

# 基于 J2EE 技术和工作流引擎构建 B/S 模式的文献资料管理系统

## Building B/S Model of the Literature Information Management System Based on J2EE Technology and Workflow Engine

任 军 李胜利 (烟台东方电子信息产业股份有限公司电自部 264000)

**摘要:**本文介绍了 B/S 模式下文献资料管理系统的设计思路和体系构成,并详细说明了系统开发中的 J2EE 技术、组件技术、工作流引擎和报表实现等关键技术。

**关键词:**J2EE B/S 工作流引擎 组件技术 文献资料管理系统

### 1 引言

企业科技文献资料的管理工作,主要包括采购申请、采购审批、采购入库、用户注册/登录、资料查询、资料借阅和资料归还等,其流程是一项复杂、琐碎、操作频繁的工作,仅仅依靠传统的手工管理或简单的微机操作等已不适当当前企业科技管理工作迅速发展的要求。实现文献资料管理工作网络化、现代化成为必然。

近年来,随着计算机网络技术、应用开发技术的发展与普及,B/S 模式的管理系统越来越被人们所青睐。

器下运行即可。系统需要安全的网上数据传输、高效的性能,还要满足先进性、可扩展性、可移植性等系统设计原则,因此采用先进的 J2EE 技术来开发部署多层 B/S 体系结构的文献资料管理系统。充分利用服务器资源,提供了跨平台、简单一致的应用环境,实现了开发环境与应用环境的分离,故而减少了软件开发及维护的工作量,节约了系统费用,更便于系统用户群的扩展。

### 2 系统总体设计

#### 2.1 系统结构设计

文献资料管理系统由采购申请模块、采购审批模块、采购入库模块、用户注册/登录模块、资料查询模块、资料借阅模块和资料归还模块,以及基本信息模块和工作流引擎模块组成。整个系统分为数据服务层、应用服务层和表现层,客户端只运行浏览器,所有的应用程序和数据库

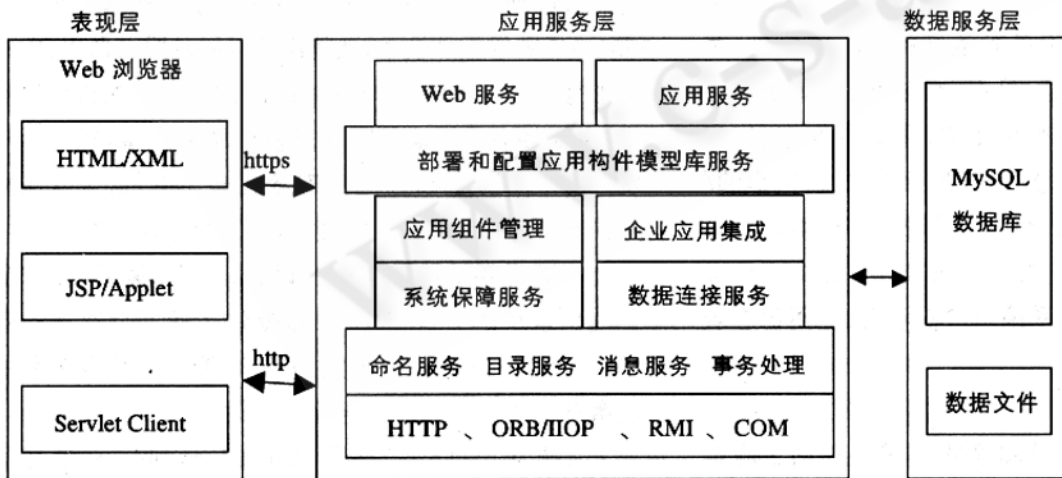


图 1 系统体系结构

B/S 模式应用系统只需在服务器上安装配置应用服务器软件,在客户端以“瘦客户”或“零客户”方式在浏览

器均运行在服务器上(如图 1 所示)。

表现层:是用户进行操作的系统最外层,在浏览器

中实现所有功能。内部用户可通过局域网根据系统赋予的不同权限进行操作和管理;外部用户经由防火墙可以安全访问。

**应用服务层:**是系统的核心组成部分,根据 Web Server 发送的请求进行分时、分类部署组件,调用 JavaBean/EJB 组件、从 Database Server 请求数据。其包含的 Web Server 响应用户的各种请求,并将相应的请示经过转换后送到应用服务层的各应用组件进行处理,得到处理结果后再将结果返回到用户表现层。其中, workflow 引擎应用组件实现 workflow 具体的过程定义、建模、执行和监控等功能。

**数据服务层:**存储系统所有的数据和管理文档,对应用服务层进行计算和处理时所需的数据请求进行响应并返回结果。可以分阶段进行数据备份以免因不可控因素丢失数据。

## 2.2 数据库设计

系统选用了开放的关系型数据库 MySQL 作为后台数据库。选用其目的是:第一,MySQL 应用简单方便,并且支持 JDBC - ODBC 连接访问;第二,MySQL 是开放的数据库软件,支持关系型数据库设计和应用,节省大笔企业费用投资,且易于维护管理;第三,文献资料管理的报表有二十多张,数据量在十万条数据左右,在数据库领域中属于中等规模应用,MySQL 完全可以胜任,且能保证系统运行性能。

通过对文献资料管理系统功能的分析,设计了文献资料管理数据库及其中各表的结构,如:字段名、字段类型、字段长度、表主键、是否允许空值等。该库中主要包括采购申请表、采购审批表、采购入库表、用户注册信息表、用户登录信息表、资料归档表、资料借阅/归还情况表、资料领域表、资料类别表、归档类型表等基本业务信息库表,以及 workflow 分类表、workflow 定义表、workflow 处理信息表和系统用户权限表、应用日志表等系统运行信息库表。

## 2.3 服务器设计

瘦客户机制的关键,是将逻辑处理转移到了服务器。作为系统的核心,应用服务器负责处理商务逻辑。商务逻辑是指存取,生成,认证数据和执行处理的代码。特定的应用商务逻辑由可重用的组件组成。一旦组件生成,就可把它部署在应用服务器中,赋予安全设置,然后就可运行。应用服务器提供了运行这些组件

的框架,并能够指定组件的持久性,事务处理要求及线程要求。

应用服务器 (Application Server) 适应业务需求,其每个组件可以方便地与数据源绑定,该组件就可以方便地访问数据,对数据进行各种操作。本系统设计应用服务器基于 J2EE 组件技术,以 EJB (Enterprise JavaBeans) 为中心的服务器端的软件组件技术。EJB 是目前以 Java 语言为前提的组件技术规范,主要在服务器端运行 Java 应用,利用其跨平台功能及其它 Internet 技术及中间件和分布对象功能,构筑大型应用系统。在应用服务器环境中,事务处理由服务器执行。开发者可以编写代码用来发出 SQL。通知服务器逻辑是否成功。如果在某个标记为一个事务的调用序列中的某个组件失败,服务器就会自动发出一个回滚命令 (Roll Back),如果全部成功就提交。服务器提供各种层次的事务控制,例如多 workflow 引擎、workflow 事务处理和 workflow 安全性等。其客户端有三种类型,一是 JSP/Servlet 生成客户端,二是运行 Java Applet,三是 HTML/XML 页面。

## 3 系统实现关键技术

### 3.1 J2EE 技术

J2EE 是由 SUN 等众多厂商共同提出方案和制定规范的开放平台,能够满足当前不断变化和日趋复杂的商业应用需求,成为企业构建新系统和集成、扩展已有系统的首选技术。J2EE 技术规范提供了 JSP、Servlet、JavaBean/EJB 等组件技术,使得用户可以从服务器平台、开发工具、中间件等众多产品中进行选择来组成自己的应用;中间件应用提高了系统开发效率和质量,减少了开发人员工作量;J2EE 平台的兼容性与扩展性,可以保护用户原有的软件与硬件投资,减少了费用和 risk。

J2EE 提供了基于组件的设计、开发、集成、部署企业应用系统的方法,J2EE 平台提供了多层分布式的应用系统模型、重用组件的能力、统一的安全模型和灵活的事务控制。

J2EE 应用可以是基于 Web 的,也可以不是基于 Web 的。在一个基于 Web 的 J2EE 应用中,用户的浏览器在客户层中运行,并从一个 Web 服务器上下载 Web 层中的静态 HTML 页面或由 JSP 或 Servlets 生成的动态 HTML 页面。在一个不基于 Web 的 J2EE 应用程序中,一个独立的客户逻辑程序,并不运行在一个 HT-

ML 页面中,而是运行在系统其他逻辑处理中,在不经 Web 层的情况下访问 EJB。该不基于 Web 的客户层可能也包括一个 JavaBeans 类来管理用户输入,并将该输入发送到在企业层中运行的 EJB 类来处理。

### 3.2 组件技术

组件技术是为了解决应用软件开发面临的重复编码严重、开发效率低而提出的,其目的是实现软件重用与功能扩充。利用组件技术不需要开发和设计多个应用系统,而是可以通过分析多模式系统的核心功能,合理进行软件重用,使得开发的系统有效地同时支持不同应用模式。组件化程序设计方法继承并发展了面向对象程序设计方法,将面向对象技术应用于系统设计,使应用系统能以组件组合的方式来建立。组件化程序设计方法强调软件的重用性和高度的互操作性,侧重于组件的定义、构建和装配,支持即插即用型应用。其重用性指所提供的功能能为多种系统使用,互操作指不同来源的组件能相互协调、通信,共同完成更复杂的功能。在架构和实施过程中,既要遵守 J2EE 技术标准和规范,如成熟的 JavaBean/EJB 等组件标准,还要将系统应用所需组件有机地部署到应用服务器的体系结构中。

利用分布式组件技术建设 B/S 结构应用系统灵活性较强,能够提高应用开发的重用性和开发效率,使应用系统各模块凝成为有机体。利用组件技术部署应用服务,可达到系统开发和应用集成要求。

本系统利用组件技术实现了数据库连接缓冲池组件、报表数据转换组件、数据通讯组件、事务处理组件、安全定义组件、错误处理组件、协调组件、界面操作组件、业务逻辑组件、报表生成组件和报表操作组件,以及 workflow 引擎处理组件等。

数据库连接池组件的运行机制是:当页面提交时,数据修改后,服务器必须连接到数据库,才能使作业得以完成。如果用户每次提交页面时,都必须执行连接或断开,性能和扩充性就会受到很大的影响。在一个事务周期中,一次数据库连接时开销相当大的操作。应用服务器为用户提供一个连接池,这些连接根据需为服务器的全部组件所共享。如果用户要修改一组数据,组件就请求服务器从池中获得一个连接,如果有连接可以使用,就将它分配给用户的请求,做全部修改。该组件还给出每件事是否成功的信号。当事务结束时,连接就立即释放。

### 3.3 工作流引擎

文献资料的采购申请和审批工作具有典型的工作流管理过程。本系统基于组件技术开发的工作流引擎实现了工作流实例的启动、任务的调度、分配与协调、工作流程的动态调整、工作历史的记录以及工作流事务的管理。

整个工作流引擎包括运行服务模块、活动管理模块、消息服务模块、事件管理模块、日志/历史管理模块、过程监控模块和工作列表管理模块等。

运行服务模块在对工作流定义进行实例化及执行的过程中需要读取定义库、组织机构库的信息,生成工作项写入数据库,并形成相应的运行日志及历史数据存入日志库及历史库。活动管理模块负责根据人员的负载情况进行活动的分配和调度,并且通过消息会话的方式通知被分配任务的相关人员。消息服务负责监听消息和发送消息。事件管理模块负责捕捉异常事件,启动异常处理。日志/历史记录模块负责保存与系统恢复有关的信息,例如工作流活动中所有的状态、事件和数据的变化信息,以便恢复崩溃的工作流实例状态。同时,记录工作执行的相关信息,例如谁在什么事件参与了什么角色执行什么任务等,为查询已结束的工作流信息提供支持。高级用户可以通过监控模块来查阅当前各工作流的执行状态,同时可以修改相应的信息,包括状态信息、审批流程等。

### 3.4 报表的实现

报表的生成机制包含访问数据源的连接机制、权限控制和页面生成等。报表工具是否易用关键在于报表数据项的定义,为了简化报表数据项的定义,系统将报表数据项分为动态数据、报表页面表达组件和数据源连接操作。动态数据是指根据用户定制 SQL 语句,访问数据库返回的动态数据,在报表生成时不需要任何处理;报表页面表达组件由各种报表页面参数,插入动态数据组的单元数据组成,自动绘出客户端报表页面。数据源连接操作指根据用户激活页面上“插入”、“修改”、“删除”和“查询”按钮后,用于操作数据库中的数据。下面举例说明查询数据库生成报表的方法:

首先执行 Select 语句进行查询数据:

```
DBAccess dbaccess = new DBAccessImp();
try{
    String sql = "SELECT * FROM literatures";
```

```
dbaccess.getConnection();  
ResultConnect resultInfo = dbaccess.Query(sql);  
//执行查询语句  
dbaccess.closeConnection();  
}  
catch(Exception e){}
```

然后将返回的动态数据赋给一数组,再由报表页面表达组件生成报表:

```
UniPaging unipage = new UniPagingImp();  
unipage.setMutilHeader(false); //不设表标题  
unipage.setPageTitles(new String[]{"序号", "资料名称", "作者", "时间", "卷册", "领域", "说明"});  
unipage.setCellAlign(new String[]{"center", "left", "left", "right", "left", "left", "left"});  
unipage.setRate(new int[] {5, 20, 15, 15, 10, 10, 25});  
unipage.setPageStyle(6); //表显示样式  
unipage.setPageRowNum(10); //每页显示数据条数  
out.println(unipage.parseString()); //输出报表
```

## 4 结语

基于 B/S 模式的文献资料管理系统,实现了企业科技资料管理人员网上自动化管理和企业科技资料应用人员的网上查询与办公,保证了信息流程中数据的一致性、实时性,减少了劳动强度,节约了办公费用,提高了工作效率,方便企业科技资料的查询和应用,使得文献资料管理工作完全流程化和自动化。

## 参考文献

- 1 Paul J. Perrone, Building Java Enterprise Systems with J2EE, 2001.
- 2 Sun Microsystems, Writing Enterprise Applications with Java 2 SDK Enterprise Edition, 2002.
- 3 李胜利、任军, 电力营销管理信息系统结构设计及实现技术, 电力系统自动化, 2003, 27(10): 73~76.
- 4 李胜利、任军等, 建设企业信息门户的技术与实现, 山东科技大学学报, 2003, Vol. 22, No. 2: 99~101.