

基于 UML 的文档管理系统研究^{*}

甘利杰

(重庆工商大学计算机科学与信息工程学院 重庆 400067)

摘要 统一建模语言 UML(Unified Modeling Language)是面向对象软件开发方法的重要技术。UML 可以对大型复杂的系统的各种成分可视化,说明并构造系统模型,以及建立各种所需的文档。本文以文档管理系统为例,介绍了 Intranet/ Web 体系结构的信息系统站点的构建,以及基于 ASP 和 ISAPI 的动态 Web 页面发布技术,并对基于 Intranet 和 Web 技术应用的一些典型问题进行了分析。

关键词 统一建模语言,文档管理,Intranet

Research on Document Management System Based on UML

GAN Li-Jie

(Computer Science & Information Engineering College, Chongqing Technology & Business University, Chongqing 400067)

Abstract The unified modelling language(UML), which is introduced in this article, is an important tendency of object-oriented software development methodology. UML can make each component of the large complicated system visual, explain and construct the system model, and build up various documents. This paper describes the architecture and supporting techniques of Web-based Intranet, and then focuses on the development of dynamic Web pages through ASP and ISAPI techniques. At last, some typical issues in Intranet-based Web application are introduced and the implementation of an enterprise document management system is taken as an example.

Keywords Unified modeling language, Document management, Intranet

1 引言

UML 是一种面向对象的可视化的标准建模语言^[1],以图形符号表示系统中的对象及其之间的关系,从不同的角度描述待开发的系统^[2~4]。UML 的建模过程^[1]主要有以下几个步骤:

(1)建立需求模型:即从功能需求出发建立用例模型,得到系统的功能。

(2)建立对象模型:包括静态模型和动态模型。静态模型使用类图、对象图、包图描述系统的对象和对象之间的静态关系,得到系统的整体结构。动态模型使用状态图、活动图、顺序图、合作图描述对象的状态和状态之间的转换条件,各个对象之间的交互关系。

(3)建立系统实现模型:使用配置图定义系统的软硬件结构及通讯机制,表示软硬件系统之间的合作关系;使用构件图描述系统由那些构件组成。

(4)检查模型之间的一致性,通常这个过程需要反复多次才能完整地描述系统。

(5)在构件图的基础上生成开发语言的代码框架。

UML 的主要特点^[1]可归纳为以下几点:(1)统一的标准;(2)面向对象;(3)可视化,表示能力强大;(4)独立于过程;(5)概念明确、建模表示法简洁,图形结构清晰,容易掌握和使用。

随着计算机的广泛应用,传统的纸质文档管理已经不能满足异地协同设计和远程信息共享的需求。Internet 和 Web 的技术为全球化电子信息的共享提供了基础^[5]。

文档管理系统的对象是大量数据,其中一部分是能够以报表形式组织的结构化数据,存储到各类数据库中;另一部分是数量庞大的非结构化数据,即工作文档。

工作文档的非结构化特征,对文档管理系统提出了特殊要求,单靠建立简单的一个或多个文档服务器是显然不能满足要求的。而专用平台开发的文档管理系统,会受到系统升级、用户培训、客户端安装等问题的困扰,无法受到普通用户和计算机技术人员的欢迎。

基于 Web 的 Intranet 给文档管理提供了机遇,Intranet 能够实现集中式的存储和安装计算机的访问许可。基于 Web 的文档管理系统被认为是分布式产品数据管理的自然扩展。围绕文档管理功能,进行基于 Web 的文档管理关键技术研究,开发基于 Web 的文档管理系统,具有理论价值和广阔的应用前景。

2 文档管理系统设计

文档系统的基本组织形式是目录,系统根据用户的要求把文档按目录分类归档保存。文档系统要求用户提交文档时,除了文档本身还应该有一些其他的信息,以方便今后对文档检索。因为这个系统不限定用户提交的文档的类型,所以用户必须提交一些附加信息,帮助系统快速定位文档。因此,系统定义了关键字集的数据段,用户在这个数据段中针对性写摘要,或者关键词组合。系统可以根据用户的请求对这个字段全文检索。通过这种方法,系统能够实现较强的检索功能。

采用 Web 三层 B/S 结构模式开发文档管理系统将商业

^{*}本研究由重庆市自然科学基金支持项目(No. 2004BB2182)基金资助。甘利杰 主要研究方向:软件工程、操作系统等。

逻辑从客户端分离出来集中在中间层服务器上,以浏览器作为客户端,从而避免了传统 C/S 结构的局限。其体系结构如图 1 所示。

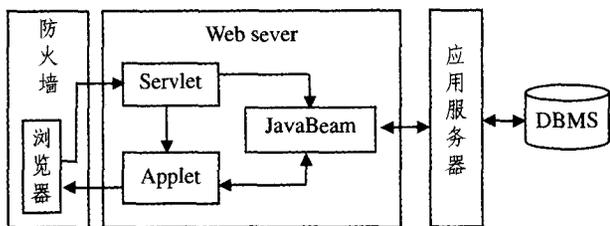


图 1 B/S 模式的文档管理系统

(1)第一层为客户层,它是一个图形用户界面。在这一层运行的程序是浏览器,不需要完成任何重要的业务逻辑,也不直接和数据库交互。

(2)第二层是中间层,负责管理对用户请求的处理,将完成文档管理全部功能。

(3)第三层是数据处理层,是数据库和应用服务器,存储和管理应用中的数据和文件。它和中间层共同完成业务逻辑、验证和存储的实现。

3 文档管理的重要环节

文档管理系统各功能的实现,从传统的面向功能设计转向面向对象设计,主要体现在采用了组件技术。

3.1 Web 客户端

文档管理系统所要处理的业务逻辑很复杂,采用 Applet 技术的 Swing 组件模式。利用 Applet 来表现页面,克服了 JSP 页面表现的单调性;利用 Swing 组件的 MVC 模式,将表现层与数据逻辑及其控制分开,提高程序的扩展性。

通过 Applet 访问服务器后端的 Servlet,由 Servlet 服务器程序作为后台控制程序,来完成大量处理。Servlet 具有控制功能,根据客户端请求的类型,访问中间件的业务逻辑,按需求然后创建 Applet 需要的 JavaBean 对象,最后将处理结果以 model 的形式发送给用户。并通过灵活运用 Swing 中的组件来丰富和控制 Applet 页面的动态显示。图 2 为 Web 服务构件与其他构件的交互序列图。

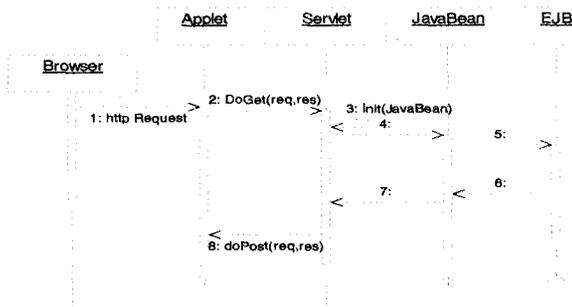


图 2 交互序列图

3.2 Web 服务器与应用服务器的远程连接

在远程通信中,浏览器作为 Web 服务器的客户端实现“请求-响应-返回请求”第一次远程连接,通过 Servlet 提供的 HttpSession 等常用类接口获取客户端信息,实现在 HTTP 客户端和 HTTP 会话两者的关联。进而将 Web 服务器作为应用服务器的 Java 客户端,实现完成系统各种业务逻辑的第二步远程连接。在第二次远程连接中,Web 服务器不直接引用应用服务器的实体 Bean 通信 Home 接口,而是通过会话

Bean 保持与实体 Bean 的会话。然后通过 Java 的名字和目录接口定位会话 Bean 的 Home 接口,继而获得会话 Bean 的 Remote 接口,通过该接口中提供的方法实现远程业务逻辑。

3.3 文档管理的组件层次

在应用服务器端主要使用两个会话 Bean 来完成文档管理的业务逻辑:一个会话 Bean 用来访问若干实体 Bean,根据文件类型的不同,每个实体 Bean 对应一个不同的表格。通过文件的类型对数据库中不同表格的记录进行增加、修改等操作,实现文档的签入、签出等操作的数据库记录属性值的改变;第二个会话 Bean 用来访问系统管理模块的有关实体 Bean,实现文档实际物理存储位置的变化。文档管理的组件层次见图 3。

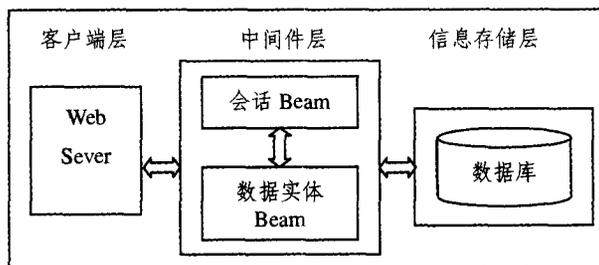


图 3 文档管理的组件层次

4 安全控制策略

WWW 服务使用的 HTTP 协议固有的静态特点,使 WWW 服务器只能响应单个的请求,不能保留用户的前后使用信息。然而,有效的安全策略需要知道用户的所有信息,并且跟踪用户的动作,从而设计了会话跟踪技术。

当每个用户端连接进入系统时,服务器分配一个 GUID (Global Unique Identifier)给客户端,GUID 按照一定的算法生成,长度足够,不同的登录得到一样的 GUID 的概率几乎为零。客户端应用程序记录下 GUID,以后每次向服务器提出请求时,都随同请求一起发送 GUID 以表示身份,GUID 对用户不可见,只记录在程序中。用户退出系统后,GUID 被清除,不在客户端计算机中存有副本。因此,用户和系统的连接的安全性得到了保证。用户不必担心自己的登录信息被使用同一计算机的其他用户窃取。因为,首先 GUID 与用户没有必然的联系,完全是随机的,只在客户端操作的用户不可能知道当前有哪些 GUID 在被用作活跃用户标志,更不可能知道哪个用户用的是哪个 GUID。其次,只在客户端的使用者也不可能知道自己的 GUID 和篡改自己的 GUID。

服务器端程序实时记录进入系统的用户信息,并且通过 GUID 跟踪记录每个用户的操作。关键时刻,实现管理员对在线用户的控制,如强制退出、限制行为等等。服务器接受用户的操作请求时,首先根据随同请求收到的 GUID 从在线用户表中取得用户信息,如果该 GUID 不在线用户表中就拒绝该用户的请求。然后,进行权限检查、执行命令等等操作。最后返回信息给客户端计算机。

用户退出处理的一个重要方面是保证安全性。用户退出处理必须在服务器端和客户端都彻底清除用户的在线信息。另外,还采用了强制不缓存所有页面的方法作为补充。因此,假设用户退出系统时,浏览器没有关闭,但 GUID 被清除和所有系统信息没有缓存,后来用户不可能在浏览器中冒充原用户或者取得原用户的任何操作结果。服务器端决定某个用户

退出时,就会删除相应的在线用户记录。这样,从服务器端保证了即使有人取得已退出用户的 GUID,也不能冒充那个用户。

5 文档管理的实现

文档管理系统将所有的文档信息分为两部分:一部分称为元数据,即文档的描述信息,如文档名称、类型、创建日期、版本、所有者以及存取的路径等,保存在数据库表格中;另一部分即具体的物理文件,存放在磁盘上。从而建立起数据库表格,每一条记录与一个物理文件的连接和对应关系。

原则上每一种文档对应一个数据库表格,但是由于文档种类繁多,为了便于用计算机管理,对各种文档从总体上进行分类管理。这样,系统屏蔽了文档存储的实际物理位置。当 Web 用户进入系统界面单击某一文档对象时,便获得了操作该对象的指针,在对该对象进行签入、签出和复制等操作时,系统自动到相应的表格中进行记录的修改和增加,来跟踪文档信息的变化,把文档的元数据存入元数据库中,相应的具体文档则放入应用服务器中指定的文件系统的相关路径中。

5.1 文档的上传和下载

在 Web 客户端,文档签入实质进行的是文档上传操作。Web 用户选定要签入的电子仓库后,签入文档首先从 Web 客户端上传到 Web 服务器,再从 Web 服务器通过文件流保存到应用服务器中,并将元数据存入数据库保存。文档上传使用自定义的 Upload 组件实现。当客户端提交上传文件的 Form 表单时,Web 服务器就调用 Upload 来完成文件上传的请求。

5.2 Web 的文档管理

Web 的文档管理功能包括:

- (1) 文档的版本控制,如文档在检出、修订时版本的增加和扩展;
- (2) 文档的操作,包括文档的创建、注册、查询,文档的检入、检出、复制、删除,以及文档的冻结、修订等;
- (3) 文档的安全控制,通过用户注册登录口令认证以及文档操作过程中角色权限认证确保文档管理的安全性;
- (4) 消息系统,通过系统内建的 JMS 同步和异步消息,实现文档管理的协同工作;

(上接第 230 页)

发挥重要作用。因此,SOA 为模型转换技术的发展提供了一个新方向——面向服务的模型转换技术,不仅能够实现 PIM 到 PSM 的自动转换以及 PIM 之间的精化,还能为业务和服务提供不同抽象层次、不同视点的模型转换,从而实现 SOA 中的服务。

参考文献

- 1 Bézivin J. From Object Composition to Model Transformation with the MDA. In: Proceedings of TOOLS' USA, IEEE TOOLS-39, Santa Barbara, August 2001
- 2 Kleppe A, 等著. 解析 MDA. 鲍志云译. 人民邮电出版社, 2004
- 3 Frankel D S 著. 应用 MDA. 鲍志云译. 人民邮电出版社, 2003
- 4 Caplat G, Sourrouille J-L. Model Mapping Using Formalism Extensions. IEEE Software, 2005, 44~51
- 5 Oldevik J, et al. Framework for model transformation and code generation. In: Proceedings of the Sixth International Enterprise Distributed Object Computing Conference (EDOC'02), IEEE, 2002
- 6 Czarnecki K, Helsen S. Classification of Model Transformation Approaches. OOPSLA'03 Workshop on Generative Techniques

(5) 邮件系统,便于系统用户进行信息交流。

5.3 Dynamic HTML 技术

经典的 Web 站点所发布的页面都是静态的。而目前的动态 HTML,把静态的属性标签增强成可以动态访问和修改的对象。

5.4 动态页面发布

在文档管理系统中,根据实际情况采用了微软的 ASP 和 ISAPI 技术。

文档管理系统在所有涉及数据库操作的 ASP 动态页面中使用了 ADO 对象。而且,在 ADO 的数据源中,指定使用 SQL OLEDB Engine For SQL Server。实际的测试表明,这样实现虽然在与数据库建立连接时需要稍多的开销,但进行数据查询和数据修改操作的效率却令人满意。

对数据库操作进行优化,对某些常用的联系紧密的数据库操作,编写数据库存储过程,像函数一样通过 ADO 对象调用。这样便于 SQL Server 对数据操作进行优化,减少数据在 DCOM 对象间传递的次数,保证数据库操作的高效性。

结束语 Web 文档管理系统通过对公司的人员分组,使得开发过程中的软件开发人员和数据处理人员能够通过消息及时沟通,保持数据同步,较好的协同工作,集中管理,分散存储也利于数据的备份和恢复,保证了数据的安全性。

通过 Java 语言良好的跨平台特性和面向对象的设计方法使基于 Web 的文档管理系统具有强大的可移植性和较少的代码量。对于打破部门之间空间和时间形成的边界,更好地管理各部门的协同具有十分重要的实用价值。

参考文献

- 1 刘超. 可视化面向对象建模技术-标准建模语言 UML. 北京航空航天大学出版社, 1999
- 2 [美] Roger S. Pressman 著, 梅宏, 译. 软件工程实践者的研究方法 (原书第 5 版). 机械工业出版社, 2003
- 3 [美] Gamma E, Helm R, Johnson R, et al. 设计模式: 可复用面向对象软件的基础. 北京: 机械工业出版社, 2005
- 4 方浩, 程传庆, 王金亭. 基于 UML 的库存管理信息系统设计. 武汉理工大学学报(交通科学与工程版), 2004, 28(8): 626~628
- 5 冯东雷, 应刚, 顾春华. Intranet 的原理与应用-企业信息系统的建设. 上海: 上海交通大学出版社, 1987

- in the Context of Model-Driven Architecture
- 7 <http://drops.dagstuhl.de/opus/volltexte/2005/11/pdf/04101.SWM2.Paper.pdf>
- 8 Sunye G, et al. Using UML Action Semantics for Executable Modeling and Beyond, Springer-Verlag, 2001, 433~447
- 9 Braun P, Marschall F. Transforming Object Oriented Models with BOTL. Electronic Notes in Theoretical Computer Science, 2003, 72(3)
- 10 Sendall S, et al. Model transformation: The heart and soul of Model-Driven Software Development. IEEE Software, 2003, 42~45
- 11 Agrawal A. Graph Rewriting And Transformation (GRaT): A Solution For The Model Integrated Computing (MIC) Bottleneck. 18th IEEE International Conference on Automated Software Engineering (ASE'03)
- 12 Raistrick C. Applying MDA and UML in the Development of a Healthcare System. LNCS 3297, Springer-Verlag, 2005, 203~218
- 13 Duddy K, et al. Model Transformation: A declarative, reusable patterns approach. In: Proceedings of the Seventh IEEE International Enterprise Distributed Object Computing Conference (EDOC'03)
- 14 <http://www.modelbased.net/mda--tools.html>