

# 基于 UML 的图书管理系统体系结构模型及实现\*

张广泉<sup>1 2 3</sup>, 刘 艳<sup>2</sup>

(1. 重庆师范大学 数学与计算机科学学院, 重庆 400047; 2. 江苏省 计算机信息处理技术重点实验室, 江苏 苏州 215006;  
3. 中国科学院 计算机科学国家重点实验室, 北京 100080)

**摘 要** 探讨软件体系结构与面向对象方法相结合的问题。以一个图书管理系统为例, 采用统一建模语言 UML 对该系统的软件体系结构建模, 并使用 VB 语言实现了该系统。

**关键词** 图书管理系统 软件体系结构 建模 UML VB

中图分类号: TP311

文献标识码: A

文章编号: 1672-669X(2005)02-0001-05

## Modeling Software Architecture and Their Implement Based on UML of Library Management System

ZHANG Guang-quan<sup>1 2 3</sup>, LIU Yan<sup>2</sup>

(1. College of Mathematics and Computer Science, Chongqing Normal University, Chongqing 400047;  
2. Jiangsu Key Lab. of Information Processing Technology, Soochow University, Suzhou Jiangsu 215006;  
3. State Key Lab. of Computer Science, The Chinese Academy of Sciences, Beijing 100080)

**Abstract** This paper presents an effective approach of combining software architecture(SA) and object-oriented method. We take a library management system as an example, and model this system's SA by using UML. Then we use VB to realize this system.

**Key words** library management system software architecture modeling UML VB

软件体系结构作为一种高层的抽象的系统设计, 已经成了决定一个软件系统成功与否的关键因素<sup>[1]</sup>。尤其对于大型系统, 软件体系结构的设计就显得格外重要, 直接影响着系统的执行效率。因此, 作为软件工程研究领域的一个部分, 软件体系结构已经受到了越来越多的软件系统设计和研究人员的重视。近年来, 基于软件体系结构的设计分析和开发方法已逐渐成为解决软件复杂性和工程设计困难的重要出路, 通过对软件系统的上层结构准确、深入的认识和设计来指导整个软件系统设计开发方法, 乃至整个生命周期活动。

在软件体系结构设计中一个非常重要的方面就是对软件体系结构的建模<sup>[2]</sup>。如何用适当的方法把

软件的体系结构模型详细精确地描述出来, 不仅影响着开发人员之间的交流和对体系结构的理解, 更影响到后续的工作, 因而软件体系结构的建模问题也成了软件体系结构研究领域一个热点和核心问题<sup>[3-5]</sup>。

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 and 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<sup>[6,7]</sup>。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

\* 收稿日期 2005-01-17

资助项目: 国家自然科学基金(60073020)、重庆市教委科学技术研究项目(040803); 江苏省计算机信息处理技术重点实验室开放课题、中国科学院计算机科学国家重点实验室开放课题

作者简介: 张广泉(1965-)男, 江苏连云港人, 教授, 博士后, 硕士生导师, 主要从事软件工程、形式化方法等研究。

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。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。

### 1 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 5 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 5 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 5 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。

#### 1.1 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 3 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,Librarianleader 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。

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,空心的菱形箭线表示聚合,代表一种整体与部分的关系。

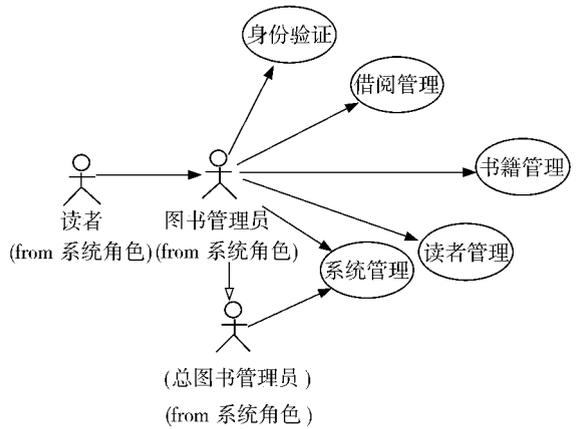


图 1 图书管理系统顶层需求模型—UML 用例图

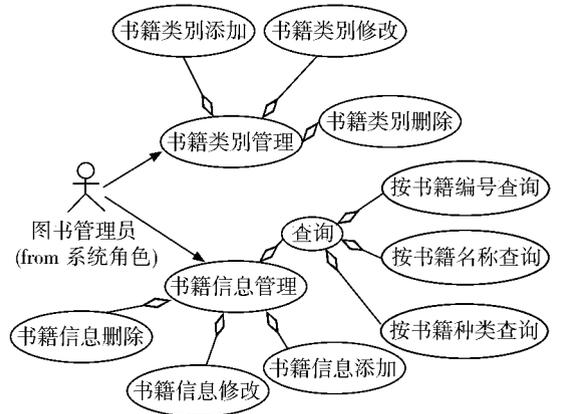


图 2 书籍管理子系统需求模型—UML 用例图

一般来说,用例定义后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。

#### 1.2 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 3 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。

### 1.3 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。为本图书管理系统构建了 4 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

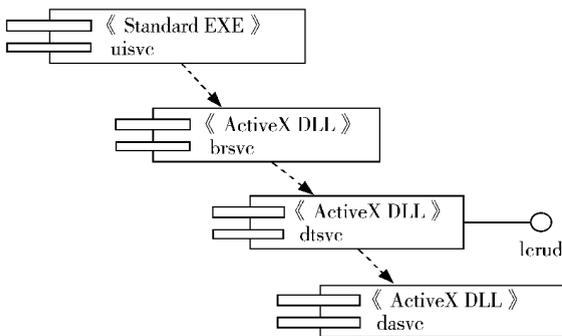


图 3 图书管理系统功能模型—UML 组件图

### 1.4 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”类互通,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。

### 1.5 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

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 lcrud 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。

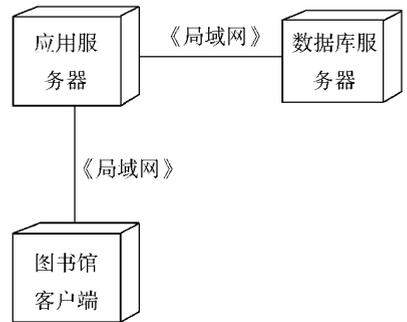


图 4 图书管理系统功能模型—UML 配置图

的动作,对于比较重要的对象需建立其状态图。活动图的主要表示系统的业务 workflow 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。“UCBorrowlendController”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“UCBorrowlendController”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“Borrowlend”向自己发送 Pack 消息来完成。“Borrowlend”构件在对“BorrowlendDT”构件发送 insert(插入)消息之前,首先要对“Book”发送 Update 消息,即更新该书籍为借出,如果这个消息发送失败,也就是这本书不可借出,那么 insert 也将执行失败。最后“frmBorrowlend”表单将根据执行结果显示提示,比如“该书已借出”;“该书不存在”等。

## 2 图书管理系统的实现

Visual Basic 是一种有力的企业开发工具,它虽然不是一种真正意义上的面向对象语言,但是它是基于对象的,它有许多强大的面向对象的能力,它支

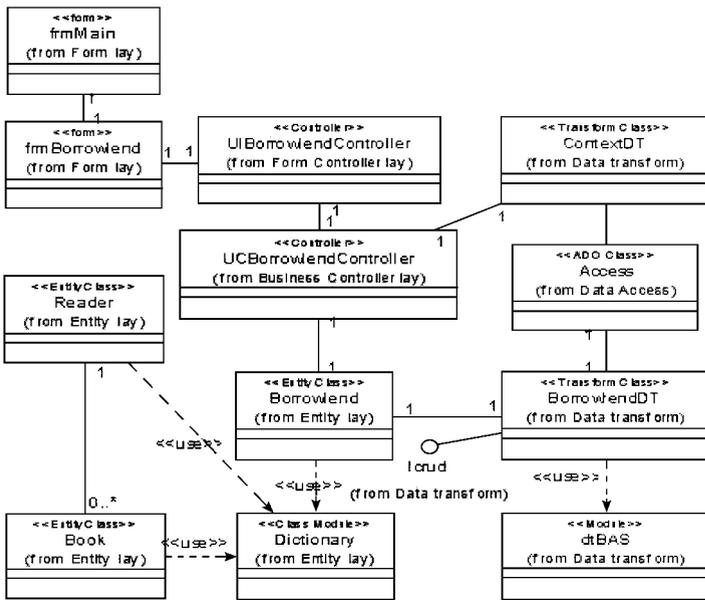


图 5 借阅管理子系统结构模型—UML 类图

```

classDiagram
    class Access {
        <<ADO Class>>
        Access()
        ~Access()
        DAConnect(String, Long, String)
        DADisconnect()
        DAQuery(String, Long, String)
        DABeginTrans()
        DACCommitTrans()
        DARollBack()
        DAUpdate(Long, String)
        DAInsert(Long, String)
        DADelete(Long, String)
        DASetContext(String, Boolean, Long, String)
        DASetComplete(Boolean, Long, String)
        DASetClose(Long, String)
    }
  
```

图 7 “Access”类

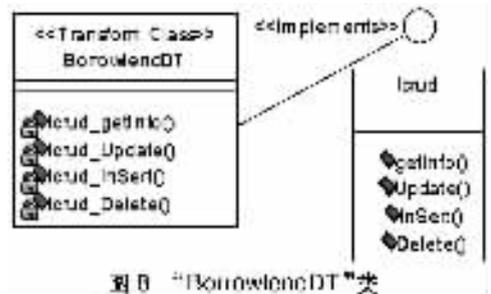


图 8 “BorrowendDT”类

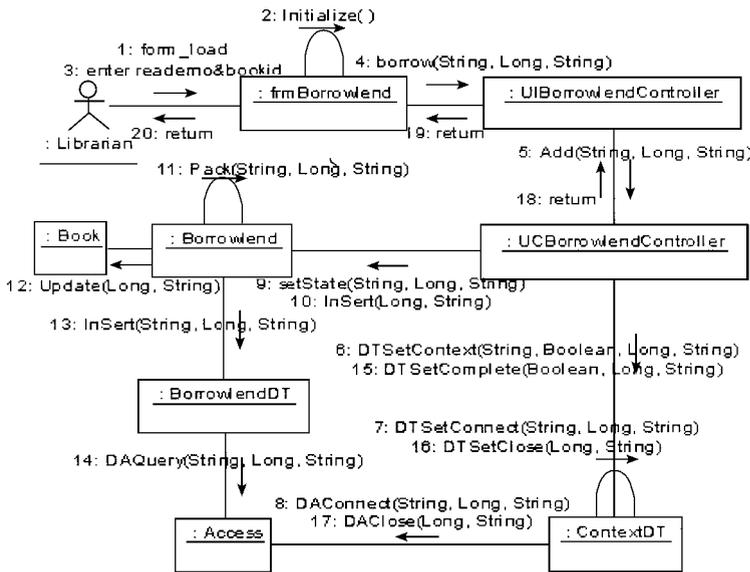


图 6 借书动态模型—UML 协作图

持类、复合类型、封装、继承、消息传递、多态。虽然这里的继承只能是接口继承而不是实现继承,但是 VB 仍然具有足够的面向对象特性,可以充分利用 UML 模型中的设计元素。UML 视图与 VB 存在着映射关系<sup>[8]</sup>,它们的结合是十分合适的。例如 组件图中的各组件对应 VB 的各类工程( EXE ,DLL );类对应着 VB 类模块或者模块,类中的操作对应其中的子程序( SUB ,FUNCTION )。

在使用 Rational Rose 这一可视化建模工具时,通过前向工程与逆向工程的结合,可使 UML 的价值大大提高。dtsvc.dll 工程中只有一个类模块——“ Access”(如图 7),该类从数据转化类中获取 SQL

语句,并直接与数据库交互。对数据库的查询由 DARetrieve 操作执行,对数据库的插入、删除、更新则由 DAQuery 操作执行。涉及到事务管理的操作是 DABeginTrans(开始)、DACCommitTrans(提交)、DARollBack(滚回)。dasvc.exe 工程包括与实体一一对应的数据转化类如 BorrowendDT 类(如图 8)、BookDT 类等,每个数据转化类都实现接口 Icrud,其功能是将与对应实体有关的请求转化成 SQL 语句。需要通过该层的服务来建立一个连接,并为多步的面向更新的 SQL 语句建立一个工作单元,所以添加了一个名为 ContextDT 类。该类直接与 theAccess 对象交互。

brsvc.exe 工程中包括实体类(图 9 为实体“ User”类)与控制器类。实体类与数据转化类交互,进行主要的功能处理。控制器类主要从客户端获取信息,向 ContextDT 类以及实体类发送消息。uisvc 标准工程中包括表单和用户接口控制器类。用户接口控制器从表单获取信息,必要时可以进行一些筛选和处理,然后将信息传递到应用服务器。它主要和 brsvc 中的控制器进行通信。在具体编码实现或者逆向工程时都有可能对模型进行修改,使模型适应应用程序的语义。例如,对于关联类“ Borrowend”,其初始模型如图 10 所示。

