

UML 在研究生成绩管理系统建模中的应用*

肖 岗,张广泉

(重庆师范大学 数学与计算机科学学院,重庆 400047)

摘 要 :以研究生成绩管理系统的开发为背景,探讨了 UML 在该系统建模中的应用问题。在对系统进行需求分析的基础上,采用 ROSE 对系统进行需求模型、静态模型、动态模型及实现模型的建模。

关键词 :UML ;面向对象方法 ;研究生成绩管理系统 ;建模

中图分类号 :TP311

文献标识码 :A

文章编号 :1672-6693(2007)01-0034-03

Application of UML in Modeling the Postgraduate Grade Management System

XIAO Gang ,ZHANG Guang-quan

(College of Mathematics and Computer Science ,Chongqing Normal University ,Chongqing 400047 ,China)

Abstract :This paper studies the application of UML in modeling the postgraduate grade management system. Started with the requirement analysis ,several kinds of models ,such as requirement model ,static state model ,dynamic state mode and realization model have been built by ROSE.

Key words :UML ;oriented-object method ;postgraduate grade management system ;modeling

统一建模语言(Unified Modeling Language ,UML)是由 3 位世界著名的面向对象技术专家 Gray Booch、Jim Rumbaugh、Ivar Jacobson 发起,在 Booch 表示法、OOSE 方法和 OMT 方法的基础上,广泛征求意见,集众家之长,反复修改后提出的通用的图形化标准建模语言^[1]。

UML 经过不断使用、发展和完善,已经成为一种定义良好、易于表达、功能强大,且普遍适用的建模语言,它为用户建模提供了完整的符号表示和不同层次的元模型,如用例图、包图、类图、状态图、对象图、活动图、顺序图、合作图、组件图、配置图等,其作用域不仅支持面向对象的分析与设计,还支持从需求分析开始的软件开发的全过程,用户可以根据自己所开发系统的特点和不同的开发阶段,灵活选用 UML 所提供的各种图和模型,以实现系统软件的柔性开发^[2]。

研究生成绩管理系统主要用于管理研究生的课程及其成绩。系统业务功能主要由以下几个部分组成:系统维护、数据导入导出、课程设置、成绩管理、数据统计、报表打印等。为保证系统结构设计良好,笔者使用 UML 来辅助进行系统分析与设计,并借鉴

了 Rational 公司提出的 RUP(Rational Unified Process,统一过程)框架模型,在对系统进行需求分析的基础上,采用 ROSE 逐步构建了系统的用户需求模型、静态模型、动态模型及实现模型等。

1 系统建模

1.1 需求模型

建立一个模型,需求分析是第一步,这里首先对系统需求进行分析,识别系统的用户和相关外部系统,以确定系统角色(Actor),它可以帮助界定软件系统的边界,引导和发掘用户需求;其次再依据系统功能来确立系统的用例(Use Case)模型。

本系统的需求分析说明大致如下:研究生处系统维护、查询成绩、学分统计、报表打印等;院系教学秘书成绩查询、学分统计、报表打印等;任课教师可以录入成绩、修改成绩、学分统计、查询成绩、报表打印等;学生查询本人成绩、学分统计等;此外还有一个相关外部系统招生管理系统,负责为本系统的数据导入功能提供相关数据接口。

由此可以确定系统的角色可以分为 5 个:研究生处、院系教学秘书、任课教师、学生及外部招生管

* 收稿日期 2006-01-21 修回日期 2006-04-05

资助项目:重庆市教委科学技术研究项目(No. 040803);中国科学院计算机科学重点实验室开放课题(No. SYSKF0303)

作者简介:肖岗(1976-)男,四川泸州人,硕士研究生,研究方向为软件体系结构与 UML 等研究。

理系统。在确定好角色的基础上,设计出了系统的用例图,如图 1 所示。

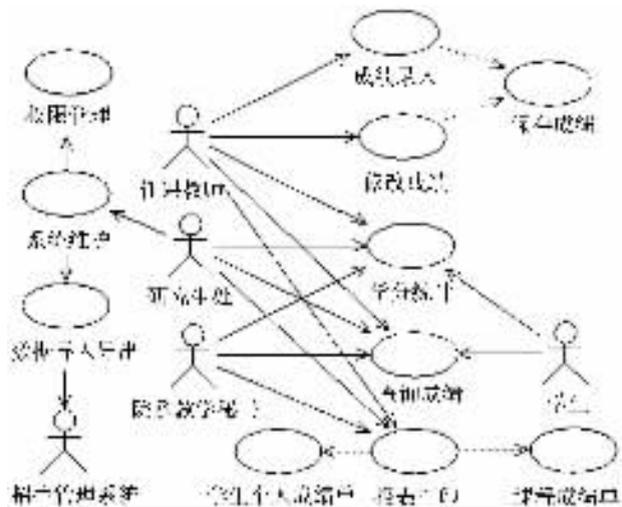


图 1 系统用例图

在设计用例图时,需要清楚地区分用例与用例图。用例简单地描述了用户要求系统所具备的动态,而用例图把用户、用例以及这两者包含在一个系统中,或者一个或多个子系统中。尽管执行者在用例图中是用类似人的图形来表示,但执行者并不一定是人,执行者也可以是一个外部系统,它可能需要从当前系统中获取信息,或者是外部信息的提供者,总之,它与当前系统有交互作用,如本例中的招生管理系统。

1.2 静态模型

建立系统静态模型包括类图、对象图、包图、组件图、配置图的建立。其中最重要的工作是确立系统的类图。类图用来描述系统中类的静态结构,它不仅定义类之间的联系,如关联、聚合、组成、依赖、泛化等,还包括类的内部结构,如类的属性和操作等。创建类图首先要做的是通过分析用例图以及问题域确定类及其关联,再确定其特性和操作,这是两个需要反复执行的过程。图 2 是整个系统的部分简化实体类图,它隐藏了相关类的属性和操作。

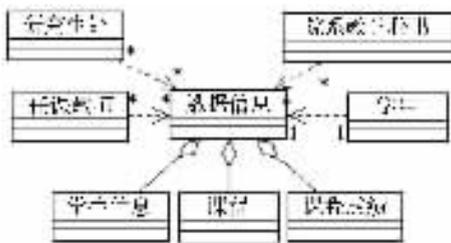


图 2 系统部分简化实体类图

在图 2 中,带有箭头的虚线表示依赖关系,研究生处、院系教学秘书、任课教师以及学生这 4 个类都依赖于数据信息类,它是本系统的主类(main

class),整个系统功能都围绕这一类展开。带空心菱形的箭头表示聚集(Aggregation)关系,它描述了两个类之间的整体-部分关系,其中一个类为整体,由一个或多个部分类组成,部分类可以没有整体类而存在,但是当它们聚集成一个整体,就用来组成该整体类,如数据信息就是整体类,由学生信息、课程、课程成绩 3 个部分类组成。

对于系统静态模型中的组件图和配置图,将在 1.4 中进行专门描述。

1.3 动态模型

在建立好系统静态模型基础上,接下来需要分析和设计系统的动态结构,以及建立相应的动态模型。在 UML 的表现上,主要是建立系统的交互图和行为图。其中交互图包括顺序图和协作图,行为图包括状态图和活动图。

顺序图和协作图在 ROSE 中是可以相互生成的,其中顺序图用来显示对象之间的动态合作关系,它用来建立以时间顺序安排的对象的交互,并且把用例行为分配给类,贯穿于 Unified Process 描述的软件开发生命周期的始终,其侧重点是时间顺序,协作图描述对象或角色之间是如何彼此通信的,其侧重点是事件。这意味着协作图更侧重于对象之间的交互而不太顾及时间因素,更侧重于说明哪些对象之间有交互。限于篇幅,这里只对任课教师登记成绩和修改成绩作协作图描述,涉及到的对象有任课教师、业务管理、成绩管理、学生、课程及成绩。任课教师登录系统,通过身份证验证后进入业务管理界面,打开成绩管理,就可以开始选择学生,然后选择该生的课程,如果该生对应课程没有成绩,则录入成绩并保存,否则就进行成绩修改操作并保存,如图 3 所示。

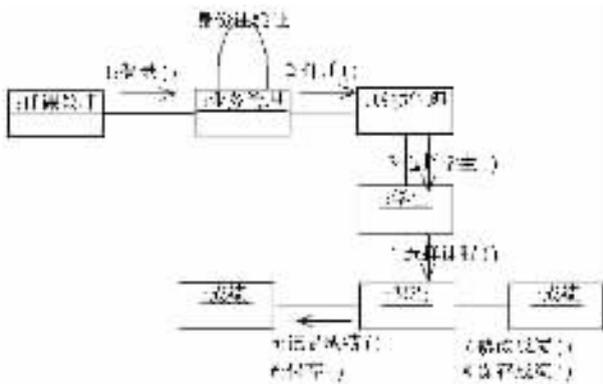


图 3 成绩录入及修改协作图

一般来讲,状态图描述了一个特定对象的所有可能状态以及由于各种事件的发生而引起状态之间的转移。活动图描述的是满足用例要求所要进行的活动及活动之间的约束关系,在实质上是一种流程

图,只不过表现的是从一个活动到另一个活动的控制流,活动图描述活动的序列,并且支持对带条件的行为和并发行为的表达^[3]。图 4 给出了任课教师管理成绩的活动图。

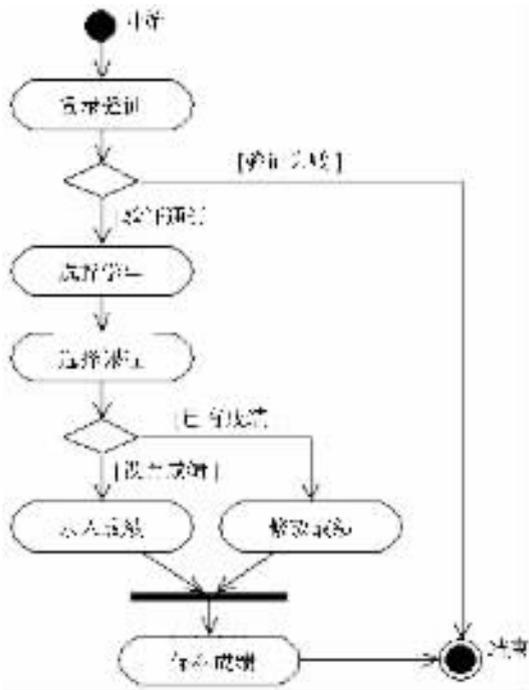


图 4 任课教师管理成绩活动图

1.4 实现模型

1.2 中提到系统的静态行为包括组件图和配置图,由于它们在本质上表示了系统实现的一些特性,描述了软件体系结构、硬件体系结构以及通信机制^[4],因此笔者认为将其归为系统的实现模型更为贴切。

组件通常为源代码、二进制代码和可执行部件等,是对建模元素物理实现的描述,组件图则是用依赖关系链接起来的组件的集合,可以描述与特定语言相关的编译时刻的依赖关系。由于组件可以带有接口,因此组件图也可以描述组件之间的接口关系和调用关系。配置图描述系统硬件的物理拓扑结构以及在这些结构上执行的组件,可以显示计算机节点的拓扑结构和通讯路径、节点上运行的软件组件、软件组件包含的逻辑单元(对象)等。利用 UML 的配置图,可以从更抽象的系统设计角度上,考察每一个软件模块、每一个软件的可执行体在物理节点之间的通信方式,或者用带箭头符号表示出不同节点上模块之间的通信方式。本系统结构基于 B/S 架构,配置图如图 5 所示。图中的立方体表示系统配置的节点,包括服务器和客户端 PC 机,结点之间的连线表示系统之间进行交互的通信协议名称,节点之间采用的是 TCP/IP 协议。

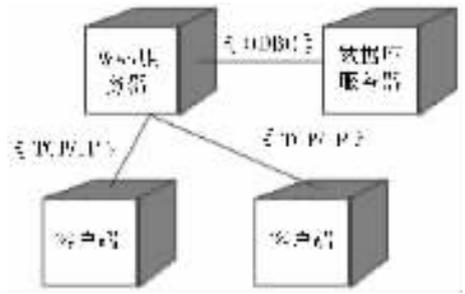


图 5 系统配置图

2 结语

上述以研究生成绩管理系统的开发为背景,运用 UML 面向对象建模机制对系统进行了分析设计。在建模过程,使用 UML 将系统的分析、设计和实现有机集成起来,便于对系统在更高抽象层次上进行维护,提高了系统的可扩展性。UML 提供的丰富视图从多个视角描述系统的不同侧面,可以有效运用于软件系统的建模、分析与设计。但是,作为一种通用的语言,UML 对软件体系结构的可构造性建模能力较弱,缺乏形式化语义,对体系结构的描述只能到达非形式化层次^[5]。笔者后续的研究就是要探讨运用 UML 和 ADL 相结合来描述软件体系结构的集成建模问题。ADL 形式化语义的精确性正好可以弥补 UML 非形式化的一些不足,二者的有机结合,不仅可以更好地描述软件体系结构模型,而且还支持下一步的求精和验证工作^[5-7]。

参考文献:

- [1] Rational Software Corporation. UML Summary. Http://www.rational.com/uml, 2006-04-22/2006-05-05.
- [2] 张莉,葛科,王云,等. UML 软件开发过程和支持环境研究[J]. 北京航空航天大学学报, 1998, 24(4): 407-408.
- [3] ERIC J B. Software Engineering An Object-Oriented Perspective[M]. 北京:电子工业出版社, 2003.
- [4] 刘超,张莉. 可视化面向对象建模技术——标准建模语言 UML 教程[M]. 北京:北京航空航天大学出版社, 2001.
- [5] 张广泉,张玲红. UML 与 ADL 在软件体系结构建模中的应用研究[J]. 重庆师范大学学报(自然科学版) 2004, 21(4): 1-2.
- [6] 张广泉. 软件体系结构与 XYZ 系统[R]. 北京:中科院软件研究所博士后研究报告, 2002.
- [7] 戎玫,张广泉. 软件体系结构求精方法研究[J]. 计算机科学, 2003, 30(4): 108-110.

(责任编辑 游中胜)