

基于软件体系结构和 UML 的图书管理系统设计与实现^{*}

戎 玫¹ 张广泉^{2,3} 刘 艳²

(暨南大学深圳旅游学院 深圳518053)¹ (苏州大学计算机科学与技术学院 苏州215006)²

(重庆师范大学计算机科学学院 重庆400047)³

摘 要 软件体系结构是目前软件工程领域一个新兴的研究方向,对于复杂的软件系统而言,其体系结构的设计已经成为系统能否成功的一个关键因素。本文以一个图书管理系统为例,采用统一建模语言 UML 对该系统体系结构建模,并通过 VB 来实现该系统。

关键词 图书管理系统,软件体系结构,建模,UML,VB

Design and Implement of Library Management System Based on Software Architecture and UML

RONG Mei¹ ZHANG Guang-Quan^{2,3} LIU Yan²

(ShenZhen Tourism College, Jinan University, Shenzhen 518053)¹

(School of Computer Science and Technology, Soochow University, Suzhou 215006)²

(School of Computer Science, Chongqing Normal University, Chongqing 400047)³

Abstract Software Architecture is an immature field in the discipline of software engineering. Good architecture design has always been a major factor in determining the success of complex systems. This article takes a library management system as an example, and model this system's SA by using UML. Then we use VB to realize this system.

Keywords Library management system, Software architecture, Modeling, UML, VB

1 引言

软件体系结构(Software Architecture)是20世纪90年代以来国际软件工程领域出现的一个新兴的、重要的研究内容。它通过对软件总体结构的描述来提供系统的高层抽象,对于大型复杂软件系统而言,其体系结构的设计已经成为系统能否成功的关键因素之一。

研究软件体系结构的首要问题是如何表示和描述体系结构,亦即所谓的体系结构建模^[1]。目前还没有统一成熟的软件体系结构建模工具,常用的软件体系结构描述方法主要有两大类:专用的体系结构描述语言 ADL 与通用的建模语言(如 UML 等)。ADL 大多基于一些形式化工具(如 CSP、Z、时序逻辑等),常见的 ADL 有 Wright、Rapid、Unicon、Aesop、Darwin、ACME、XYZ/ADL^[2]等,它们吸收了传统程序设计中严格精确的语义和语法的特点,针对软件体系结构的整体性和抽象性特点,定义和确定了适合于软件体系结构表达与描述的有关抽象元素,能有效支持所描述系统的分析、求精和验证,但是,ADL 不足之处是难以被开发人员所理解,不便于交流和使用的,很难融入到当前软件开发的实践中。

统一建模语言 UML 是一种语义丰富、通用、可视化的建模语言和事实上的国际工业标准,易于理解和交流。UML 提供的丰富的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的建模、分析与设计^[3,4]。基于此,本文以一个具有典型异构体系结构的图书管理系统为例,该图书管理系统运行在图书馆局域网上,假定在图书馆中有功能强大的工作站机器和多个台式 PC 机,台式机作为终端放置于多个借书还书处。系统的应用服务器和数据服务器设置在工作站上,图书管理员可以在图书馆的台式 PC 机上使用该系统。我们采用

可视化建模工具 Rational Rose 2003 支持 UML 建立其体系结构模型,探索 UML 在软件体系结构建模方面的特性;在为图书管理系统体系结构建立模型后,将采用 Visual Basic 6.0 来实现该系统,VB 有足够多的“对象特性”,可以充分利用 UML 模型中的设计元素;采用的数据库平台是 SQL Server 2000,其功能强大,操作简便,可跨平台使用,非常适合图书管理系统。

2 基于 UML 的图书管理系统体系结构建模

软件体系结构的模型根据不同的侧重点分为五类:功能模型、结构模型、框架模型、动态模型、过程模型。UML 提供了五类视图(用例图、静态图、行为图、交互图、实现图),而这些视图正适合从不同的侧重点为系统体系结构建立上述各种模型。这些模型将有机地统一在一起,对软件体系结构进行全面的刻画。

本图书管理系统体系结构建模除了上述五种模型外,还涉及到需求模型以及体系结构风格,限于篇幅,这里只简要介绍其中的一部分。

2.1 需求模型

UML 的用例图(Use Case)较详细和确切地描述了用户的功能需求,使系统责任更为明确,是其他 UML 视图的核心和基础。其他视图的构造和发展依赖于用例图中所描述的内容,系统的最终目标就是提供用例视图中描述的功能。图 1 是系统的顶层需求模型,该图由各使用者和所驱动的用例以及用例之间的相互关联组成。可以确定系统有三个活动者:读者(Reader)、图书管理员(Librarian),总图书管理员(Librarianleader)。读者通过图书管理员与系统交互,图书管理员通过系统进行读者管理、书籍管理、借阅管理和系统管理,Li-

^{*} 本项研究得到国家自然科学基金(60073020);国家科学院计算机科学国家重点实验室开放课题(SYSKF 0303);重庆市教委科学技术研究项目(040803);江协省计算机信息处理技术重点实验室开放课题基金资助。戎 玫 博士、副教授,主要研究方向为软件工程、数据通信等。

brarianleader 是 Librarian 的泛化, Librarianleader 能够进行整个系统管理, 而 Librarian 只能进行部分系统管理。

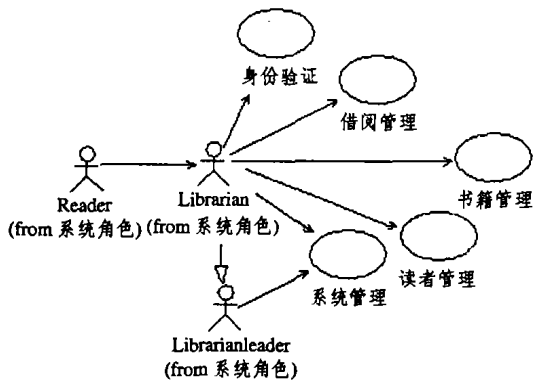


图1 图书管理系统顶层需求模型——UML 用例图

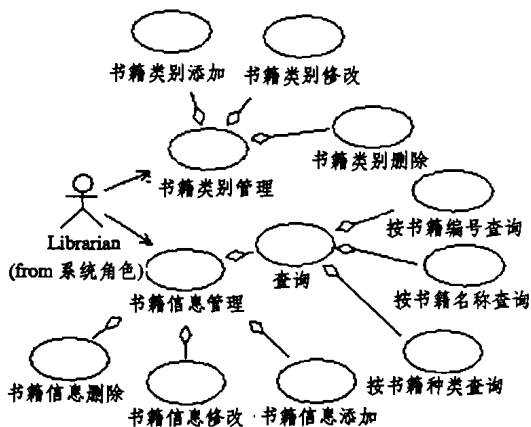


图2 书籍管理子系统需求模型——UML 用例图

在顶层需求模型建立后, 需要进一步构建更加精确的需求模型。以用例“书籍管理”为例, 图2是书籍管理子系统需求模型。用例“书籍管理”可以分解为: “书籍类别管理”和“书籍信息管理”两个用例; 空心的棱形箭线表示聚合, 代表一种整体与部分的关系。

一般来说, 用例定义后我们需要为一些重要用例建立一些简单的行为模型, 从而使该用例更为清晰, 也为我们在建立结构模型时更容易把握哪些构件是应该抽象出来的。

2.2 体系结构风格

描述软件体系结构风格是基于体系结构建模的重要内容。该图书管理系统体系结构风格与其子系统构件体系结构风格不同, 属于异构的体系结构。该系统运行于图书馆局域网, 整体上是一种典型的三层 Client-Server 体系结构风格, 各种服务分离在不同层次, 易于开发和维护。为了满足表示层多变的需求, 各子系统采用了模型-视图-控制器(MVC)风格。该风格把一个应用的输入、处理、输出流程按照 Model、View、Controller 的方式进行分离。模型封装内核数据与状态, 视图从模型获取数据, 用户输入的数据通过控制器与系统交互。在本系统中, 表单就是视图, 实体就是模型, 控制器则包括用户接口控制器和应用控制器。

2.3 功能模型

该模型认为体系结构是由一组功能组件按层次组成, 下层向上层提供服务。我们为本图书管理系统构建了四个组件: 标准的 uisvc. exe, brsvc. dll, 以及 dtsvc. dll 和 dasvc. dll。但是在发布时, 将 brsvc. dll 编译为 ActiveX EXE 组件(进程外), dtsvc. dll 也将编译为 ActiveX EXE 组件(进程外)。图3为 UML 组件图, 各组件之间相互通信, 下层为上层提供服务,

虚箭线表示依赖关系。dasvc 组件提供数据访问服务, 执行 API 的请求, 直接与数据库进行交互; dtsvc 组件提供数据转化服务, 对信息服务的逻辑请求转换为数据兼容的语言(SQL 语句), 该组件中的数据转化类将实现 Icrud 接口; brsvc 组件提供传统的功能服务; uisvc 组件提供传统的表示服务。组件间的通信技术采用 Microsoft 的组件对象模型(COM)和 DCOM(对于建造者来说, DCOM 就是带有“长线”的 COM)。以字符串数组作为层间通信结构, 灵活稳定。uisvc 组件将运行于图书馆客户机, brsvc 组件运行于应用服务器, dtsvc、dasvc 以及数据库放置在数据库服务器上, 各节点通过局域网连接。UML 配置图很好地定义了系统中软硬件的物理体系结构, 如图4所示。

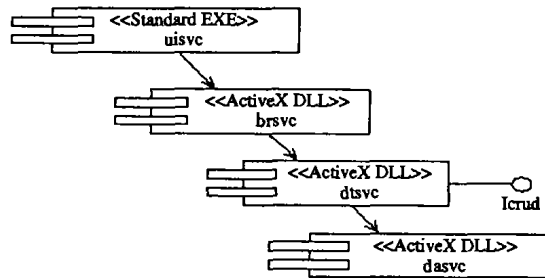


图3 图书管理系统功能模型——UML 组件图

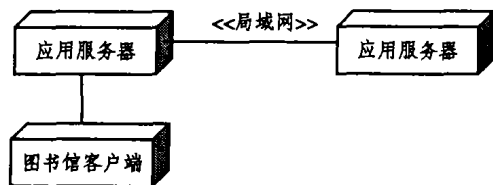


图4 图书管理系统功能模型——UML 配置图

2.4 结构模型

结构模型侧重于描述系统结构的细节, 类图可以用来构建系统的结构模型。类图是 UML 的核心技术, 表示系统中的类和类与类之间的关系, 是对系统静态结构的描述。一个典型系统中通常有若干个类图。图5是借阅管理子系统的结构模型。“Borrowlend”是“Book”与“Reader”的关联类, 它定义了一组属于该关联的特性, 比如 Borrowdate(借阅时间)等, 这样的特性不适合放于任何一个被关联的类中。“Borrowlend”类与“BorrowlendDT”类互相通信, 后者将前者对数据库的请求转化为 SQL 语句后, 将 SQL 语句作为参数传递给“Access”类, 由“Access”类直接访问数据库。

2.5 动态模型

动态模型描述系统大颗粒的行为性质, 可用 UML 的交互图以及状态图和活动图来描述图书管理系统的动态模型。交互图描述构件间消息传递的连接关系及其空间分布, 揭示构件之间的交互过程, 用于描述一个用例, 它有两种类型: 顺序图和协作图。状态图表现了一个对象(或模型元素)的生存史, 显示触发状态转移的事件和因状态改变而导致的动作, 对于比较重要的对象需建立其状态图。活动图主要表示系统的业务 workflows 和并发处理过程, 对于一些比较重要的流程需要建立其活动图。图6是借书用例的协作图。“UCBorrowlend-Controller”向“Borrowlend”发送 setState, 将从用户接口获得的借阅信息赋给“Borrowlend”的属性。然后“UCBorrowlend-Controller”向“Borrowlend”发送 insert 消息时, 需将“Borrowlend”的属性重新组装成数组传递到下一层, 这个过程由

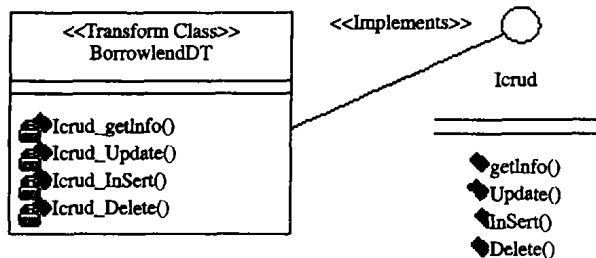


图8 “BorrowlendDT”类

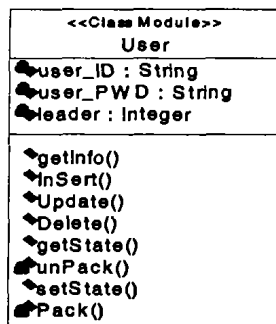


图9 “User”类

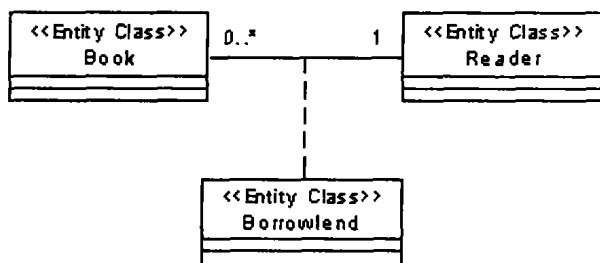


图10 关联类“Borrowlend”

因为 Rose 在生成关联类的 VB 代码方面存在缺点,所以从这个模型得不到我们想要的 VB 代码。我们需要的代码,或者说应该生成的代码如下:

```

'Borrowlend Class
Public myReader as Reader
private mcolbook as Collection
  
```

所以,不管是在代码生成前修改模型还是代码生成后进行逆向工程,我们都应得到如图 11 的模型。Mcolbook 与

myReader 作为角色名,“+”表示 Public,“-”表示 Private。

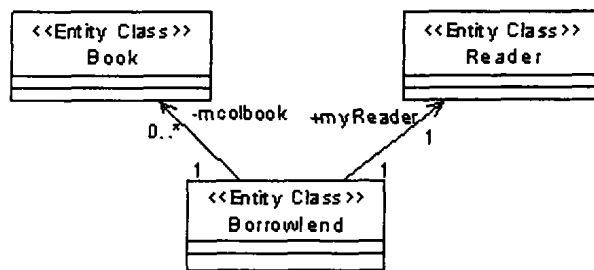


图11 修改后的模型

由实现过程可知代码的主要资源是类和组件图。在整个实现的过程中,层次关系非常清晰,体现了由软件体系结构建立模型的优势。当代码实现后,我们应该将应用程序代码逆向工程到可视化模块中,实际上,正向工程、编码、逆向工程——这是个重复的过程。

结束语 本文以一个图书管理系统开发为背景,探讨了基于 UML 和软件体系结构的建模与实现问题。由建模过程可以看出 UML 在体系结构建模方面存在很好的优势,它提供了一个统一的交流平台,而实现过程则充分体现了 VB 的面向对象特性。软件体系结构与面向对象开发方法相结合,不但从整体上正确描述了该图书管理系统的体系结构并且可以充分利用 UML 面向对象的优点,最终使得项目成功完成。

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的层次。因此,下一步工作将探讨 UML 和体系结构描述语言 ADL 相结合来描述软件体系结构问题,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,不仅可更好地描述体系结构模型,还支持下一步的求精和验证工作^[6]。

参考文献

- 周莹新,艾波. 软件体系结构建模研究. 软件学报,1998, 9(11): 867~872
- 唐稚松,等. 时序逻辑程序设计与软件工程. 北京:科学出版社,2002
- 刘超,张莉. 可视化面向对象建模技术——标准建模语言 UML 教程. 北京:北京航空航天大学出版社,2000
- 张龙祥. UML 与系统分析设计. 北京:人民邮电出版社,2002
- Paul R. Developing Applications with Visual Basic and UML. Addison Wesley,1999
- 戎玫,张广泉. 软件体系结构求精方法研究. 计算机科学,2003, 30(4):108~110

(上接第216页)

们开发的通用 MIS 引擎可以用来定制不同行业的管理信息系统,在实际的应用中,取得了很好的效果。对我们的通用 MIS 引擎来说,配置项数据就是对具体 MIS 应用进行分析设计的结果,因此,一个 MIS 应用可按以下方式得到:

MIS 应用系统=通用 MIS 引擎+MIS 应用的分析设计结果

在这个过程中,无需任何编码。可以说,这是未来开发 MIS 应用的趋势:把握 MIS 应用的本质,构建大粒度可复用 MIS 组件,让 MIS 开发人员从编码工作中解脱出来,将更多精力集中在需求的提取与分析上。

参考文献

- Persse J R. Implementing the Capability Maturity Model[M]. Wi-

ley,2001

- Singhai A, Sane A, Campbell R H. Quarterware for Middleware [C]. In: Proc. of the 18th IEEE Intl. Conf. on Distributed Computing System, Amsterdam, May 1998. 192~201
- Burton B A, Aragon R W, Baily S A, et al. The reusable software library[J]. IEEE SOFTWARE, July 1987. 25~33
- 张海藩. 著. 软件工程导论[M]. 清华大学出版社,2003
- 张畅,徐冬溶,潘云鹤. 于多源类比的 MIS 表格生成[J]. 计算机研究与发展,1998
- 杨肖鸳,等. 面向敏捷制造的虚拟企业管理信息系统框架研究[R]. 2001. 40~49
- 李相枢. 基于软件体系结构的 MIS 系统数据采集构件[J]. 计算机科学,2002,4
- Kirland M. 基于组件的应用程序设计[M]. 北京:博彦科技发展有限公司译. 北京大学出版社,1999