

基于 UML 的 Wiki 系统设计与实现^{*})

李学俊 李龙澍 程慧霞 徐 怡

(安徽大学计算智能与信号处理教育部重点实验室 安徽大学计算机学院 合肥 230039)

摘要 目前 Wiki 系统处于初始发展阶段,各种应用系统具有不同的功能和特点,并没有统一的标准。为了建立标准的 Wiki 系统,本文将面向对象建模语言 UML 引入 Wiki 系统的设计和实现过程中,采用 Rational Rose 2003 建立其功能需求模型、数据模型和 Web 模型。建模语言的标准化和 Wiki 系统模型,提高了 Wiki 系统的软件重用和开发效率,并有利于 Wiki 系统的标准化和多样性。

关键词 维基, Wiki 系统, 统一建模语言, 建模

The Design and Implement of Wiki system Based on UML

LI Xue-Jun LI Long-Shu CHENG Hui-Xia XU Yi

(Key Lab IC&SP, School of Computer Science and Engineering, Anhui University, Hefei 230039)

Abstract Currently Wiki system develops initially, and its application systems have different functions and characteristics, also it has not unified standard. To build standard Wiki system, object-oriented UML is used to the process of the design and implement of Wiki system, and the tool of Rational Rose 2003 is applied to construct function requirement model, data model, and Web model. The standardization of modeling language and Wiki system model can improve software reuse and development efficiency of Wiki system, and benefit the standardization and diversity and Wiki system.

Keywords Wiki, Wiki system, UML, Modeling

1 引言

Wiki 一词来源于夏威夷语的“wee kee wee kee”,原本是“快点快点”的意思,被译为“维基”或“维客”。Wiki 指一种超文本系统实现技术,用以实现超文本系统。这种超文本系统支持面向社群的协作式写作,同时也包括一组支持这种写作的辅助工具。我们可以在 Internet 平台对 Wiki 页面进行浏览、创建、更改,而且创建、更改、发布的代价远比 HTML 文本小;同时 Wiki 系统还支持面向社群的协作式写作,为协作式写作提供必要帮助;最后, Wiki 的协作者自然构成了一个社群, Wiki 系统为这个社群提供简单的交流工具。与其它超文本系统,如博客 Blog、BBS 相比, Wiki 有使用方便及开放的特点,所以 Wiki 系统可以帮助我们在一个社群内共享某领域的知识,构建知识社区共享平台。

随着 Wiki 社区的应用发展,已经出现了很多支持该社区的 Wiki 系统。按照系统运行环境,主要分为四类,基于 PHP 环境的 MediaWiki、CooCooWakka 等;基于 JSP 环境的 JspWiki 等;基于 CGI 环境的 TWiki、UseModWiki;基于 ASP 环境的 JsWiki、OperatorWiki、OpenWiki 等。然而目前 Wiki 系统处于初始发展阶段,各种系统具有不同的功能和特点,并没有统一的标准^[6,7]。

为了建立标准的 Wiki 系统,有必要引入建模语言对 Wiki 系统进行功能需求建模、数据建模和 Web 建模。而面向对象建模语言 UML (Unified Modeling Language) 正是被对象

管理组织 OMG (Object Management Group) 确定为标准的面向对象建模语言,它是对软件开发过程中各种制品进行可视化、详述、构造和文档化的、面向对象的、独立于过程的建模语言。因此,我们抽象出各种 Wiki 系统的共同特点和本质特征,运用 UML 描述 Wiki 系统的设计和实现过程,采用 Rational Rose 2003 建立 Wiki 系统的功能需求模型、数据模型和 Web 模型。建立基于 UML 的 Wiki 系统模型,可以提高 Wiki 系统的软件重用和开发效率,有利于 Wiki 系统的标准化和多样性,促进 Wiki 系统的研究和应用。

2 UML 概述^[1,4~5,8]

面向对象的软件开发技术包括对象、类、消息、封装、继承、多态等概念,并有相应一整套从需求分析到部署实现的软件工程方法。由于面向对象中“对象”概念的引入,将数据和方法封装在一起,提高了模块的聚合度、降低了模块的耦合度,在更大程度上支持了软件重用,从而十分有效地降低了软件的复杂度,提高了软件的生产率。面向对象建模技术 OMT (Object Modeling Technique) 利用四种模型,即对象模型、功能模型、动态模型和用例模型,在建模语言的支持下,完成对整个系统的建模。

UML 是在多种 OOA/OOD 方法基础上形成的建模语言,继承了各种面向对象和一些结构化建模语言的优点。它不仅融入了软件工程领域的新思想、新方法和新技术,同时也吸引了许多学者的研究,并得到了很多 IBM、微软等软件厂

^{*} 基金项目:国家自然科学基金(60273043),安徽省自然科学基金(050420204),安徽省教育厅自然科学基金项目(2006KJ098B, KJ2007B153)。
李学俊 博士研究生,讲师,研究方向为 Web 挖掘、机器学习、智能软件;李龙澍 硕士,教授,博导,研究方向为智能软件、知识工程、软件体系结构;程慧霞 教授,研究方向为人工智能、软件体系结构;徐 怡 博士研究生,研究方向为粗糙集、智能软件。

商的支持,因此已被对象管理组织 OMG 确定为标准的面向对象建模语言。目前最新的 UML 规范说明是 2004 年 10 月发布的 UML2.0 版本。UML 规定了一整套建模元素和相应的可视化描述,用于描述、说明、可视化地构造和建档软件开发各个阶段所产生的制品,如模型、源代码、测试用例等。UML 的基本建模元素包括类、对象、参与者(Actor)、用例(Use Case)、包(Package)、关联(Association)、聚合(Aggregation)、组合(Composition)、依赖(Dependency)及泛化(Generalization)来表示元素及其间的关系,以构成 UML 中的各种图(Diagram)。这些图分别是描述系统功能的用例图(Use Case Diagram);描述系统静态逻辑结构的类图(Class Diagram)、对象图(Object Diagram)和包图;描述系统动态行为特性的顺序图(Sequence Diagram)、通信图(Communication Diagram)、交互概观图(Interaction Overview Diagram)、状态机(State Machine)和活动图(Activity Diagram);描述系统物理实现的组件图(Component Diagram)和部署图(Deployment Diagram)。为了满足不同人员从多个角度描述软件系统,UML 提供了 5 种预定义视图(View):表示系统功能需求的使用例视图(Use Case View)、表示系统概念设计和子系统结构的逻辑视图(Logic View)、说明源代码及其结构的实现视图(Implementation View)、说明系统中并发执行和同步情况的进程视图(Process View)、定义系统运行硬件节点的物理结构的部署视图(Deployment View)。视图是 UML 中图的组合,用户也可以根据需要自定义视图。

UML 得到了软件厂商的支持,出现了很多支持 UML 建模的工具软件,如 IBM 公司的 Rational Rose 2003、Microsoft 公司的 Visio Professional 2003、Borland 公司的 Together 2006 等。其中 Rational Rose 2003 是当前应用最广泛的 UML 建模工具之一,它最大的优点是与其他工具无缝集成,例如需求管理工具 Rational RequisitePro、评估软件性能的工具 Rational Quantify 等。

独立于软件开发过程的 UML 统一了面向对象软件开发的建模语言,为用户、分析人员、设计人员、开发人员、测试人员、系统集成人员建立了便于交流的可视化、规范化、结构化、文档化的共同语言。

3 基于 UML 的 Wiki 系统建模

3.1 Wiki 系统的功能分析

基于 Wiki 系统平台,网民使用 Wiki 语言编辑页面,Wiki 引擎把 Wiki 页面转换成 Html 页面,返回到客户端,同时自动记录 Wiki 页面的编辑版本,构建成 Wiki 知识库。Wiki 系统的实现机制如图 1。Wiki 系统可以实现 Wiki 页面编辑、版本比较、知识分类查询、系统管理等功能。

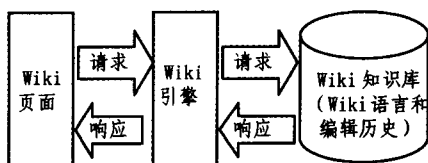


图 1 Wiki 系统的功能及其实现机制

3.2 功能需求模型

参与者是系统以外的、需要使用系统的或者与系统交互的人、外部系统、设备等。用例是参与者使用系统的一项功能时所进行交互活动的动作序列。用例图是显示一组用例、参

与者及其间关系的图,它用于描写系统的功能性需求,是其它 UML 图的核心和基础。系统的最终目标就是现实用例图中的所有功能^[1,4,5]。图 2 是 Wiki 系统的总体需求模型,该用例图由两个参与者和所驱动的六个用例相互关联组成,其中两个参与者:Wiki 用户和系统管理员;六个用例:注册、登录、页面编辑、版本比较、知识分类查询以及系统管理功能。在建立系统总体模型后,需要进一步细化需求模型。图 3 是系统管理子系统需求模型,以“系统管理”用例为例,该用例可以分解为“帐户管理”和“页面管理”,其中“页面管理”由页面添加、页面保护、页面移动、页面删除、页面参数设置五个用例聚合而成。通常,还需要为重要复杂的用例建立一些以顺序图表示的行为模型,以便更加清晰地描述这些用例。

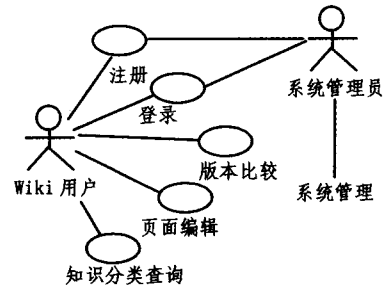


图 2 Wiki 系统总体需求模型——UML 用例图

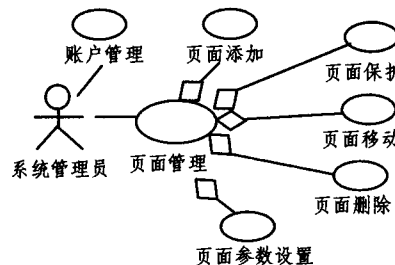


图 3 系统管理子系统需求模型——UML 用例图

3.3 数据模型

通常采用 E-R(Entity-Relationship)图进行数据建模,但是它只能表示数据,不能对行为建模,如不能为数据库中的存储过程、触发器建模。本文采用 UML 中的类图的扩展 Data Modeler 描述数据模型。对于关系数据库来说,类描述数据库表,类的操作描述存储过程、触发器,类图描述数据库模式^[4,5]。

为了实现在 Wiki 知识库中存储 Wiki 语言形式的知识以及编辑历史信息,需要构建相应的数据模型,如图 4 所示。该图仅给出了数据模型中的重要实体:WikiPage(Wiki 页面)、Revision(编辑历史信息)、WikiText(Wiki 语言形式的知识)、User(Wiki 用户)。每个实体的简要描述如下:

WikiPage: PageID(Wiki 页面编号)、PageTitle(Wiki 页面标题);

Revision: RevID(历史版本编号)、PageID(Wiki 页面编号)、TextID(Wiki 知识编号)、UserID(用户编号)、EditDate(编辑时间);

WikiText: TextID(Wiki 知识编号)、Text(Wiki 知识);

User: UserID(用户编号)、UserName(用户名)、Password(密码)、RegisterDate(注册日期)。

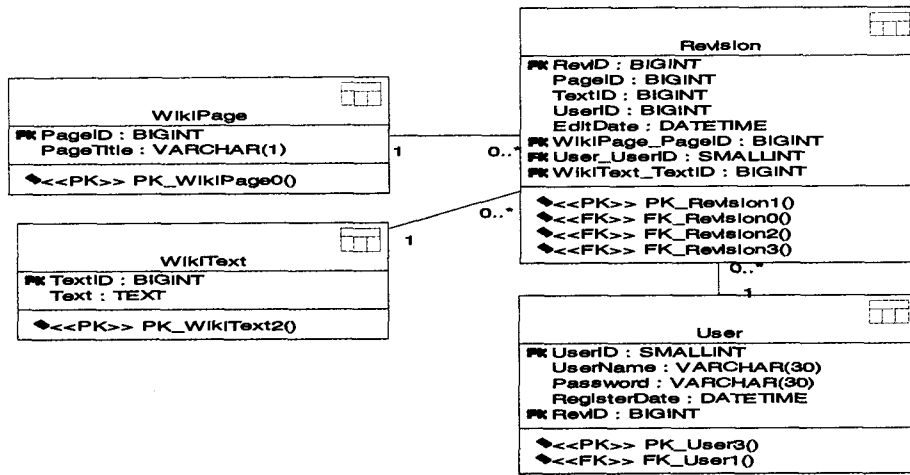


图 4 Wiki 系统数据模型——UML 类图扩展

基于上述数据模型,可以实现 Wiki 系统的主要功能——Wiki 页面编辑。利用 UML 建模工具 Rose 的正向工程,还可以生成建立数据库的 DDL 脚本或者直接创建数据库。

3.4 Web 模型

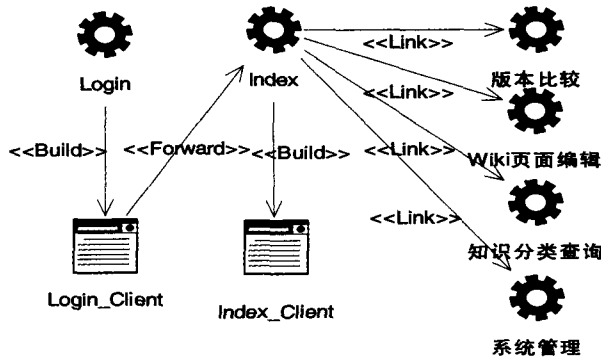


图 5 Wiki 系统 Web 模型——UML 类图 WAE 扩展

采用 Web 模型表示 Web 应用系统的模型元素及其间关系,以描述其体系结构。由于 UML 没有预定义应用于 Web 建模的模型元素,因此采用 Jim Conallen 提出的 UML 的扩展

机制 WAE(Web Application Extension for UML)构建 Web 系统模型。在 WAE 机制中,Web 模型元素包括服务器页、客户机页、表单以及 Build、Link、Forward、Submit 等关联^[4,5]。Build 表示服务器页创建客户机页,Link 表示两个客户机页之间或一个客户机页到一个服务器页的超链接,Forward 表示对于 JSP 页面,一个服务器页到另一个服务器页或客户机页的重定向。基于 WAE 机制,构建 Wiki 系统的主要功能对应的 Web 模型,如图 5 所示。该图清晰地描述了 Wiki 系统的构成元素及其软件体系结构。从图 5 中的 Web 模型,可以看出该 Wiki 系统具有的功能,同时也很清晰地反映出该 Web 应用系统的网络拓扑结构。

4 应用实例

我们将 UML 应用于 Wiki 系统的设计与实现,采用 UML 建模工具 Rational Rose 2003 构建了 Wiki 系统的功能需求模型、数据模型和 Web 模型。在上述模型的基础上,改进现有的运行于 PHP+MySQL 环境的 MediaWiki 1.5rc4 系统,部署实现了改进的 Wiki 系统——Oracle 网络知识库,如图 6 所示。通过这种 Wiki 形式的网络知识平台,可以集中大家的智慧,共同构筑 Oracle 网络知识库,共同学习 Oracle 技术。

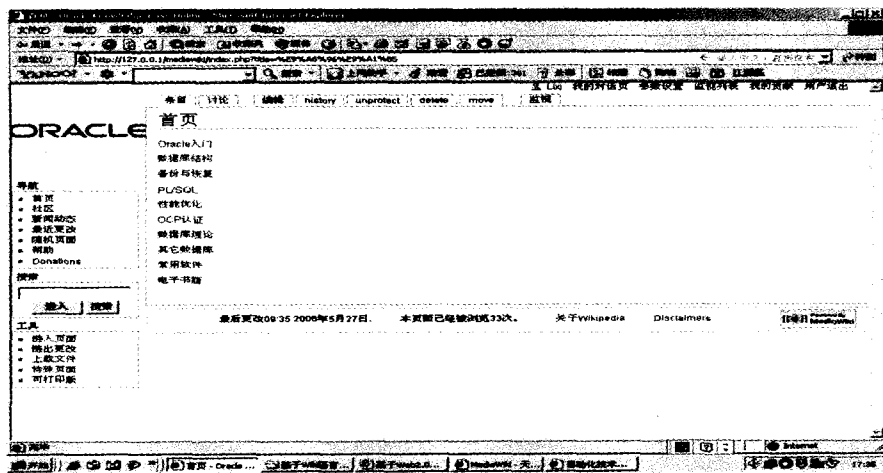


图 6 基于 Wiki 系统的 Oracle 知识库

结束语 本文建立的基于 UML 的 Wiki 系统模型,大大提高了 Wiki 系统的软件重用和开发效率,并有利于 Wiki 系统的标准化和多样性。

然而,UML 缺乏形式化语义,对体系结构的描述只能达到非形式化的层次,对软件体系结构的可构造性建模能力较 (下转第 274 页)

显然,采用无主机模式的好处在于各进程间不需通过通信也能够形成自己所需的模型信息,并且能避免复杂的不连续数据通信,而且读取数据的代码与串行子结构程序完全相同。因此,本平台采取了这种模式来建立并行有限元模型。

5.2 并行有限元求解器

并行求解器是并行有限元平台开发的重点和难点,相对工作量比较大。

无论采用何种并行求解器,各进程在进行并行求解过程中都需要通过互相通信来交换数据。与串行程序的编制和调试相比,并程序的编制与调试要复杂得多(一般不能通过设置断点进行调试,而只能通过打印信息进行调试),因此,设计一个简洁的并行代码显得非常重要,这不仅有利于程序的调试,还可以减少错误的产生。本平台采用的数据通信通道技术能有效体现这一特性。

6 算例

对如图8所示的悬臂梁结构进行损伤分析,梁的截面尺寸和配筋如图8。将该梁划分为2000个单元,6321个节点,荷载划分为20步逐步施加。

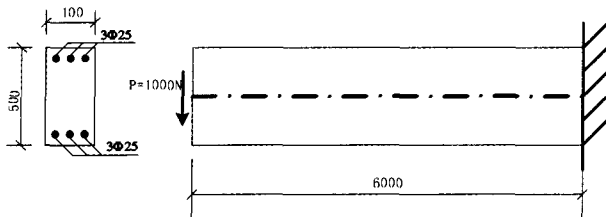


图8 算例示意图

进行并行分析时,分别将该梁划分为2个子结构和4个子结构,计算环境为由4台PC组成的机群(每台配置AMD1.8GHzCPU、256MB内存、window2000操作系统,网络传递速度为每秒100MB)。计算分析的时间见表1。

表1 串行程序及并行程序的分析时间对比

| 比较对象 | 一步迭代的时间分配(秒) | | | | 总分析时间 (190 迭代步) |
|-------|--------------|-------|------------------|--------|--------------------|
| | formK | formF | solve | update | |
| 串行 | 0.26 | 0.20 | 3.24 | 83.70 | 4h40m |
| 2台处理机 | 0.14 | 0.10 | 6e ⁻⁴ | 39.76 | 2h15m |
| 4台处理机 | 1.66 | 0.05 | 0.013 | 17.90 | 1h8m |

从表1中可以看出,每一迭代步中的update部分占总分析时间最大的比例(基本大于90%),主要花在计算各单元高斯点上损伤系数的过程中。经并行处理及实现计算与通信重叠后,这一过程所需的时间明显减少,从而降低了整个分析过程所要的时间(2台处理机和4台处理机的计算时间均达到

120理想的加速比),明显地展示出并行计算的效果。

结论 面向对象并行有限元数值平台具有良好扩充性,并且维护非常方便。该平台除了为用户提供一个健壮的有限元体系外,还为用户提供多种优化的系统方程存储方法、多种高效的线性方程组求解器,以及丰富的数学工具,可帮助用户快速地开发串行或并行有限元程序,最大限度减轻用户的开发工作量,使用户把更多的精力集中在开发与其研究目的有关的单元和材料模型上。

本平台采用了两种并行策略,以MPI并行程序设计函数库为并行支撑环境,实现了面向对象有限元并行计算功能。对想用并行程序的用户来说,他们可不需了解并行编程,所作的开发工作与串行程序完全一样,而并行计算功能由平台提供。从而,极大地方便了用户的使用。

对于该平台,本课题组目前正在进行和今后将要进行的工作有:

(1) 丰富平台的单元类型和材料模型

该平台已开发了梁杆单元、平面单元、板单元等,但单元的材料本构模型需要进一步增强;

(2) 增加新的分析方法

传统的非线性分析方法如增量法、迭代法(Newton-Raphson、修正Newton-Raphson法)和混合法一般只能分析材料的上升段,而不能分析材料的下降段。在这种情况下,需要开发其它更有效的分析方法如逐步搜索法、虚加刚性弹簧法、位移控制法、弧长法等;

(3) 增加新的并行策略及开发更高效的并行求解器

当在较小的并行环境下(处理机节点数较少)进行大规模分析时,本分析平台有可能因单机内存的限制导致计算效率不理想。在这种情况下,并行多重子结构法是非常好的解决方案。

参考文献

- 阮红河,袁勇,柳献.分布存储环境并行有限元研究进展[J]. 同济大学学报(自然科学版), 2005, 33(1):21~27
- Khan A I, Topping B H V. Parallel finite element analysis using Jacobi-conditioned conjugate gradient algorithm[J]. Advances in Engineering Software, 1996, 25:309~319
- 曹骥. MPI环境先进有限元程序方法研究[D]. 上海同济大学建筑工程系, 2002
- 袁勇,曹骥. 机群先进计算机的结构分析平台[J]. 结构工程师, 2003, 66:23~329
- Papadrakakis M, Bitzarakis. Domain decomposition PCG methods for serial and parallel processing[J]. Advances in Engineering Software, 1996, 25:291~307
- 都志辉. 高性能计算机并行编程技术—MPI并行程序设计[M]. 清华大学出版社, 2001
- 阮红河. 结构分析理论的并行有限元方法[D]. 上海同济大学建筑工程系, 2004
- 和力,鱼滨,和燕,吴丽贤. 一种用UML开发组件式Web应用系统的过程[J]. 计算机工程, 2005, 31(1):95~97
- 王少锋. 面向对象技术UML教程[M]. 北京:清华大学出版社, 2004
- 埃里克森(美),等著. 余安萍,等译. UML2工具箱[M]. 北京:电子工业出版社, 2004
- MediaWiki引擎源码下载[EB/OL]. <http://www.cncode.com/downinfo/121.html>, 2005, 8
- 天下维客[EB/OL]. <http://www.allwiki.com/>, 2005, 9
- 周莹新,艾波. 软件体系结构建模研究[J]. 软件学报, 1998, 9(11):867~872

(上接第253页)

弱。因此,可以研究将形式化语义描述能力强的层次状态图、Petri网、进程代数、体系结构描述语言ADL和UML结合起来描述软件体系结构,以弥补UML非形式化的不足^[2]。

参考文献

- 李玉萍,毛少杰,张桂林. 面向对象技术和UML在剧情产生系统中的应用[J]. 系统仿真学报, 2005, 17(6):1311~1314
- 戎玫,张广泉,刘艳. 基于软件体系结构和UML的图书管理系统设计与实现[J]. 计算机科学, 2005, 32(6):224~227