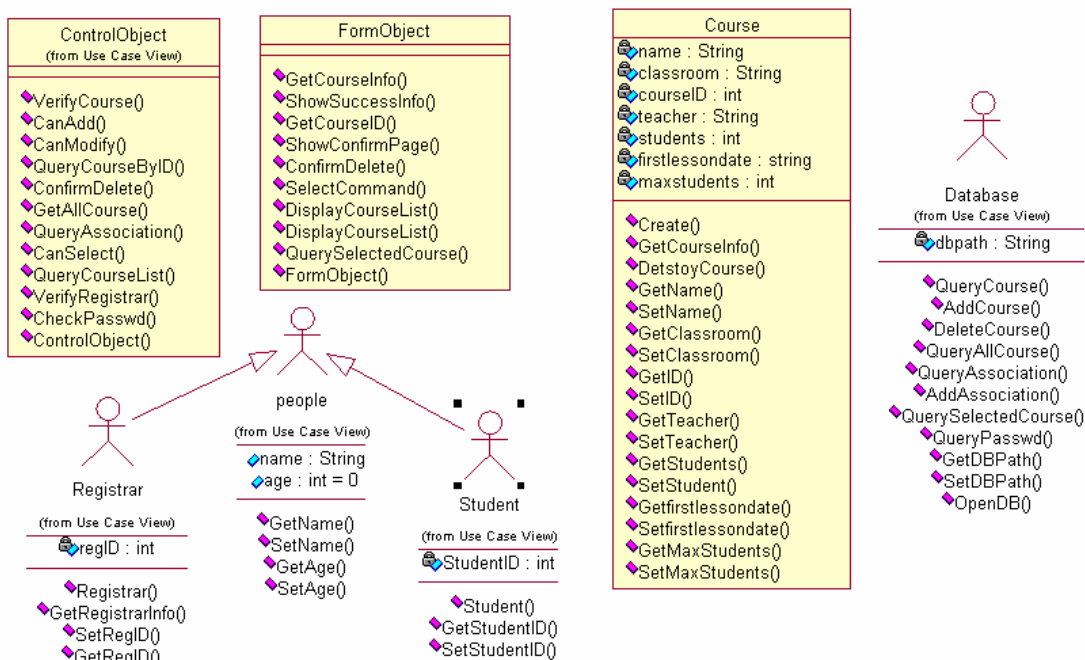


图 24

(2) 在类图的工具栏中选择“Creates a unidirectional association”图标，在类图窗口中，按下鼠标左键，从 student 指向 FormObject，则在 student 和 FormObject 之间添加了关联，然后调整一下类的位置。如图 25，为了显示方便，将类的属性和操作的“Show All Attributes”和“Show All Operations”设置为不选，将类的属性和操作隐藏起来。

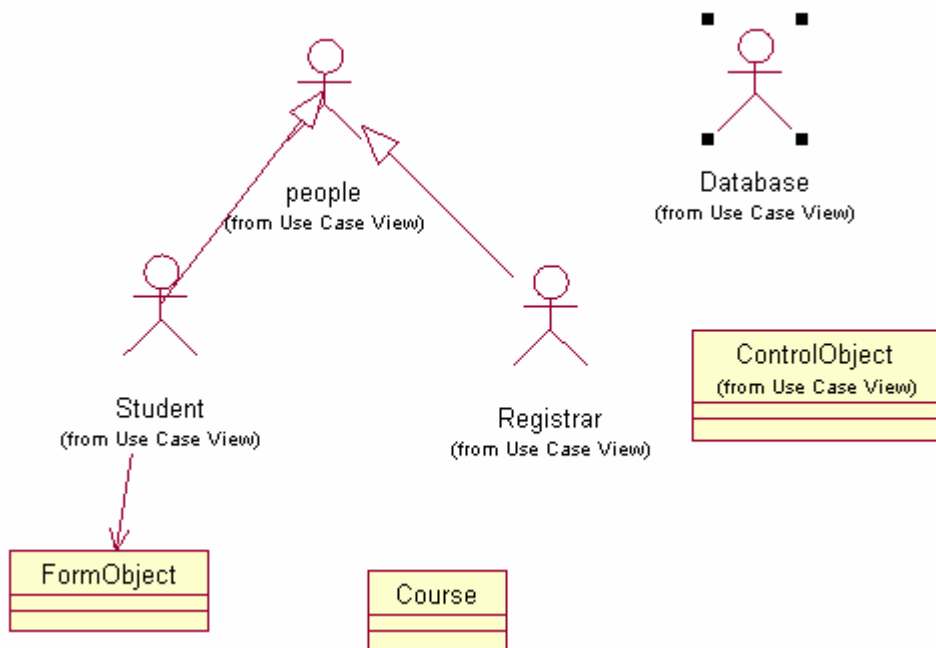


图 25 添加关联

(3) 可以给新添加的关联命名。在类图窗口中，选择 student 和 FormObject 之间的关联，按下鼠标右键，在弹出的菜单中选择“Open Standard Specification”，弹出如图 26 所示的对话框。

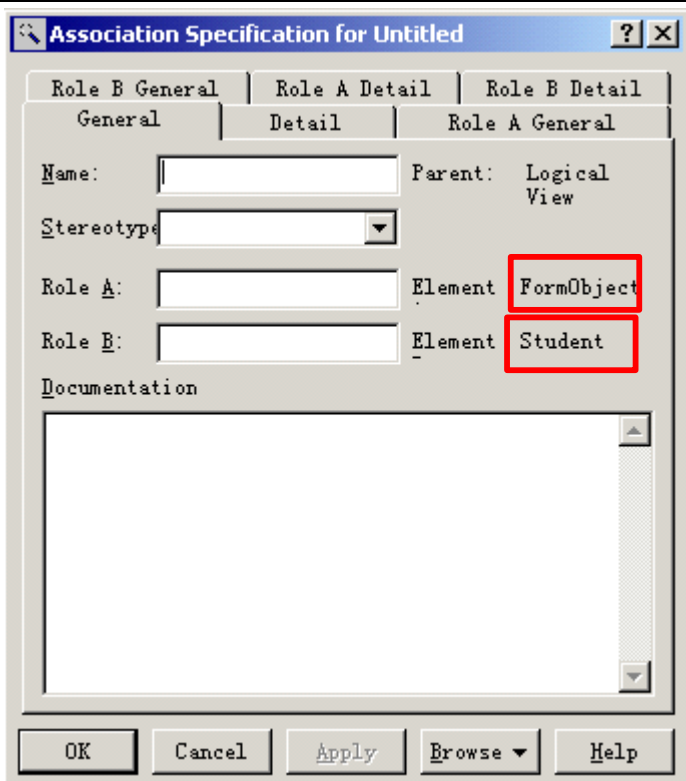


图 26

(4) 在该对话框中可以设置关联的属性。关联两端的对象，箭头指向的称为“Role A”，另一端称为“Role B”。如图 27 所示，可以将 Role A 记作“theFormObject”。

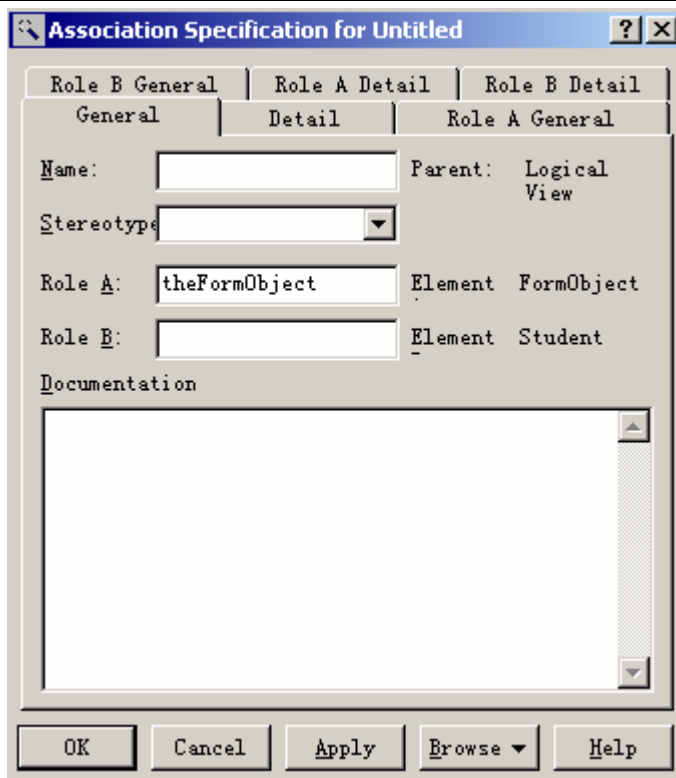


图 27 设置角色名

(5)如图 28 所示，在“Association Specificafi For…”对话框中点击“ Role B Detail”选项卡，将 Multiplic 设置成 1,再点击“Role A Detail” 选项卡，将 Multiplic 设置成 0..n。

(6)同理，将 Student 设置为 1，设置完成后，类图变成如图 29 所示的形式。

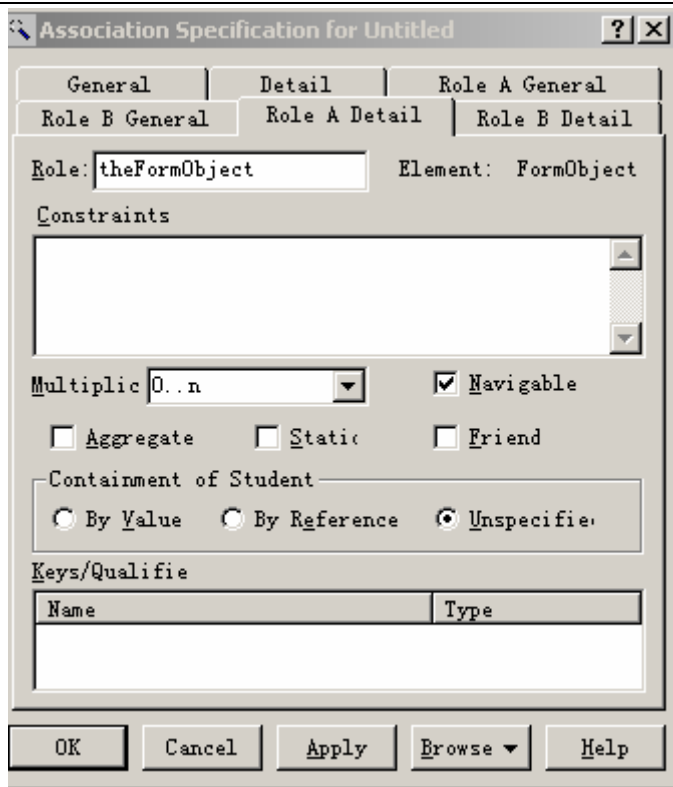


图 28 设置关联的角色的重数

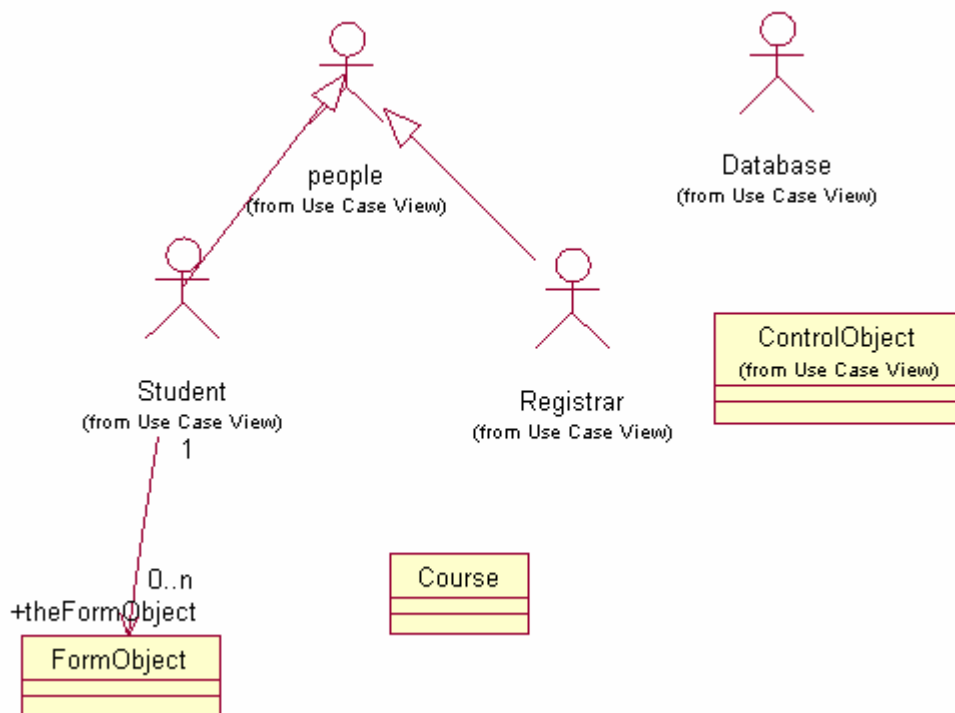


图 29 设置了关联的类图

现在使用图 29 和图 30 来说明如何在类图中添加及修改聚合关系。

图 29 建立一个新的和选课系统无关的类 text, 并假设一个界面对象中有 0 至多个 text 对象, 则界面类和 text 类之间存在聚合关系。选择 FormObject 和 text 之间的关联, 按下鼠标右键, 在弹出的菜单中选择“Open Specification”, 则弹出“Aggregation Specification For...”对话框, 点击“Role B Detail”选项卡, 如图 29 所示, 将对话框中的 Aggregate 项选中。

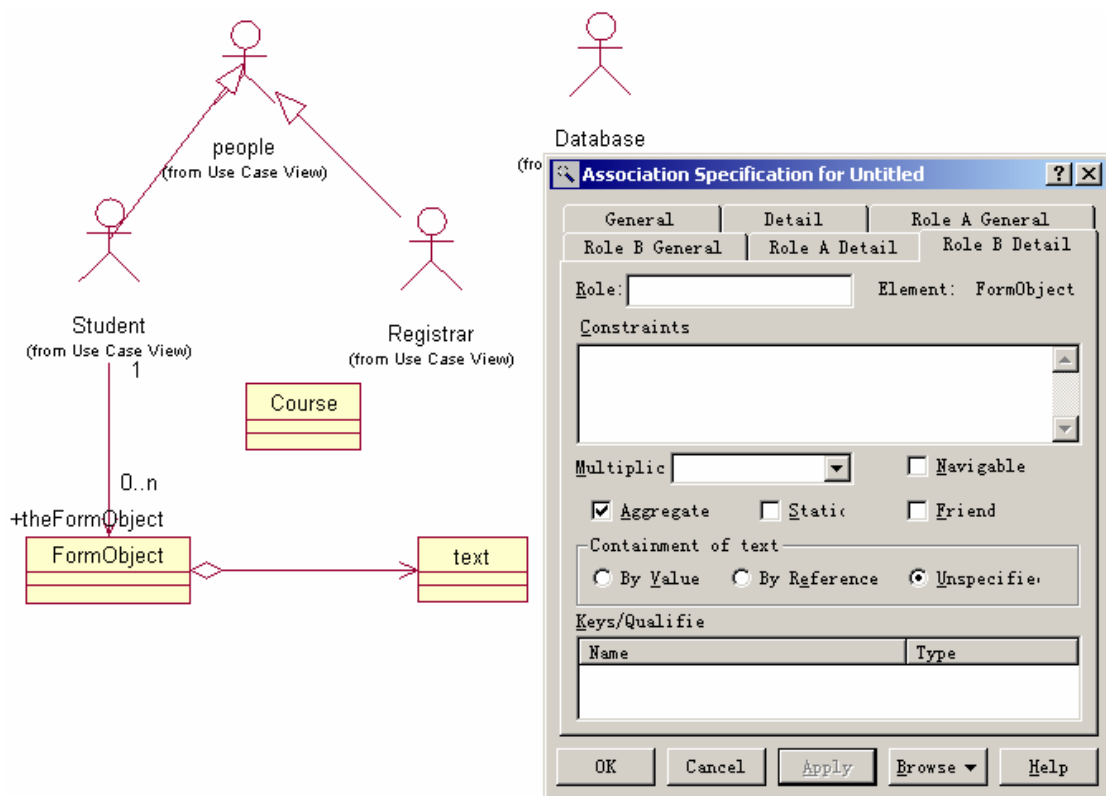


图 29 聚合关系的建立

将图 30 中的 Containment of text 设置为 By Value，按下“Apply”按钮，类图中聚合的标记变成图 30 所示的组合的标记。

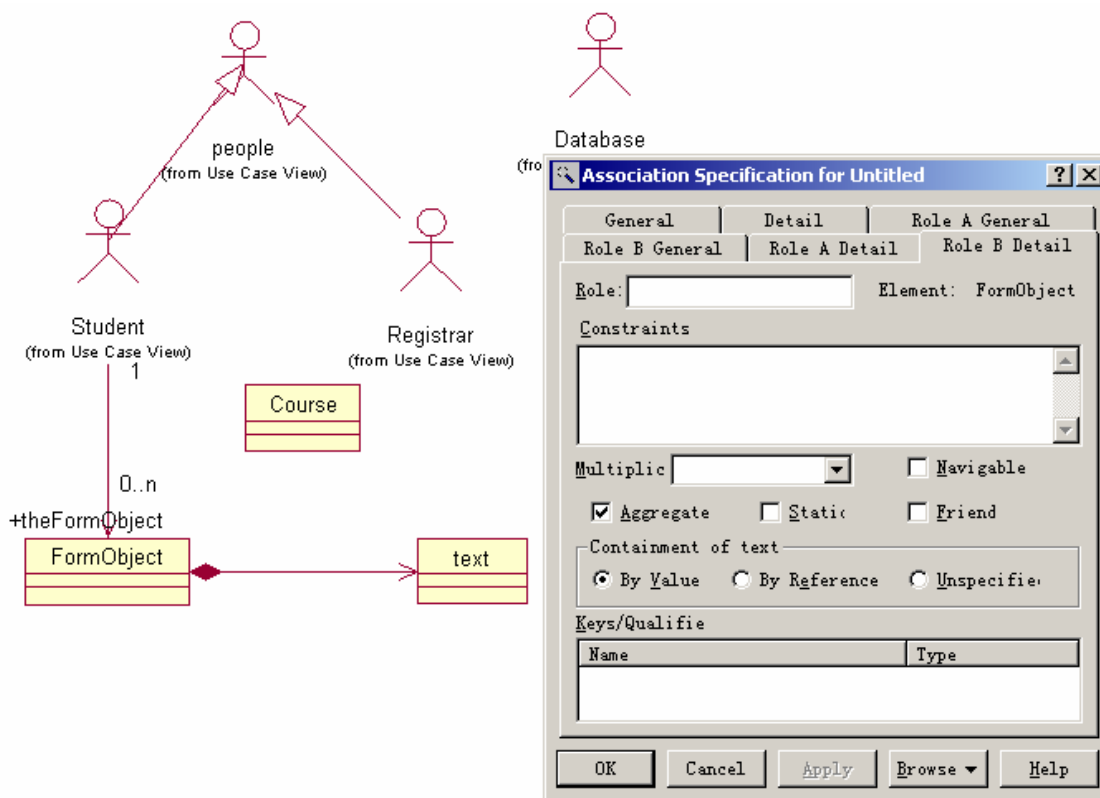


图 30 将聚合修改成组合

(7)回到图 29，使用图 29 和图 30 所示的方法，完成如图 31 所示的选课系统的完整类图。

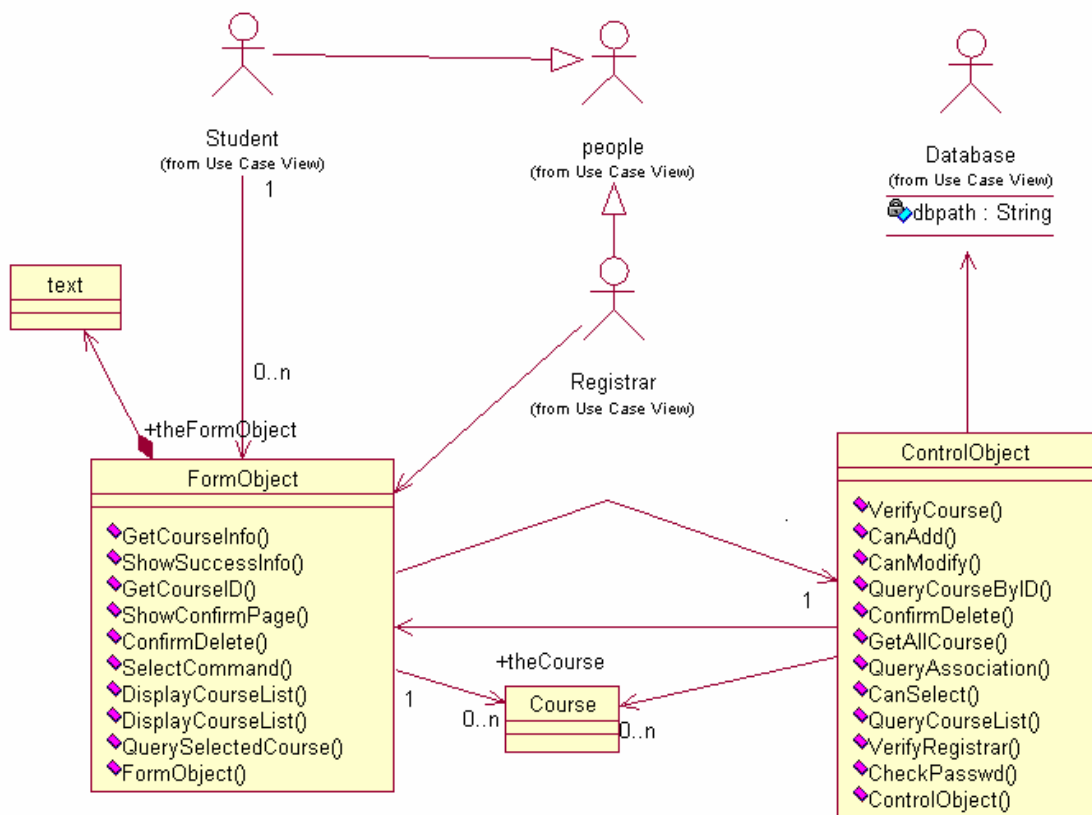


图 31 选课系统的完整类图

3.43 小结

现实世界中的对象存在各种各样的关系，分析这些关系，主要有：

(1) 继承关系。继承反映了自然界中特殊和普遍之间的关系，用它可以模拟客观世界。利用继承很容易在原有类上增加新的东西。

(2) 类之间的关联。关联关系描述了给定类的对象个体之间的语义连接，是两个类或多个类之间的一个关系。连接是关联的一个实例。聚合是表达主体部分关系的关联，组合是关

联的更强的形式。聚合和组合是关联的特殊形式。



本实验介绍了类的属性、操作，继承和类之间的关联、聚合、组合关系，它们的 UML 表示方法，绘制了选课系统的完整的框图。

选课系统的组件图和布局图设计和实现。

四、实验要求

1. 硬件基本配置：Intel Pentium 3 CPU 或以上级别的 CPU，大于 128MB 的内存
2. 软件要求：Windows98 / 2000 / Me/XP 操作系统，Office 97 / 2000 系统程序。
3. 实验学时：2 学时。

五、实验步骤

一、UML 组件图

组件视图 (component view) 包含模型代码库、可执行文件、运行库和其他组件的信息。组件是代码的实际物理模块，系统的组件图用来显示代码模块间的关系。

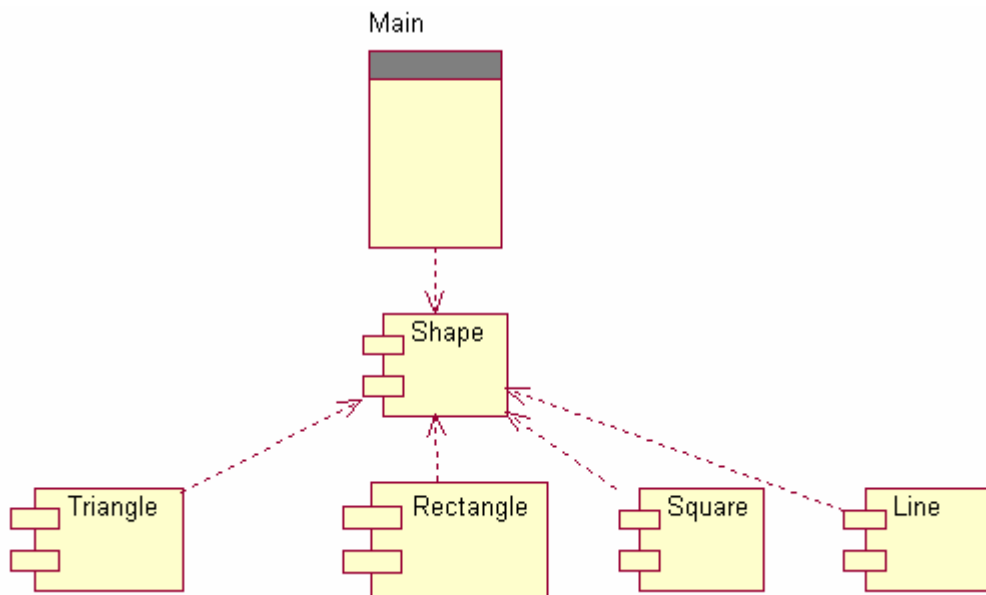
1 组件图的基本概念

组件图显示组件以及它们之间的依赖关系，一般来说，组件就是一个实际文件，可以有几种类型：

- (1) 源代码组件。一个源代码文件或者与一个包对应的若干个源代码文件。
- (2) 二进制组件。一个目标码文件，一个静态的或者动态的库文件。
- (3) 可执行组件。在一台处理器上可运行的一个可执行的程序单位，即所谓可执行程序。

组件图可以用来显示编译、链接或执行时组件之间的依赖关系，以及组件的接口和调用关系。在一个简单的画图的 C++ 程序中，包含 3 种类：main 类(主程序类)放在 main. cpp 中；shape 类(基类)放在 Shape. cpp 中。由它派生的 Line 类(负责画线的类)放在 Line. cpp 中，Triangle 类(负责画三角形的类)

放在 Triangle. cpp 中, Square 类(负责画正方形的类)放在 Square. cpp 中, Rectangle 类(负责画矩形的类)放在 Rectangle. cpp 中。把编译、链接和执行时上述程序组件之间的依赖关系放在一张组件图中,如图 1 所示。从图中可以看出, main. cpp 的编译依赖于 Shape. cpp、Line. cpp、Triangle. cpp、Square. cpp 和 Rectangle. cpp。



本实验选用选课系统的实例来说明组件图的制作方法。

2.1 选课系统组件图

在选课系统中,包括 MainProgram 类(主程序)、People 类、FormObject 类、ControlObject 类、Student 类、Registrar 类、Course 类和 DataBase 类。People 类是 Student 类和 Registrar 类的基类,所以 Student 类和 Registrar 类依赖 People 类。FormObject 类和 ControlObject 类都和 Course 类相关,FormObject 类和 ControlObject 类依赖 Course 类。ControlObject 类和 DataBase 类相关,ControlObject 类依赖 DataBase 类。

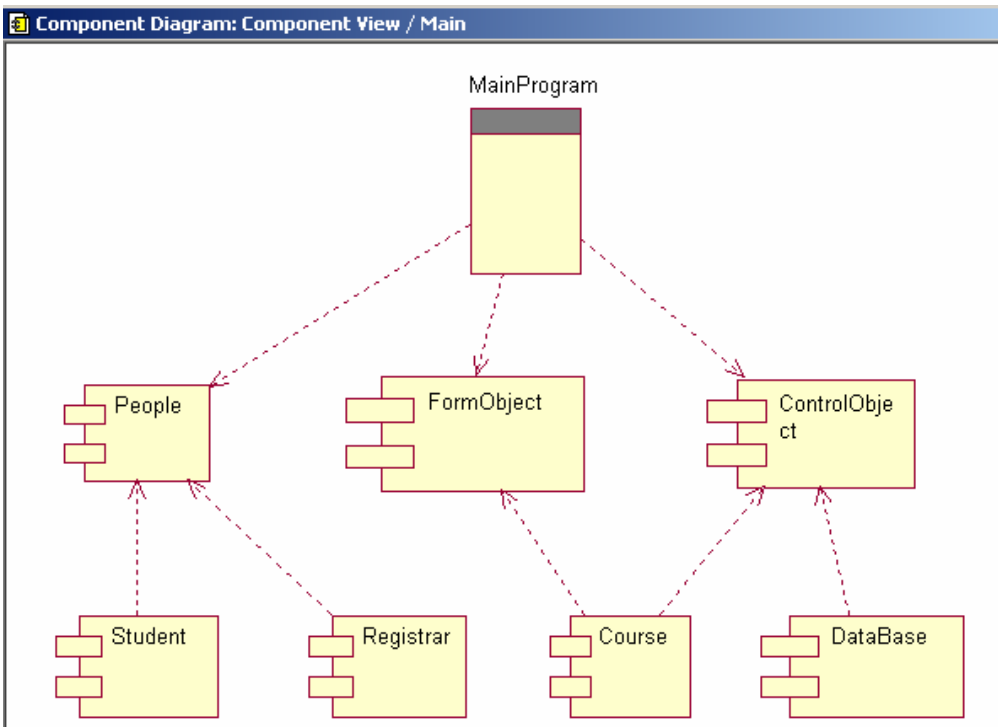


图 2 所示是网上选课系统的组件图。

组件图的创建

下面介绍在 Rose 中创建组件图的过程。

(1) 在组件视图中双击 Main 图，得到图 3 所示界面，为编辑组件图做好准备。

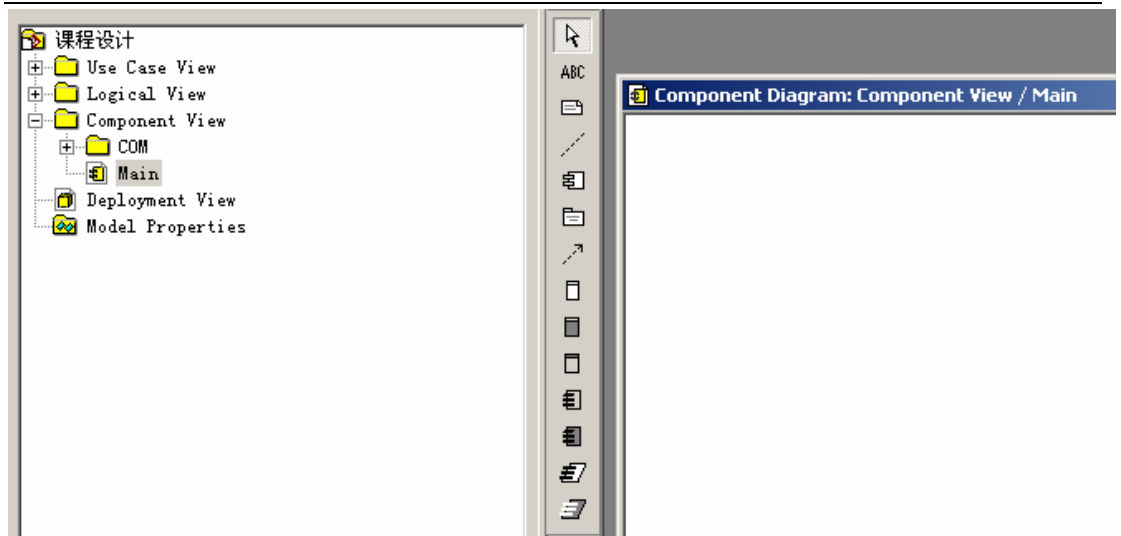


图 3 组件图

(2) 组件图窗口成为当前窗口时，工具栏变成图 4 所示的形式。



图 4 组件图窗口的工具栏

(3) 如图 5 所示，在浏览器中右键选择“component View”，在弹出的菜单中选择“New”，再在弹出的菜单中选择“component”，则创建一个新的组件。

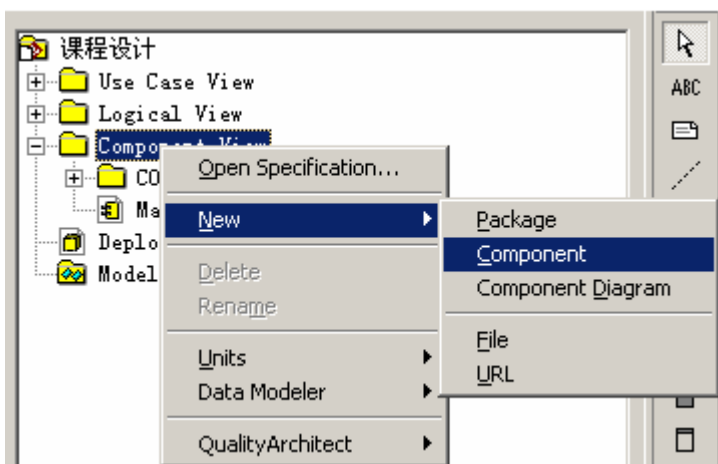


图 5 新建组件

- (1) 在浏览器中选择新创建的组件，按下鼠标右键，在弹出的菜单中选择“Open Specification”，弹出图 6 所示的对话框。

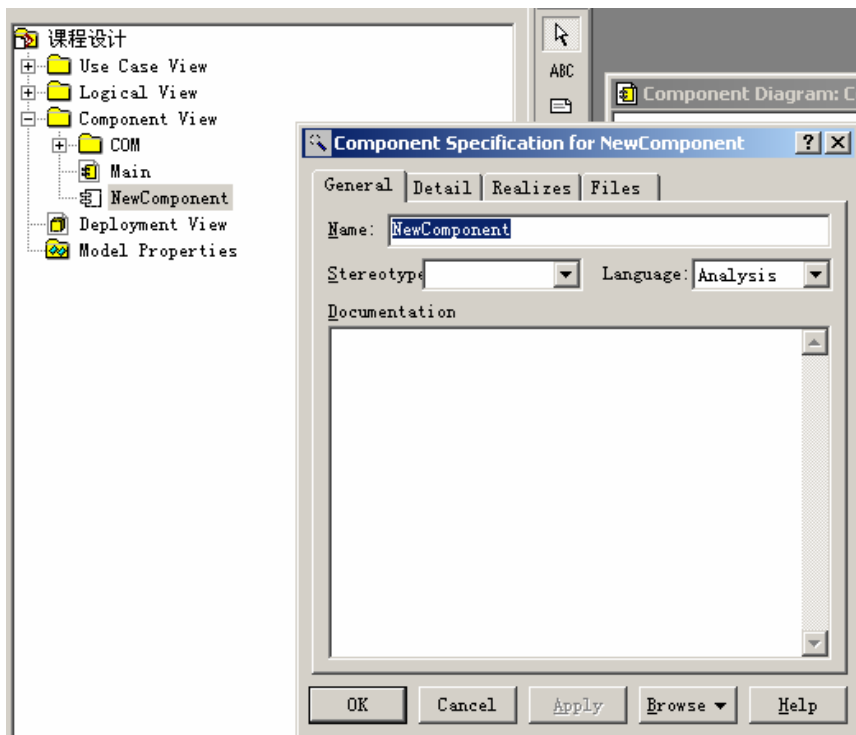


图 6

(5) 在对话框中可以修改组件名称、设置组件的版型、指定实现的语言(见图 7)。在这里,指定新创建的组件的名称为 MainProgram, 组件版型为 Main Program, 实现语言为 Java (或 Visual Basic)。

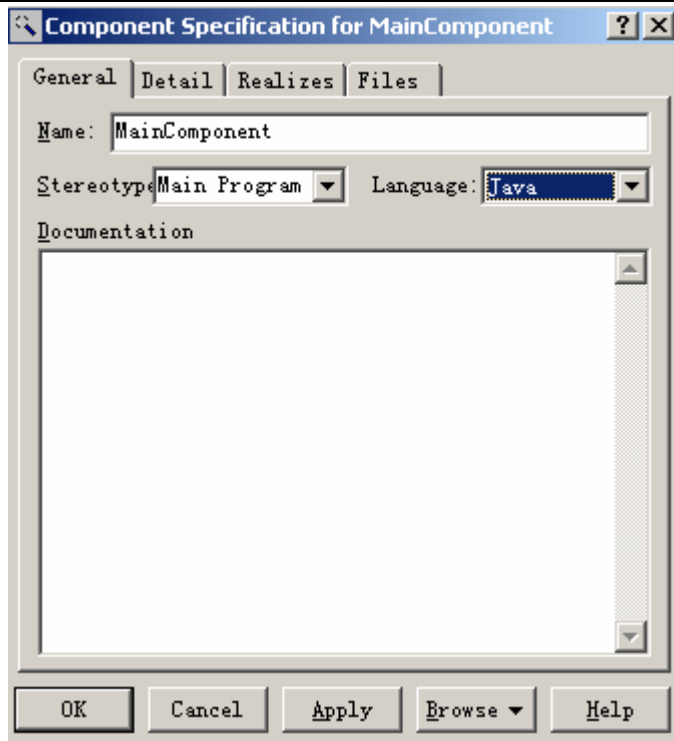


图 7 设置组件的属性

- (2) 组件图描述的是系统的实现，因此要指定实现组件功能的文件。点击 File 选项卡，在列表框中按下鼠标右键，在弹出的菜单中选择“Insert File”，弹出打开文件对话框。在对话框中键入 Main. java，按下“打开”按钮，则 Component Specification 对话框如图 8 所示。

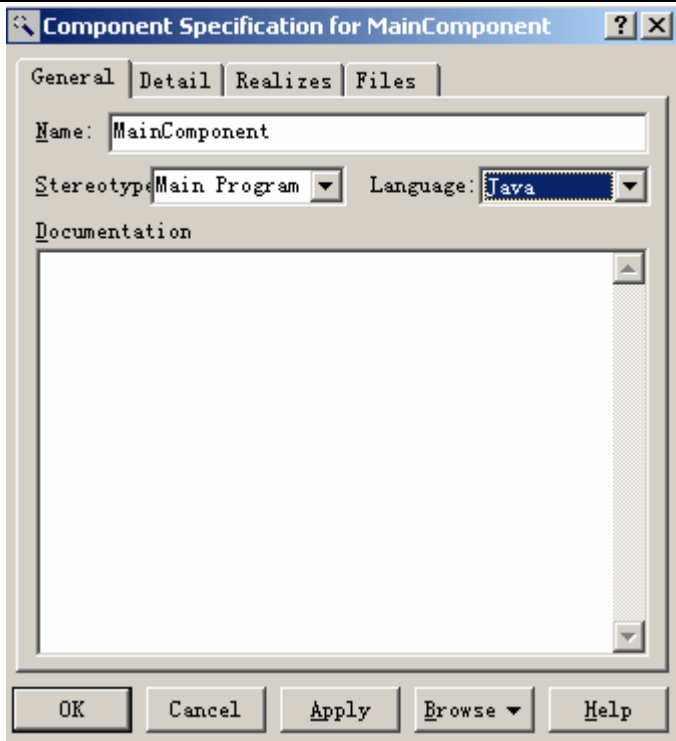
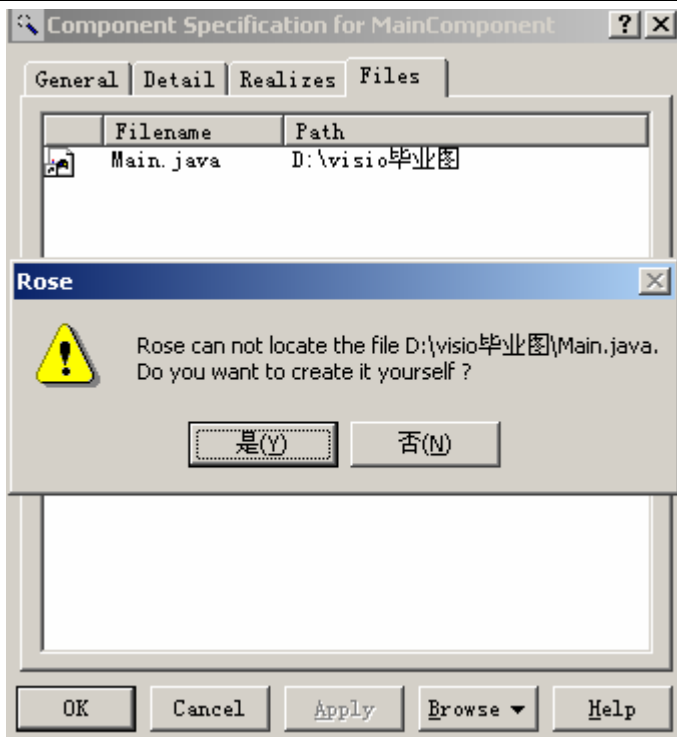
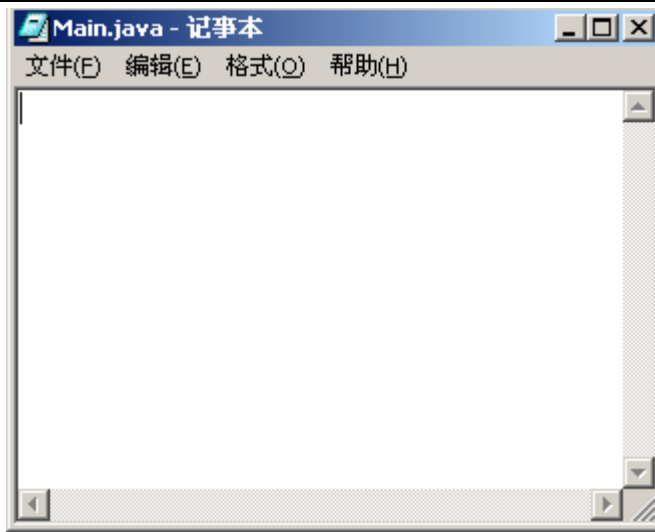


图8 与组件关联的文件

(3)选择 Main. java，双击，弹出如图9所示的对话框，让用户确认是否创建文件。



(4) 选择“是(Y)”，弹出记事本，可以在其中输入源程序。图 10 所示输入的源程序，输入完毕，将其保存。



(9) 在浏览器中选择 MainProgram，将它拖到组件图窗口中(见图 11)。

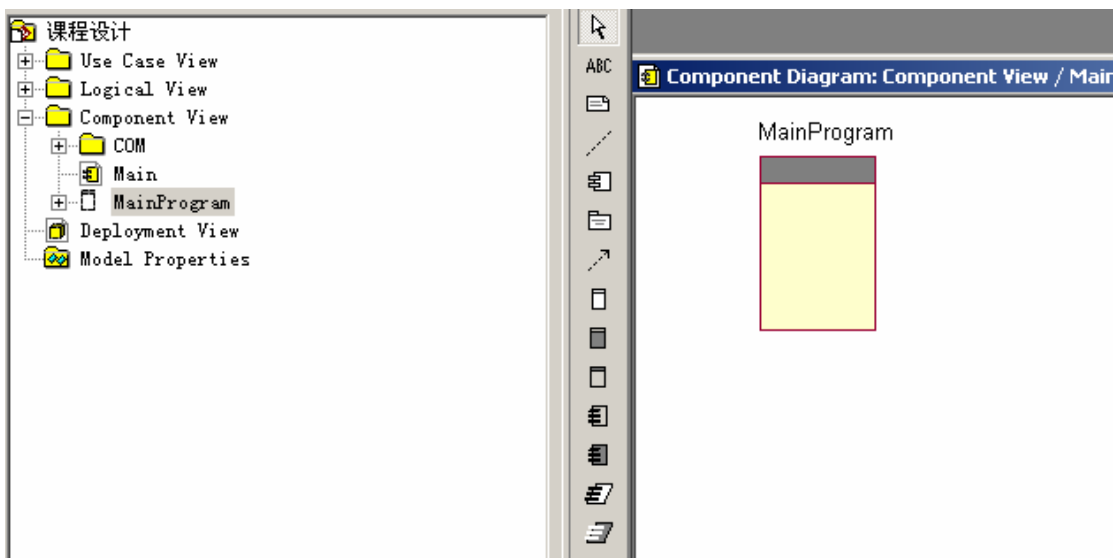


图 11

(10) 设置组件和类的关系。采用同样的方法创建 People 组件。从 Use case

View 中将 People 拖动到该组件上，可以建立该组件到 People 类的映射。或者可以打开图 12 所示对话框的 Realizes 选项卡，右键点击 class name，选择 assign，也可以建立映射关系。建立了关联后，相关类的图标上有红色的钩。

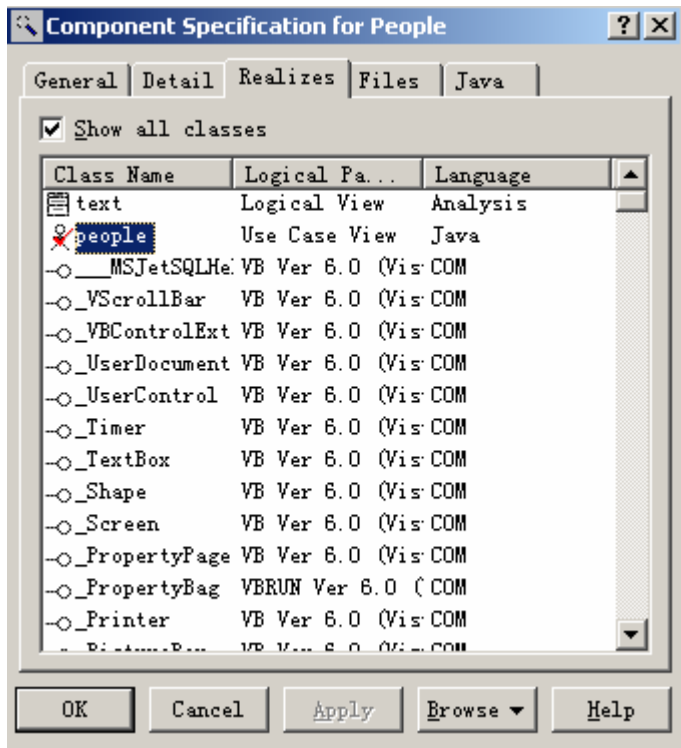


图 10-12 设置类和组件的关系

(11) 重复以上过程，加入其他各个组件。还需要在组件之间建立依赖关系，这种依赖关

系主要就是编译时的顺序。例如，知道 Student 和 Registrar 类都是派生自 People 类，所以编译时需要先找到 People.java 文件，然后才可以编译 Student 和 Registrar 的 java 文件。在其后编译的组件向需要提前编译的组件画出 Dependency 箭头，表示其依赖性。这样就完成了组件图的绘制，详见图 13。

Component Diagram: Component View / Main

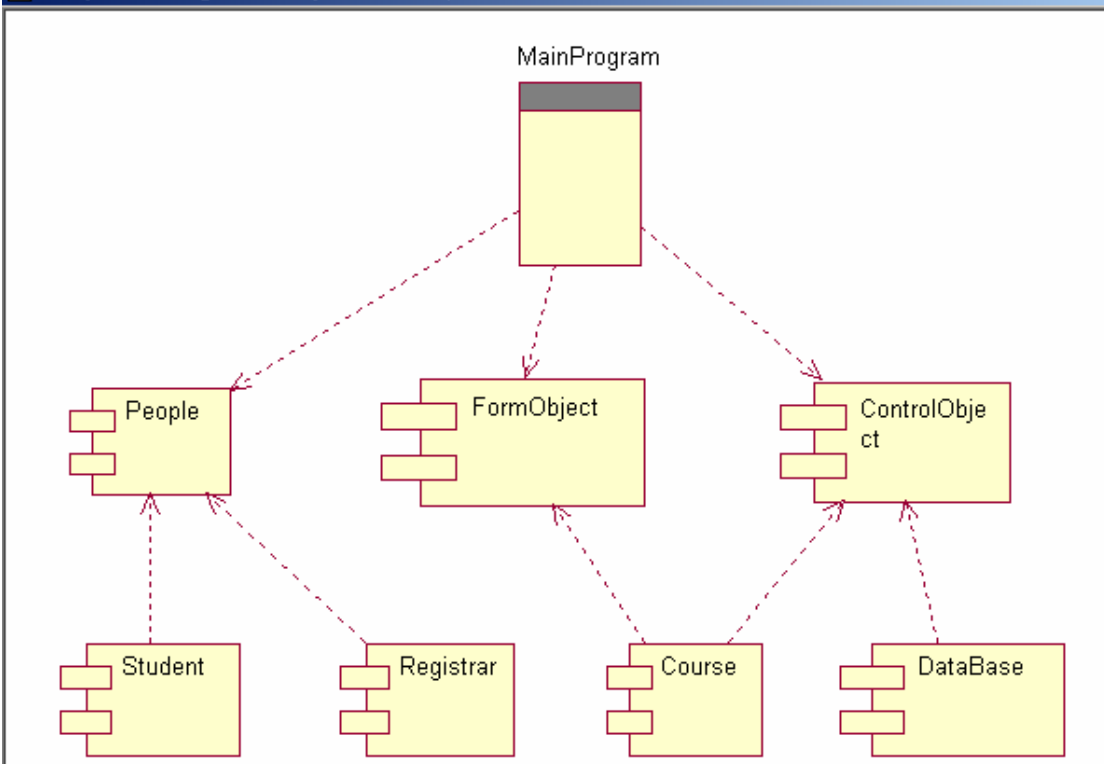


图 13 完整的组件图

3 小结

组件图显示代码本身的逻辑结构与系统的实现相关。组件的类型有：

- (1) 源代码组件。
- (2) 二进制组件。
- (3) 可执行组件。

本实验作了选课系统的组件图的设计和实现，很明显，这是源代码组件的例子。通过本实验的学习，希望同学们能够掌握组件的基本概念、组件图的设计和实现。



二、布局图

布局图表示系统的实际部署，与系统的逻辑结构不同，它描述系统在网络上的物理部署。

1. 布局图的概念

布局图用来描述系统硬件的物理拓扑结构以及在此结构上执行的软件。布局图可以显示计算节点的拓扑结构和通信路径、节点上运行的软件、软件包含的逻辑单元(对象、类等)。特别是对于分布式系统，布局图可以清楚地描绘硬件设备的配置、通信以及在各设备上软件和对象的配置。布局图中的节点代表某种计算构件，通常是硬件，譬如一个简单的设备或者传感器，也可以是一台主机。在布局图中，组件代表可执行的物理代码模块，例如一个可执行程序。逻辑上它可以与类图中的包或类对应起来。因此，布局图中显示运行时各个包或类在节点中的分布情况，如图 14 所示。由此可见，布局图是描述任何基于计算机的应用系统(尤其是基于 Internet 和 Web 的分布式计算系统)的物理配置(或者逻辑配置)的有力工具。

在布局图中，节点表示一个物理设备以及在其上运行的软件系统，例如一台 Unix 主机、一个 PC 终端、一台打印机、一个传感器等。节点之间的连线表示系统之间的通信路径，在 UML 中称为连接。通信类型则放在连接旁边的“<< >>”号之间，表示所用的通信协议或网络类型。需要注意的是，布局图中的各个节点的安置不受地理位置的限制，小则可以放在一间房间内，大则可以安置在地球上的不同国家或地区。两个节点之间的通信路径仅仅表明节点之间存在着联系，该连接可以采用不同的通信协议。

布局图包括以下要素：

- (1) 进程，在自己的内存空间执行的线程。
- (2) 处理器，任何有处理功能的机器。
- (3) 设备，任何没有处理功能的机器，如打印机。

2 实例

本实验介绍选课系统的布局图的创建。

2.1 选课系统布局图

图 14 是选课系统的布局图。从中可以看出，在服务器端使用了两台主机，一台作为数据库服务器，使用了 Oracle 服务器，另一台是同一局域网上的机



器，作为 HTTP 服务器，上面运行了 HTTP 服务器进程以及业务逻辑，同时还连接了打印机。而客户端则是在因特网上任意具有 Web 浏览功能的机器，客户端和 HTTP 服务器通过因特网连接，从 HTTP 服务器获取信息。当然，数据库服务器和 HTTP 服务器也可以放在同一台主机上，这也是一种合理的布局。

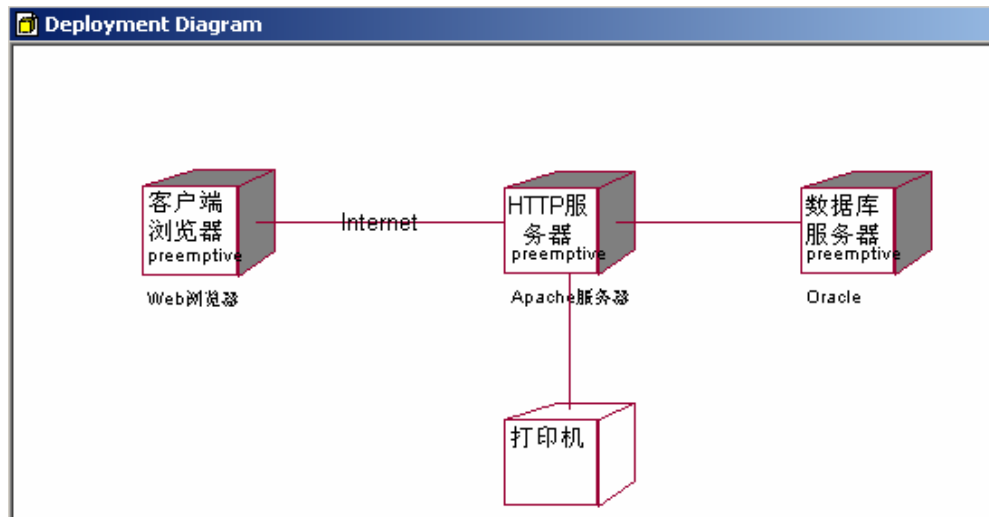


图 14 是选课系统的布局图。

2.2 布局圈的创建

布局图窗口的工具栏如图 11-2 所示



图 11-2 布局图窗口工具栏

在 Rose 中通过以下步骤建立布局图。

(1) 在浏览器中选择“Deployment View”，双击，弹出如图 15 所示的布局图窗口。

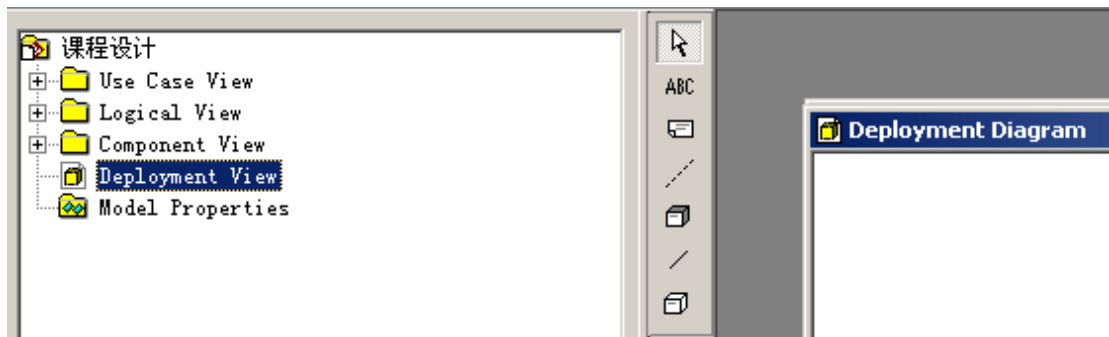


图 15 布局图窗口

(2) 在布局图窗口工具栏中选择 Processor 图标，放置到布局图窗口中，创建一个处理器。

如图 16 所示，将其名称改为“客户端浏览器”。

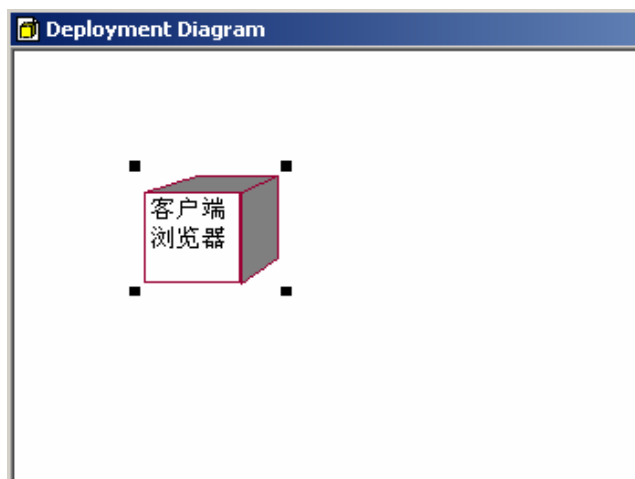


图 16

(3) 在布局图窗口中选择“客户端浏览器”对象，按下鼠标右键，在弹出的菜单中选择“Open Specification”，弹出如图 17 所示的对话框，设置这个对象的名称和版型。

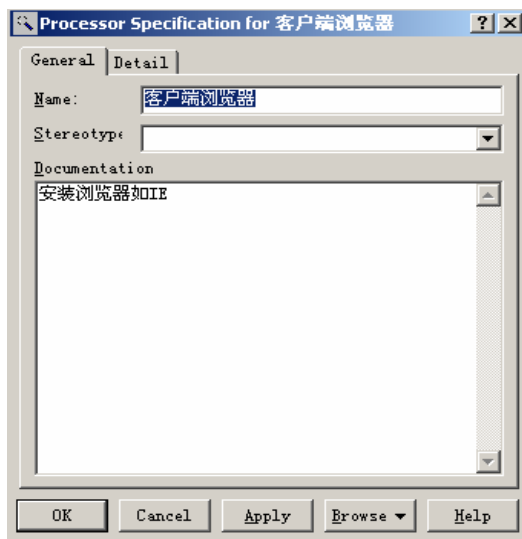


图 17 设置处理器属性

(4) 选择“Detail”选项，在“Processes”列表框内按下鼠标右键，在弹出的菜单中选择“Insert”，设置处理器上运行的进程（见图 18）。

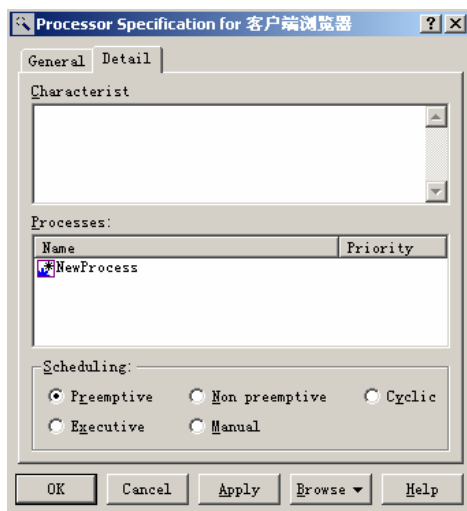


图 18

(5) 双击新添加的进程，弹出图 19 所示的对话框，将进程的名称改成“Web 浏览器”。

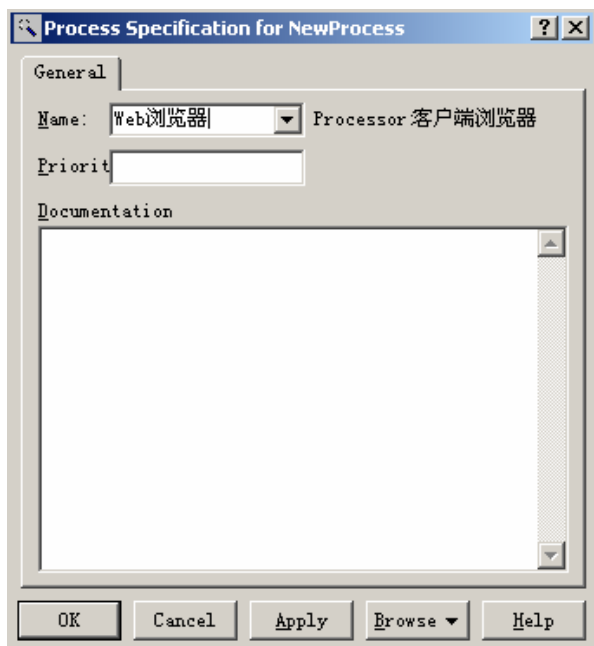


图 19

(6) 返回布局图窗口，采用同样的方法在图中添加“HTTP 服务器”，在工具栏中选择“Connection”图标，在“Web 浏览器”和“HTTP 服务器”之间添加连接关系(见图 20)。

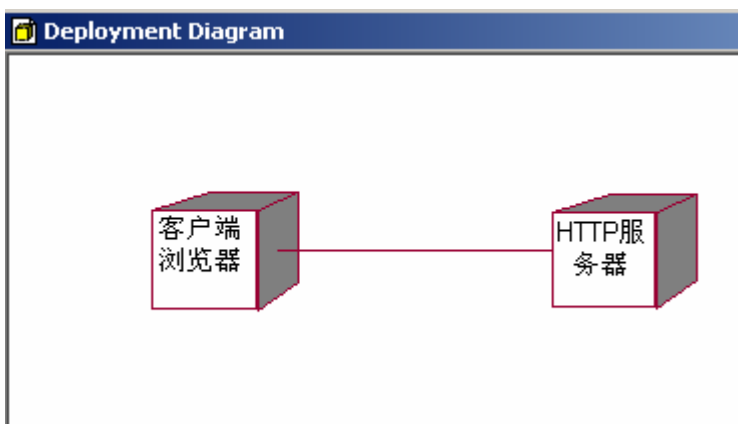


图 20

(7) 在布局图窗口中右键单击连接图标，在弹出的菜单中选择“Open Specification”，弹出图 21 所示的对话框，设置连接的名称和版型。

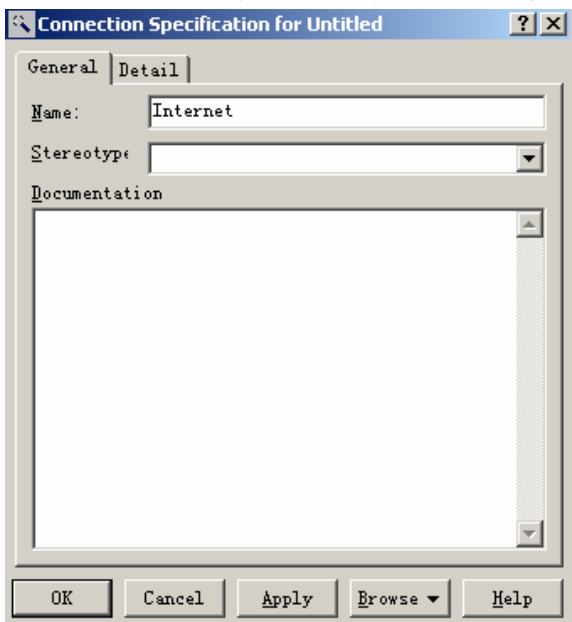


图 21

(8) 按下“OK”按钮，返回布局图窗口。

(9) 选择客户端浏览器，按下鼠标右键，在弹出的菜单中将“Show Scheduling”、“Show Processes”选中，布局图变成图 22 的形式。

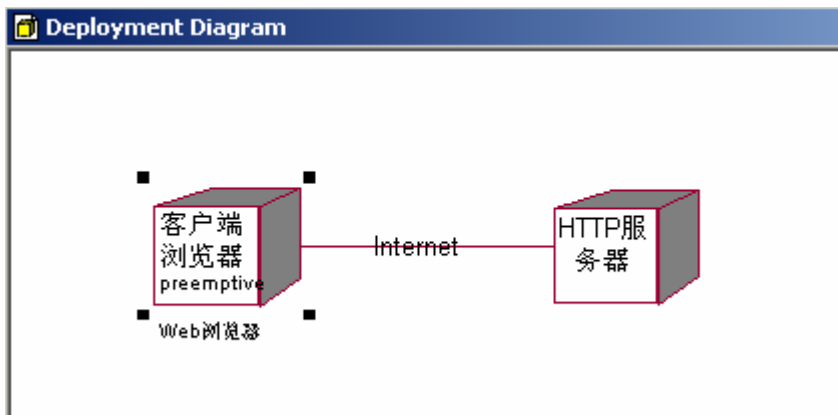


图 22 显示进程

(10) 在 HTTP 服务器上增加进程 (见图 23)。

(11) 在布局图上增加数据库服务器和打印机，打印机连接在 Apache 服务器上，完成整个布局图的设计 (见图 24)。

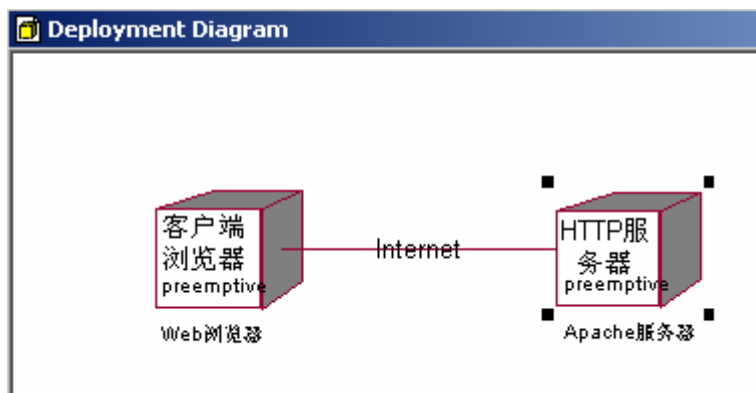


图 23 在 HTTP 服务器上增加进程

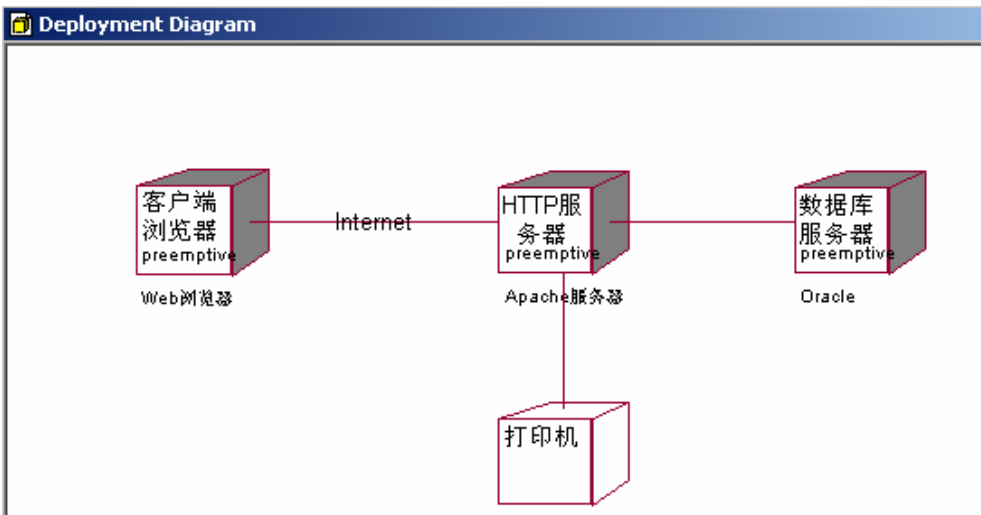


图 24 完整的布局图

3. 小结

布局图显示组件系统的硬件分布，与系统的实现相关。布局图包括的要素有：

- (1) 处理器。
- (2) 进程。
- (3) 设备。

本实验介绍了选课系统的布局图的设计和实现以及典型的布局图。通过本实验的学习，应该能够掌握布局图的设计和实现。



附录 6 课程设计题目示例

题目一 “教务管理系统之子系统——系内课程安排”

1. 系统简介

每学期的期中，学院教务处分别向各个系发出下学期的教学计划，包括课程名、课时、班级类别（本科、专科、高职）、班号等；系教学主管人员根据教学任务和要求给出各课程的相关限制（如：任课教师职称、和班数、最高周学时数等）；任课教师自报本人授课计划，经所在教研室协调确认，将教学计划上交系主管教学计划的主任，批准后上报学院教务处，最终有教务处给出下学期全系教师的教学任务书。

假设上述排课过程全部为人工操作，现要求改造为能利用计算机实现的自动处理过程。

1. 限定条件

- (1) 每位教师的主讲门数不超过 2 门/学期：讲师以下职称的教师不能承担系定主课的主讲任务。
- (2) 系级干部的主讲课时不能超过 4 学时/周。
- (3) 本学期出现严重教学事故的教师不能承担下学期的主讲任务。
- (4) 本系统的输入项至少应包含 3 个：教务处布置的教学计划、系教师自报的讲课计划和系定的有关讲课限制条件。
- (5) 本系统的输出项至少应包含 2 个：教务处最终下达的全系教师教学任务书和系各教学班一学期的课程表（可不包含上课地点）。

题目二、“学校教材订购系统”

1、系统简介

本系统可细化为两个子系统：销售系统和采购系统

销售系统的工作过程为：首先由教师或学生提交购书单，经教材发行人员审核是有效购书单后，开发票、登记并返给教师或学生领书单，教师或学生即可去书库领书。

采购系统的主要工作过程为：若是脱销教材，则登记缺货，发缺货单给书库采购人员；一旦新书入库后，即发进书通知给教材发行人员

以上的功能要求在计算机上实现。

2、技术要求和限制条件



- (1) 当书库中的各种书籍数量发生变化（包括领书和进书时），都应修改相关的书库记录，如库存表或进/出库表。
- (2) 在实现上述销售和采购的工作过程时，需考虑有关单据的合法性验证
- (3) 系统的外部项至少包含三个：教师、学生和教材工作人员。
- (4) 系统的相关数据存储至少包含 6 个：购书表、库存表、缺书登记表、待购教材表、进/出库表。

题目三、“机票预订系统”

1、系统简介

航空公司为给旅客乘机提供方便，需开发一机票预定系统。各旅行社把预定机票的旅客信息（姓名、性别、工作单位、身份证号码、旅行时间、旅行目的地等）输入到该系统，系统为旅客安排航班。当旅客交付了预定金后，系统印出取票通知和帐单给旅客，旅客在飞机起飞的前一天凭取票通知和帐单交款取票，系统核对无误即印出机票给旅客。此外航空公司为随时掌握各航向飞机的承载情况，需定期进行查询统计，以便适当调整。

2、技术要求及限定条件

- (1) 在分析系统功能时要考虑有关证件的合法性验证（如身份证、取票通知、交款发票等）。
- (2) 对于本系统还应补充以下功能：
 - (1) 旅客延误了取票时间的处理
 - (2) 班机取消后的处理
 - (3) 旅客临时更改机票班次的处理
- (3) 系统的外部项至少包含三个：旅客、旅行社和航空公司。

题目四：“学校内部工资管理系统”

1、系统简介

假设学校共有教职工约 1000 人，10 个行政部门和 8 个系部。每个月 20 日前各部门（包括系、部）要将出勤情况上报人事处，23 日前人事处将出勤工资、奖金及扣款清单送财务处。财务处于每月月底将教职工的工资表做好并将数据送银行。每月初（3 日前）将工资条发给各单位。若有员工调入、调出、校内调动、离退休等数据变化，则由人事处通知相关部门和财务处。



2、技术要求及限定条件

- (1) 本系统的数据存储至少包含：工资表、部门汇总表、扣税款表、银行发放表
- (2) 除人事处、财务处外，其他只能部门和系部名称可简化，如系 1，系 2.....等
- (3) 工资、奖金、扣款细节可由学生自定

题目五、“实验室设备管理系统”

1、系统简介

每学年要对实验室设备使用情况进行统计、更新，其中：

- (1) 对于已彻底损坏的作报废处理，同时详细记录有关信息。
- (2) 对于有严重问题（故障）的要即使修理，并记录修理日期、设备名、修理厂家、修理费、责任人等。
- (3) 对于急需但又缺少的设备需以“申请表”的形式送交上级领导请求批准购买。新设备购入后要立即进行设备登记（包括类别、设备名、型号、规格、单价、数量、购置日期、生产厂家、购买人等），同时更新申请表的内容。
- (4) 随时对现有设备及其修理、报废情况进行统计、查询，要求能够按类别和时间段（某日期之前）查询。

2、技术要求及限定条件

- (1) 所有工作由专门人员负责完成，其他人不得任意使用。
- (2) 每件设备在作入库登记时均由系统按类别自动顺序编号，形成设备号；设备报废时要及时修改相应的设备记录，且有领导认可。
- (3) 本系统的数据存储至少应包含：设备记录、修理记录、报废记录、购买申请。
- (4) 本系统的输入项至少包含：新设备信息、修理信息、申请购买信息、报废信息、具体查询统计要求。

本系统的输出项至少包含：设备购买申请表、修理/报废注销/设备资金统计表。

题目六、分房管理系统

某大学拟开发一个计算机分房管理系统，要求系统具有分房、调房、退房和咨询统计等功能。房产科把用户申请表输入系统后，系统首先检查申请表是否合理，对不合法申请表拒绝接受；对合法表根据类型进行处理。



分房申请,根据申请者情况(年龄、工龄、职称、职务、家庭人口等)计算其分数,高于阈值的进行排队。分房时,读空房文件,把好房优先分给排在分房队列前面的人,并将房屋信息与申请者一起写入住房文件中。

若为退房申请,从住房文件中删除有关信息,并将房屋信息计入空房文件。

若是调房申请,根据申请者情况计算分数,并进行以上退房和分房处理。

系统可进行房屋信息维护(输入、删除、排序),以及各种住房信息统计、查询、打印等。

题目七 基金管理系统

为实现政务公开,某省拟采用网上自然基金申报和审批,需开发一套包括基金申报(使用说明、文档提交、申报情况查询)、基金信息公告(基金公告、基金新闻、最新立项)、网上审批与鉴定(专家定量网上审批、项目鉴定)、基金项目管理(发布申报计划、批复申请、查询申请项目等)与推荐优秀基金项目等功能的办公网站。

要求具有基金申报管理(申报文档提交,申报情况查询等),信息公告维护(增加,撤除和修改等),审批流程管理(管理员可指定审批流程,对需执行审批的专家显示审批流程,进行审批提示),项目鉴定管理(管理员可指定鉴定流程,对需执行鉴定的专家显示审批流程,进行鉴定提示)等功能。

题目八 汽车俱乐部管理系统

为实现汽车俱乐部行业信息的规范化管理,某软件开发商要求开发一汽车俱乐部管理系统,面向俱乐部工作人员进行会员信息的管理。主要功能包括:用户(俱乐部工作人员)身份验证和用户权限组织;提醒服务(提醒各种该执行的车务活动);会员管理(会员信息、会员费收缴记录录入,查询,修改);车务管理(新车上牌代办,保险代办,验证代办,落户代办等);救援服务(呼救记录,救援情况记录以及救援车辆管理);旅行服务(旅行信息记录、修改、查询);统计报表、图表(所有当日、当月、当年向会员提供的各种服务信息及其收费的详细情况列表及汇总);系统管理(设置提醒提前日期,用户管理等);俱乐部管理(俱乐部部门、工作人员、救援车辆管理);系统帮助。

题目八、学生学籍管理系统

功能说明

本学生学籍管理系统分为学生信息管理、学生成绩管理、考试管理和用户管理四个功能



模块。

(1) 学生信息管理模块主要是对学生信息（如学号、姓名、性别、家庭住址等）进行管理。本模块又分为三个子模块：

- ◇ 学生信息录入
- ◇ 学生信息查询
- ◇ 学生信息修改/删除

其中，学生信息查询不需登录即可使用，信息录入和修改/删除模块则需要先登录后才能使用。

(2) 学生成绩管理模块主要是对学生成绩进行管理。本模块又分为五个子模块：

- ◇ 学生成绩录入
- ◇ 学生成绩查询
- ◇ 学生成绩修改/删除
- ◇ 计算总分与排名
- ◇ 目标分管理

其中，学生成绩查询不需登录即可使用，其他模块则需要先登录后才能使用。

(3) 考试管理模块主要对考试进行如下管理：

- ◇ 添加新考试
- ◇ 对已有的考试信息进行修改/删除
- ◇ 对考试科目进行添加/删除

本模块需要先登录后才能使用。

(4) 用户管理模块主要对使用本系统的用户进行如下管理：

- ◇ 添加新用户
- ◇ 对已有的用户进行更改用户名、密码和权限（管理员或普通用户）等操作
- ◇ 删除用户

本模块只有管理员才能使用，普通用户不能进入。



附录 7 Microsoft Visio 软件的使用

Visio 是目前世界上最常用的商业绘图软件，Visio 的用途通俗的说，就是把我们要表达和阐述的内容，结合图形更加形象的描绘出来。Visio 是一个图形化管理软件，它可以将管理活动以图形的方式展示出来，便于跟踪管理。然而，传统方式在处理图形化的工作过程中，既费时，又费力，且重复劳动影响工作效率与质量。Visio 不但将绘图过程简化，而且融入了图形化管理的规则。所以，它越来越成为办公管理活动中不可缺少的工具。Visio 专门提供给工程技术人员或一般商业人士使用，是一种快速的绘图软件，能够让你轻松做出专业化、高质量的图形或图表。

步骤 1: VISIO 基础入门

打开 Visio 就会看到一个和 Office 非常相似的界面，Visio 这个软件就像是个强大的模板，它定义好了很多领域的图形模板，让你游刃有余的拖放就能组成一个强大的营销图表、网络结构等等。

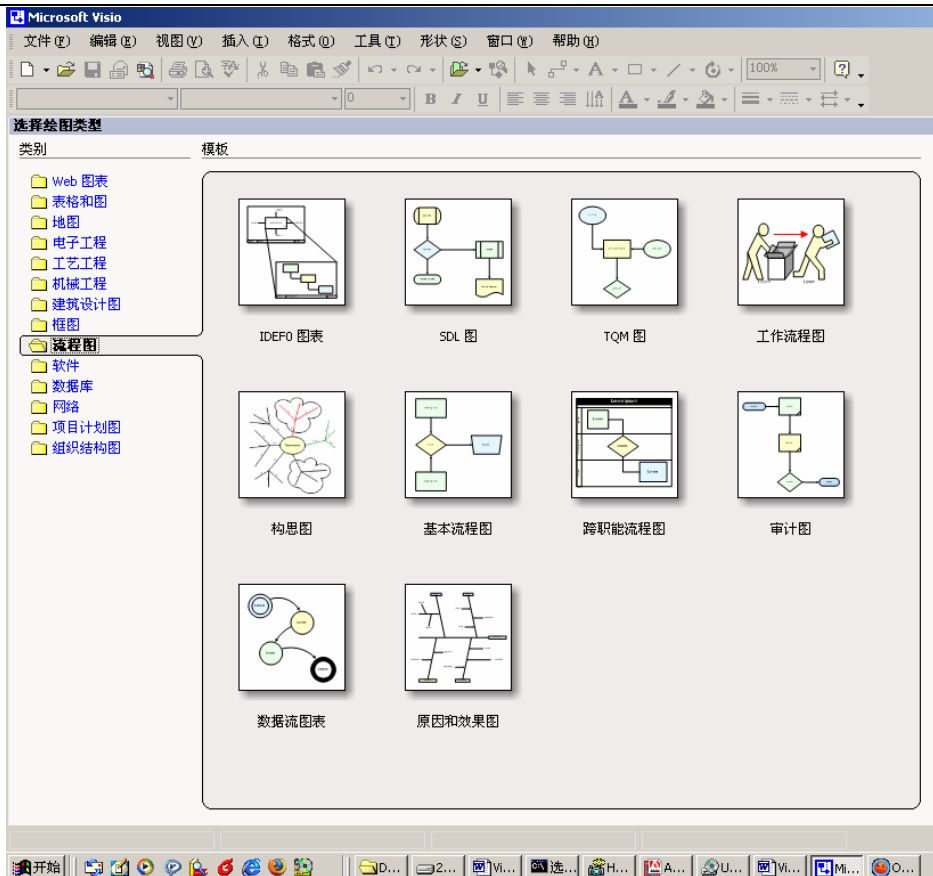
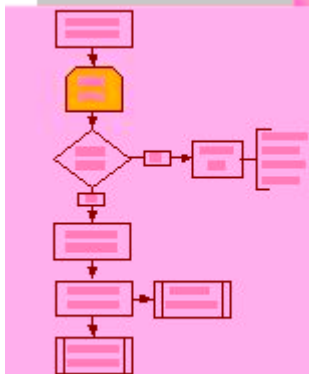


图 1

Visio 包括了很多类型的图表模板，由于篇幅的关系，这里只对流程图、物理网络图和组织结构图做一下简单的说明。

流程图：



流程图是对流程的直观表示。

在流程图文件夹中可找到用于创建各类流程图（如审计图、原因和效果图以及工作流程图）的模板和模具。什么人需要使用流程图以及如何使用？通过使用流程图，着重测试新来的应聘人如何处理棘手的任务，经理人员可在面试过程中筛选出胜任的人选。保险公司可以使用流程图来记录风险评估流程。会计师可以使用流程图来描述财务管理、资金管理和财务库存流程。

物理网络图:

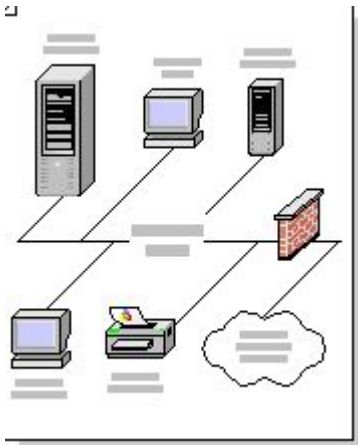


图 3

物理网络图表示组织的网络，包括电缆线路和物理硬件。

在网络文件夹中可找到基本网络模板，通过它打开的模具可用来创建各种物理网络图。什么人需要使用物理网络图以及如何使用？设备经理可将物理网络图并入灾难恢复计划、布线图以及包含有关公司资产及其相关成本信息的文档。网络经理可使用物理网络图来显示产品在组织内的使用分布状况。组织中的任何人都可通过参考网络图来找到他们能够连接的设备，例如打印机和复印机。

组织结构图：

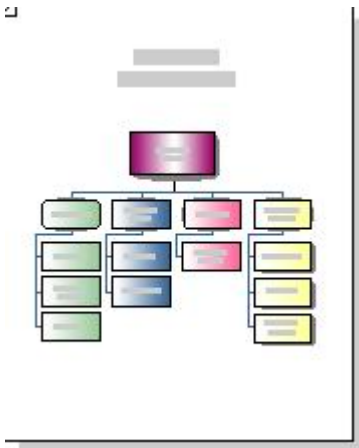


图 4

组织结构图显示了汇报关系以及雇员分属的部门。

在组织结构图文件夹中可找到用来创建组织结构图的模板和模具。什么人需要使用组织结构图以及如何使用？人力资源专员可创建组织结构图，并将其张贴到公司的 **Intranet** 中。项目经理可使用组织结构图来显示团队结构和任务细分状况，以便生成项目日程安排。经理人员可使用组织结构图来直观地表示如何进行部门重组或估计人员配置的需要。

提示：你还可以查找 **Visio** 中自带的图表示例。在文件菜单上指向新建，单击浏览绘图示例，双击一个文件夹，然后打开绘图示例，里面有很多做好的图表事例，非常适合初学者使用，模仿。



步骤 2: 创建一个图表的实例

举流程图的例子。请跟着下面的操作步骤来进行操作。

具体过程如下: 选择一个模板-->移动形状和调整形状的大小-->添加文本-->连接形状-->设置形状格式-->完成和使用图表。

A) 选择一个模板

打开 Visio 界面以后选择一个流程模板, 如图:

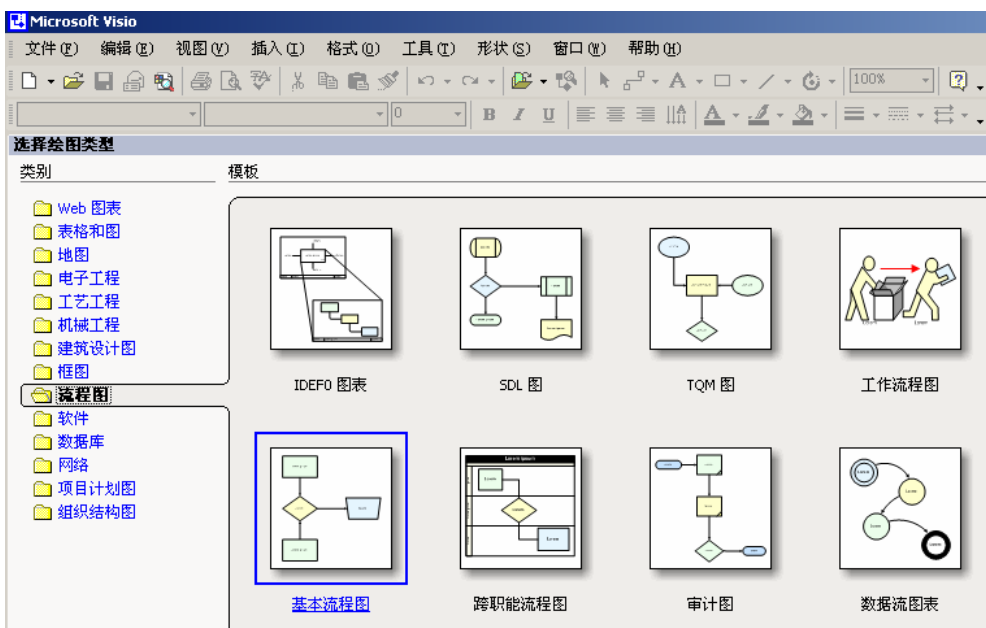


图 5 选择一个基本流程图模板

用模板开始创建 Microsoft Visio 图表。模板是一种文件, 它打开一个或多个模具, 模具中包含创建图表所需的形状。打开模板后, 将形状从模具中拖至绘图页, 添加到图表中。如果需要其他形状, 您可以打开其他模具或使用查找形状功能查找。

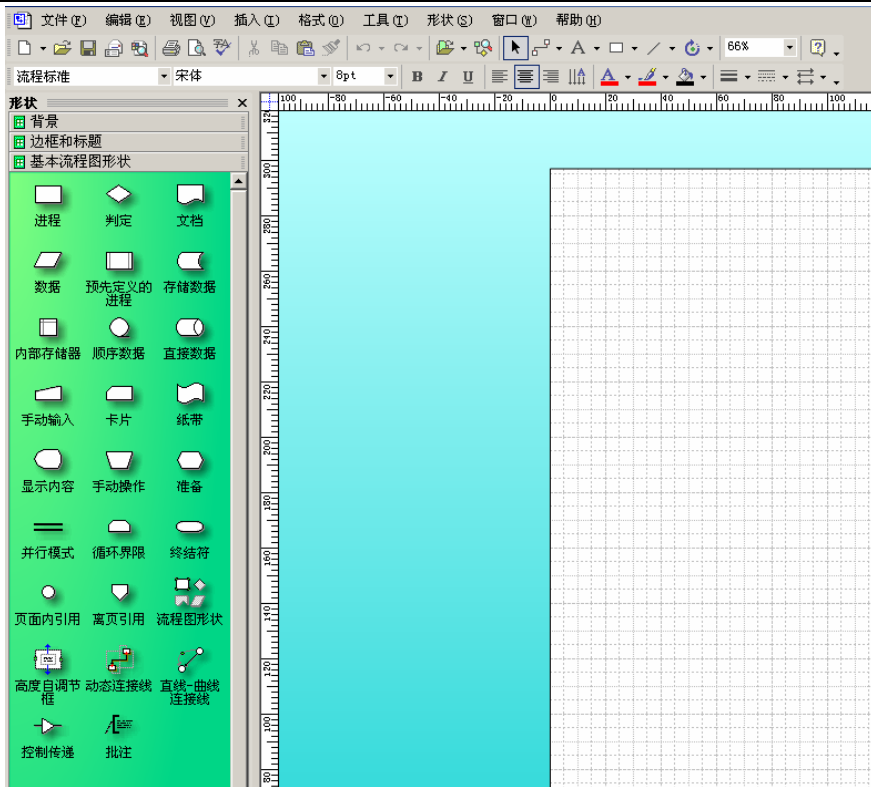


图 6

B) 移动形状和调整形状的大小

很少有人能在创建图表时能一次性正确地定位各个形状。如果您属于大多数 Microsoft Visio 用户，您会向图表添加一些形状，在工作时按照需要移动形状并调整它们的大小。当您移动形状并调整它们的大小时，可能还需要放大形状以便操作。

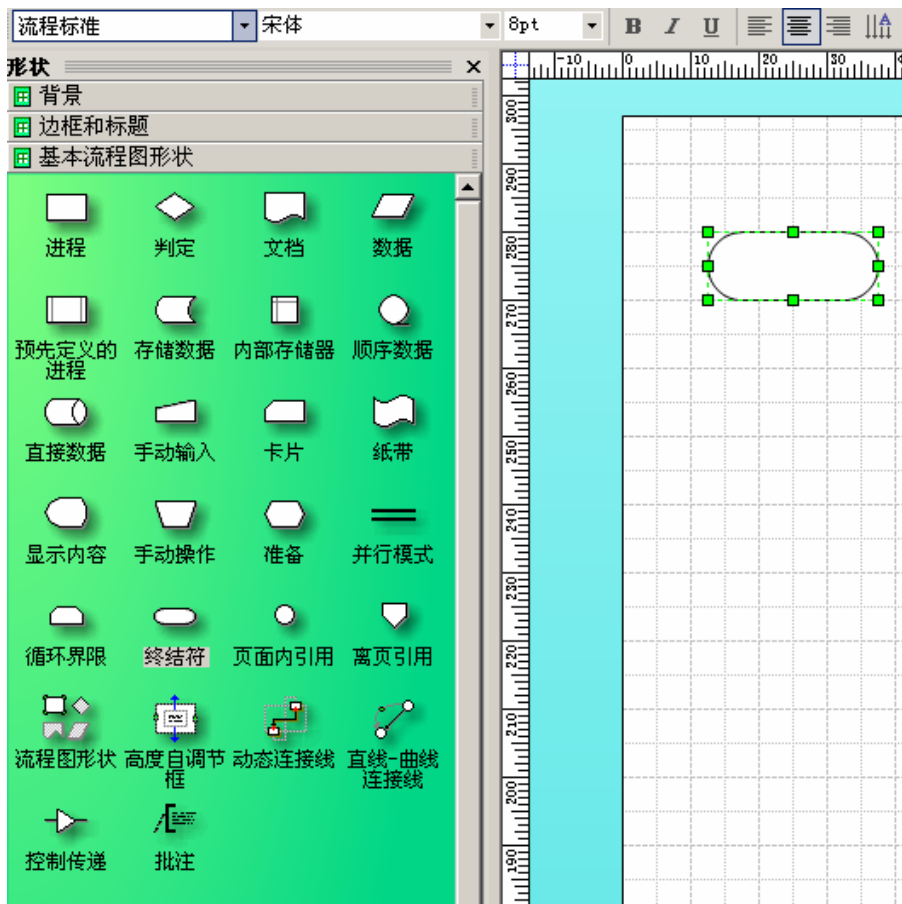


图 7

C) 添加文本

您可以向 Microsoft Visio 图表添加两种类型的文本—独立文本和与形状相关的文本。独立文本，例如标题或带项目符号的列表，是独立的文本块，与任何形状都不相关。与形状相关的文本或在形状中输入的文本会随形状移动。这两种类型的文本的添加方式都很简单，只需单击并输入。移动文本也只需单击并拖动。

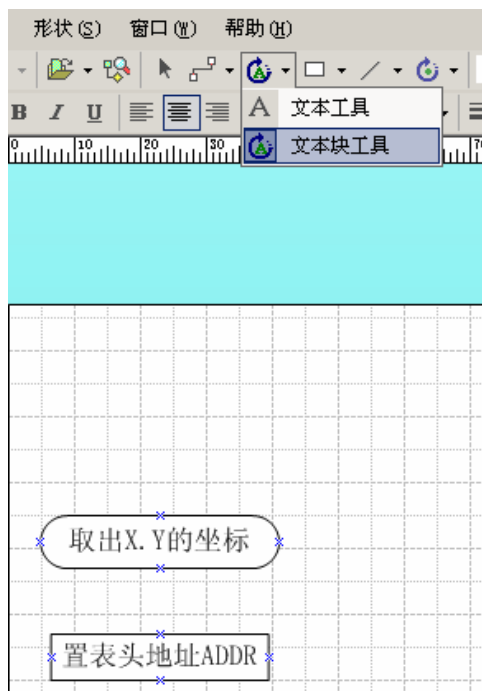


图 8

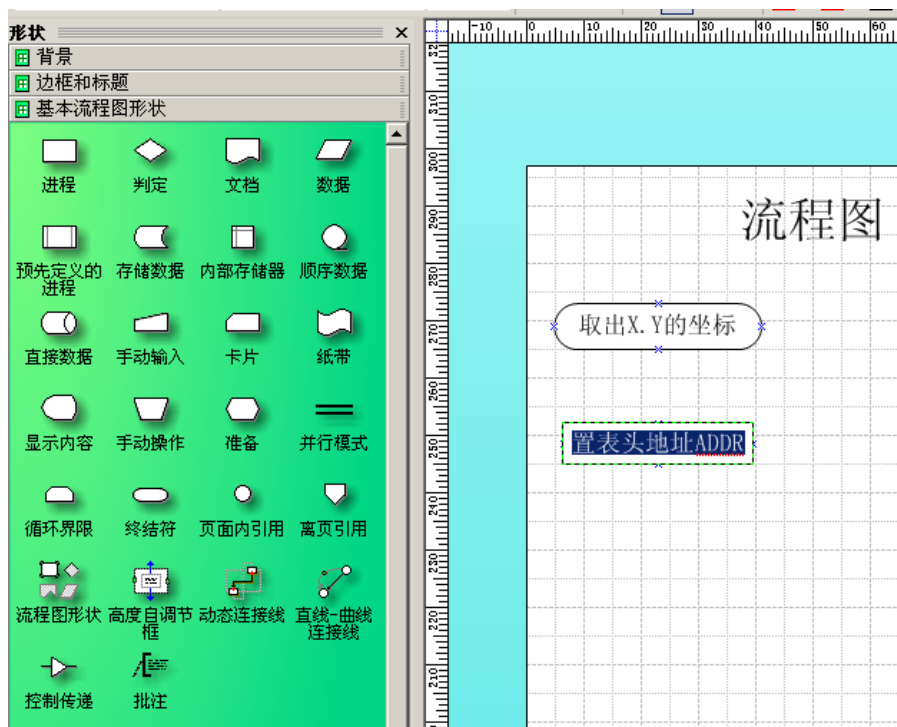


图 9

D) 连接形状

流程图、组织结构图、框图、网络图和 Web 图这些图表都有一个共同的形状，即称为连接线的一维形状。连接线连接了图表中的形状，当您移动一个具有连接的形状时，它仍保持连接状态。例如，当您移动流程图形状时，连接线将随之一起移动。

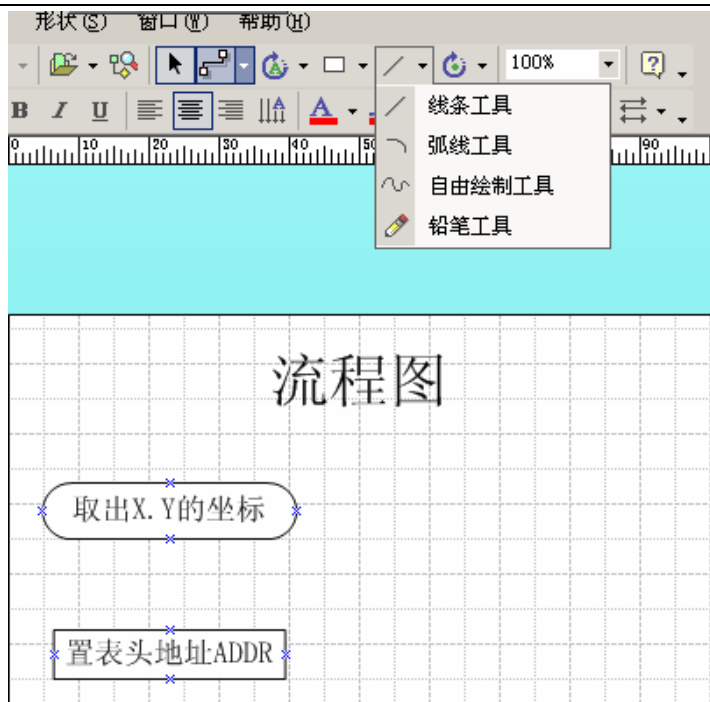


图 10

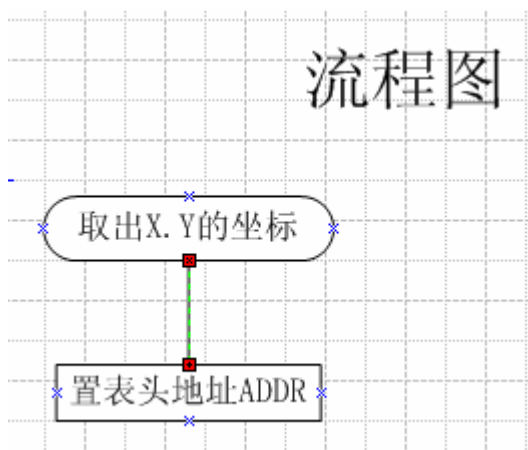


图 11

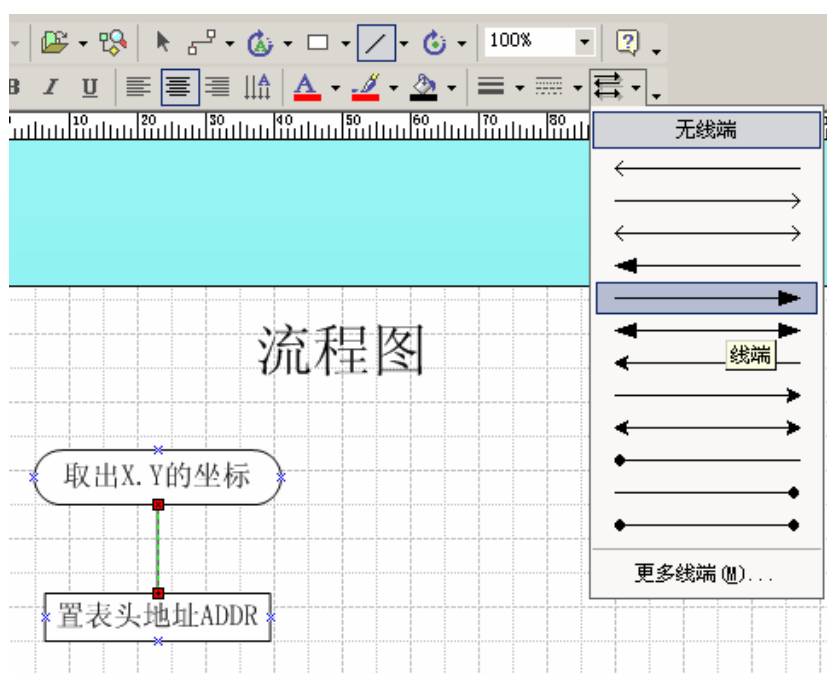


图 12

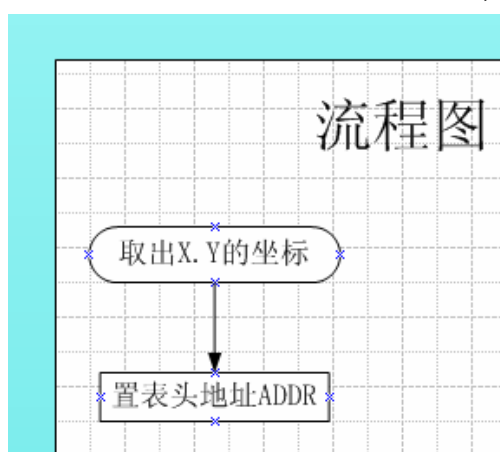


图 13

连接线还可以代表形状之间的关系。例如，流程图中的连接线代表工序流程。组织结构图中的连接线代表组织层次结构。数据库模型图中的连接线代表数据流。

其他情况下，连接线不代表关系；它只是用来连接两个形状。例如，您可以将网络设备连接到网络支架上。当您移动网络支架时，所有连接的设备将随之一起移动，无需逐个移动设备。

E) 设置形状格式

在 Microsoft Visio 中设置形状的格式十分快捷，只需用鼠标单击即可。先单击形状，然后单击工具更改其属性，例如颜色、图案、线条粗细等等。您可以使用颜色来突出图表中的特定形状，或者用线条图案或线条粗细来表示信息流的特定类型。



图 14

您还可以使用控制手柄更改形状的外观。例如，可以在椅子形状上拖动控制手柄来旋转椅子，在网络形状上拖动控制手柄来添加分支，或者在饼图上拖动控制手柄来添加扇区。

F) 完成和使用图表

现在您已经差不多掌握了创建任何 Visio 图表的基本技巧，现在需要了解如何处理已完成的图表。首先当然要保存它，然后可以打印、以其他文件格式保存，或将它包含到 Microsoft Word 文档中，等等,就像您以前使用 Office 软件一样，使用这些功能就可以了。

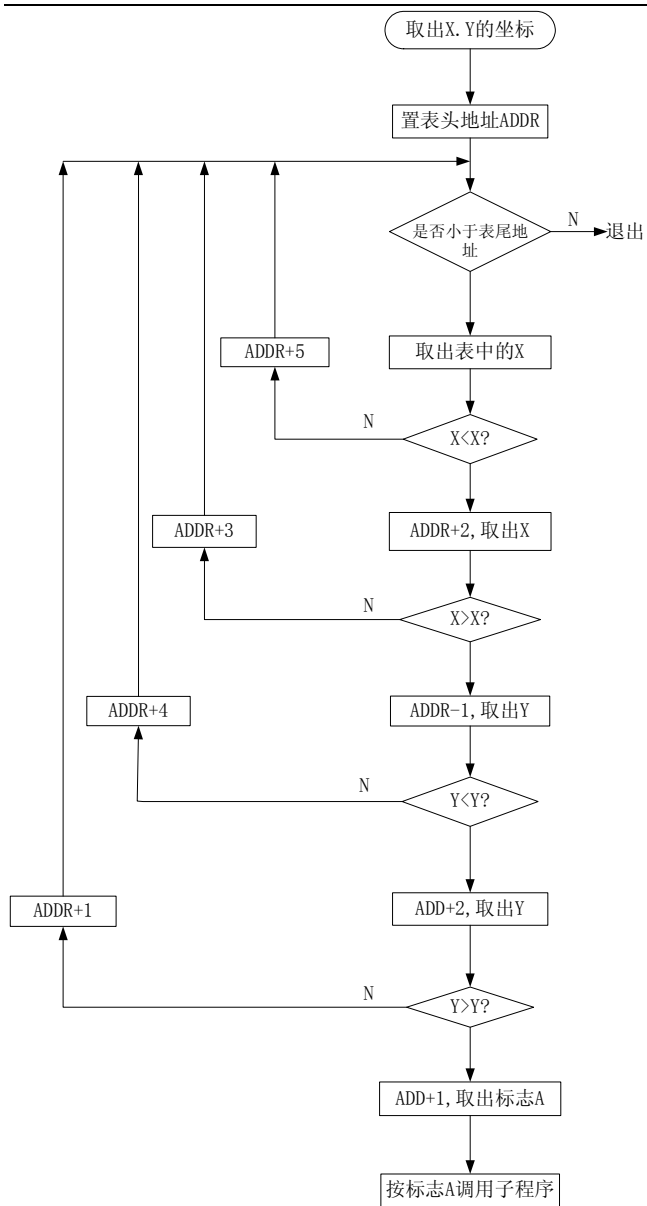


图 14

