DOI:10.16208/j.issn1000-7024.2007.21.033

基于J2EE架构的科研管理系统的设计与实现

刘泽轩， 江春华
(电子科技大学 软件学院，四川 成都 610054)

摘 要：J2EE作为新一代企业应用开发平台，具有显著的优势，正在被越来越多的企业所采用。对J2EE架构进行了详细的介绍，运用企业级应用的分层构建思想，完成了科研管理系统的设计与实现，同时对系统中的关键技术进行了阐述，为基于J2EE架构的应用开发积累了实际经验。

关键词：J2EE；EJB；MVC；科研管理系统；B/S
中图法分类号：TP311.52 **文献标识码**：A **文章编号**：1000-7024(2007)21-5218-03

Design and implementation of science research management system based on J2EE technology

LIU Ze-xuan, JIANG Chun-hua
(School of Software Technology, University of Electronic Science and Technology of China, Chengdu 610054, China)

Abstract: As the novel platform of application and development for enterprise, J2EE has many advantages. Now, it is adopted by more and more enterprises as their first project. The J2EE technology is introduced, the distributed construction idea of enterprise application is applied, the design and implementation of science research management system is completed, and the key technology of this system is detailed. As a whole, some experiences used for the development based on J2EE are accumulated.

Key words: J2EE; EJB; MVC; science research management system; B/S

0 引 言

随着Internet技术的迅速发展，建设数字化校园，在办公自动化的基础上实现办公网络化，已经成为高等院校实现管理现代化和信息化的关键。科学研究是高等学校的重要工作之一，很多高校学科种类多，科研项目及成果涉及科学领域广、数量大，这些因素都给管理造成了一定的困难，通过计算机进行管理无疑是最有效也是必须的手段。相对于传统的人工管理方式，使用计算机进行信息管理，大大提高了工作效率及安全性。尤其对于相对复杂的信息管理，人工管理起来可能会非常的费时费力，而且很容易出现错误，计算机则能够充分发挥它的优越性，使管理变得简单和轻松。因此，开发科研管理系统，实现高校科研信息管理数字化非常具有现实意义。本文介绍了如何设计与实现一个基于J2EE架构的科研管理系统，并对其中的关键技术进行了较为深入的探讨。

1 J2EE

J2EE事实上是使用Java技术开发企业级应用的一种工业标准，它是Java技术不断适应和促进企业级应用过程中的产物。Sun推出J2EE的目的是为应用Java技术开发服务器端应用提供一个平台独立的、可移植的、多用户的、安全的和基于标准的企业级平台，从而简化企业应用的开发、管理和部署。J2EE旨在为支持Java语言服务器端部署而提供与平台无关的、可移植的、多用户的、安全和标准的企业级平台。随着J2EE技术的不断成熟，现已逐渐成为企业Web应用开发的标准。其最终目的就是成为一个能够使企业开发者大幅度缩短投放市场时间的体系结构。

J2EE使用多层的分布式应用模型，应用逻辑按功能划分为组件，各个应用组件根据他们所在的层分布在不同的机器上。J2EE典型的4层结构如图1所示。

(1)客户层：主要是由HTML用户、Java Applets等GUI和

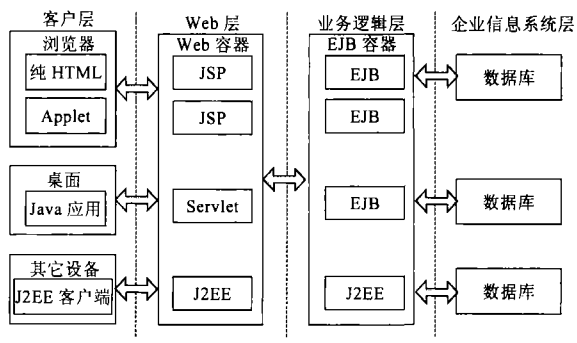

图1 J2EE的4层结构

收稿日期：2006-11-12 **E-mail**：sean_lzx@163.com
作者简介：刘泽轩(1982－)，男，北京人，硕士研究生，研究方向为软件开发方法；江春华(1962－)，男，湖北仙桃人，副教授，研究方向为中间件技术和网络信息处理。

Java 应用组成，用户通过 GUI 与应用程序交互。

(2)Web 层：主要由 Web 容器的 JSP 和 Servelet 实现。主要作用是接受来自表示客户端的用户反馈，并根据接收客户端请求，对用户的请求产生相应的回应。

(3)业务逻辑层：本层负责处理应用的核心业务逻辑，执行复杂的业务逻辑，主要由 3 种 EJB 来处理，分别是：会话 Bean、实体 Bean、和消息驱动 Bean。其中会话 Bean 是商务过程对象，主要执行商务逻辑、商务规则和工作流程等；实体 Bean 能够永久性存储数据的持久对象，它包含了核心商务数据，实体 Bean 的实例是一个对应到数据库中的视图；消息驱动 Bean 是能够接收 JMS 消息的特殊 EJB 组件，允许一个业务层组件异步接收 JMS 消息。

(4)企业信息系统层：企业信息系统层也称为数据层，它是驻留业务数据的地方，这一层为企业的信息系统服务，包括数据库系统、事务处理系统、遗留系统和企业资源规划(ERP)系统等。企业信息系统层是 J2EE 应用与非 J2EE 应用或遗留系统集成的连接点。

2 系统的总体设计

2.1 系统的体系结构

本系统采用 MVC(模型-视图-控制)设计模式，将输入、处理、输出流程按照模型、视图、控制的方式进行分离，这样系统被分成模型层、视图层、控制层 3 个层。模型层进行业务流程/状态的处理以及业务规则的制定。视图层代表用户交互界面。控制层则从用户接收请求，将模型与视图匹配在一起，共同完成用户的请求。在本系统体系结构的设计中，我们将 J2EE 架构和 MVC 模式设计结合在一起，MVC 设计模式具体对应到 J2EE 架构如下：JSP 对应视图层，因为整个应用系统主要通过 JSP 来与外界进行交互；Servlet 对应控制层，作为 JSP 与 EJB 之间的中间枢纽；EJB 对应模型层，主要进行数据业务的处理。

为了提高开发的效率和软件的可重用性，在决定了系统采用基于 J2EE 架构和 MVC 模式的多层体系结构实现的基础上，对于各层组件选用与协作的实现，我们采用了 JSP+Servlet+EJB 的架构模式。系统体系结构图如图 2 所示。

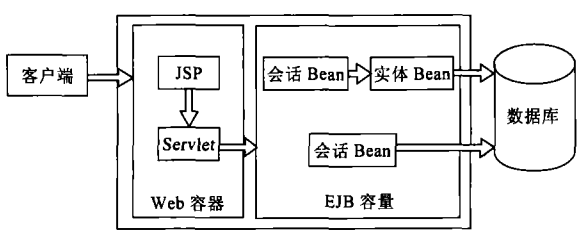

图 2 系统体系结构

(1)Web 容器：所有用户界面都要在这里得到。我们采用 JSP 实现用户界面，Servlet 实现控制。用户页面主要分为 3 大类：验证界面、操作界面、信息反馈界面。用户通过 JSP 实现的验证界面调用同样是 JSP 实现的操作界面，选择相应的操作项即在 JSP 页面中调用实现控制的 Servlet，再由 Servlet 与 EJB 容器通信，实现对其中的 EJB 的调用，调用反馈的结果通过 JSP 实现的信息反馈界面展现给用户。

(2)EJB 容器：事务逻辑和规则化在此实现。这一层主要由 EJB 组件组成，包括对数据库访问的实体 Bean，以及调用实体 Bean 的会话 Bean。在 EJB 之间的关系中，会话 Bean 实际上是实体 Bean 的客户端。在应用程序的末端，实体 Bean 通过访问数据库中的表存储实体的状态。各个业务逻辑的具体实现都是通过会话 Bean 和实体 Bean 间的协作实现的。

(3)访问数据库：EJB 容器中的实体 Bean 是通过数据库连接和数据库管理系统交互信息的，对于不同的数据库系统，其访问形式是不同的；本系统对数据库访问相对比较灵活，部分通过实体 Bean 访问，部分通过会话 Bean 访问，但访问的实质是通过标准的 JDBC 访问数据库，这样可以使系统具有很强的可扩展性。

2.2 系统的功能模块划分

本系统主要实现对科研项目的计划管理和质量管理，同时也对其它如成果、专利、论文、论著、科研人员等信息进行维护。要求系统提供数据录入、更新和删除的界面，且操作尽可能简单明了，此外要求可以生成一些统计信息报表并提供打印报表的功能。为了预防系统突发情况或人为的误操作导致的数据毁坏，系统还要有进行数据备份和恢复的功能，而且必须提供导出接口，可将数据库信息输出成别的数据库格式，提高兼容性，方便数据信息的传输。

通过对需求的详细分析，按照功能的不同，我们将本系统划分为项目计划管理模块、项目质量管理模块、科研人员管理模块、成果管理模块、专利管理模块、论文论著管理模块、系统管理模块这 7 个模块，有些模块的数据之间存在联系制约的关系。各部分都提供了数据维护、查询和报表的基本功能。系统功能模块如图 3 所示。

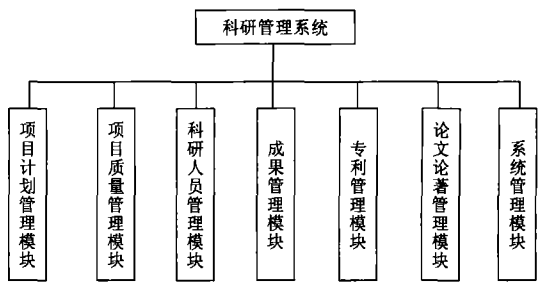

图 3 系统功能模块

3 系统的详细设计与实现

3.1 Web 层

在本系统中，Web 层实际包含了 MVC 模式中的视图层和控制层这两个层次结构。视图层我们使用 JSP 实现，控制层使用 Servlet 实现。

在 Web 层的设计中，必须特别注意将视图层和控制层严格分离。在一些设计得并不理想的系统中，我们常常可以看到，在 JSP 页面代码中会夹杂着大量对业务逻辑处理以及对数据库进行访问的代码。这样做会因为代码相对集中使开发人员更加方便地对代码进行处理，但会使系统各层之间的耦合度大大增加，使日后的维护变得非常困难，系统的可扩展性也会大大降低。因此，我们必须严格按照 MVC 模式设计系统，将代码分离开：在 JSP 页面里只出现必要的动态控制代码，页面中所有需要的数据都由 Servlet 传递，所有用户提交的数

据等都是通过表单提交给 Servlet 处理。访问合法性判断、页面跳转、响应用户请求等工作，都是由 Servlet 完成。例如，在修改科研人员信息页面中，首先由相应的 Servlet 通过 setAttribute()方法将所需数据设置为 session 的属性；然后 JSP 页面通过getAttribute()方法在将数据读出后删除该属性，并将数据显示在页面相应的位置；当用户修改信息完成后提交时，所有需要保存的数据通过表单提交给相应的 Servlet，由 Servlet 进行后续处理。这样做的好处是使视图层与控制层完全分离开，使系统无论是在安全性上或是在可扩展性上都非常好。

3.2 业务逻辑层

本层在MVC模式中即为模型层，也是整个系统中最为重要的一层。业务逻辑层负责处理应用程序核心的业务逻辑，系统中的高层用例就是由业务层组件来实现的，同时业务逻辑层还为 Web 层组件提供必要的接口服务。

前文中我们已经分析了整个系统的体系结构以及各个层之间的工作方式，在业务逻辑层中，会话 Bean 负责处理系统的业务逻辑，实体 Bean 保持数据的持久性。系统工作时，Web 组件首先需要调用会话 Bean，而会话 Bean 再调用一个或多个实体 Bean。这样设计的好处是会话 Bean 只把业务接口暴露给 Web 层，而实际的数据持久层采用一个或多个实体 Bean 来实现。因为数据持久层隐藏在抽象层次之后，所以实现了 Web 层与数据持久层的隔离。实体 Bean 代表持久对象，它与数据库中的表相映射，生成数据库的对象视图。实体 Bean 的持久性可以由容器管理或由 Bean 管理。在本系统中，我们采用了由容器管理持久性(CMP)的实体 Bean。这样设计的优点是由容器负责同步 Bean 状态与基础数据库。当客户修改 Bean 的状态时，容器自动提供所需的 SQL 语句负责保证使用 CMP Bean 时数据的一致性和完整性，因此使用 CMP 可以极大地减少程序编码。当我们创建好一个 CMP Bean 后，我们进行 Bean 与数据库的映射。由于实体 Bean 的持久性是由容器管理的，所以我们在查询数据库时使用 EJB 查询语言(EJB-QL)。EJB-QL 是标准的、可移植的、用于表达容器管理持久化实体 Bean 查询操作的语言。EJB-QL 不能单独使用，必须定义在 EJB 部署描述符的<ejb-ql>标记中。下面给出一个 CMP Bean 中 findAllUsers 方法在部署描述符中的代码，它实现了从用户表中查询所有数据项的功能：

```
<query>
    <description></description>
    <query-method>
        <method-name>findAllUsers</method-name>
        <method-params>
        </method-params>
    </query-method>
    <ejb-ql>select object(o) from User o</ejb-ql>
</query>
```

在本系统的设计中，因为 JSP 或是 Servlet 与 EJB 有可能不在同一虚拟机上，所以会话 Bean 采用了远程接口。而实体 Bean 不由 Web 组件或者其它客户层组件直接调用，所以实体 Bean 的接口都设计成了本地接口，供 EJB 容器内部的会话 Bean 直接调用。

3.3 数据层

数据库的设计由两个组成部分：逻辑设计和物理设计。逻辑设计主要形成独立于机器特点、独立于各个DBMS产品的抽象实体，以及各实体之间的关系和约束。物理设计则是根据DBMS特点和处理的需要，进行物理存储安排，设计索引，形成数据库内模式。在本系统的实际设计过程中，我们首先根据具体需求得出系统的实体-关系图，然后对应实体-关系图得出具体数据库中表的设计。将实体-关系图中的实体对象映射为表，而其属性则映射为表中的列。在众多表中，我们采用主键和外键映射它们之间的引用关系。由此我们就得出了系统数据库的物理模型。在本系统数据库的设计中，我们采用符合第三范式标准和非规范化相结合的方式处理各表，根据具体的情况选用适当的规范标准。这样，整个数据库设计允许有适当的冗余，既提高了数据库的运行效率，又使其数据操作不失灵活性。

4 关键技术举例

4.1 访问合法性判断

在整个系统的设计中，安全是非常重要的因素之一。根据需求，系统不允许匿名登录，并且必须能够允许不同用户以不同身份登录，因此访问合法性的判断是非常重要的。访问合法性的判断是在 Servlet 中进行的，下面分析本系统中对访问合法性判断的思路：首先当用户登录时，如果登录成功则为其创建一个 session，然后将从数据库中读出的用户相关信息例如用户名、用户类型等存在一个用户信息类UserVO的实例 userInfo 中，并通过 setAttribute() 方法将其设为 session 的一个属性，在用户退出系统前该属性一直伴随 session 存在。当用户进行操作时，首先判断该用户的session是否为空，如果为空则判断为非法访问。如果用户的 session 存在，则通过 getAttribute() 方法读取 userInfo 属性。如果 userInfo 为空或 userInfo 中的用户类型与当前操作的用户类型不符，则判断为非法访问。通过上述对访问合法性的判断，我们可以完全避免各种非法访问情况的发生，大大增强系统的安全性。下面给出一段 Servlet 中访问合法性判断代码片断作为示例：

```
public void doGet (HttpServletRequest req,  HttpServletResponse resp) throws ServletException, IOException {
HttpSession session = req.getSession(false); // 如果 session 不存在则不创建
if (session == null) {//如果 session 为空，则访问非法，直接跳转到登录页面
 resp.sendRedirect("login.jsp");
}
UserVO userInfo = (UserVO) session.getAttribute("user-Inf-o");
//从 session 中获取 userInfo 属性
if (userInfo ==null &&! userInfo.getUserClass().equals("01")) {
//如果userInfo属性不存在或者用户类型不是01，则访问非法，直接跳转到登录页面
        resp.sendRedirect("login.jsp");
    }
    …
}
```

(下转第 5247 页)

才能对日志进行查询、删除和修改。

4 结束语

随着刀片PC的逐步推广，它本身的管理功能也在逐步完善。性能管理在刀片PC管理域中占具很重要的地位，用户对设备本身的使用情况提高了要求，需要从性能方面满足用户的需求，并能给其它管理域一个分配使用资源的依据，因此，本文通过建立一个基于SNMP的刀片PC性能管理模型，阐述了性能管理在刀片PC管理域中的地位和作用，成为故障管理和安全管理有利的保障条件。

参考文献:

[1] 杨家海,任宪坤.网络管理原理与实现技术[M].北京:清华大学出版社,2000.

[2] Richard Stevens W. TCP/IP详解卷1:协议[M].范建华,译.北京:机械工业出版社,2000.

[3] 李学渊.基于SNMP网络性能管理的研究[J].计算机与数字工程,2005,33(4):11-15.

[4] Aldbusser S. Application performance measurement MIB Draft-ietf-Monmib-apm-mib-08[S].IETF,2003-03-01.

[5] 王平,赵宏,陈海涛,等.一个基于SNMP的简单网络管理系统的设计与实现[J].小型微型计算机系统,2001,22(9):1047-1050.

[6] Java management extensions instrumentation and Agent specificationv1.2[S].Sun Microsystems Inc, 2002.

[7] 李木金,王光兴.一种被用于网络管理的性能分析模型和实现[J].软件学报,2000,11(2):251-255.

[8] Carlton R Davis. PSec:VPN的安全实施[M]. 周永彬,译. 北京:清华大学出版社,2002.

[9] 杨秀梅,吕卫锋.CIM的网络性能管理[J].计算机工程与设计,2006,27(8):1425-1428.

(上接第5220页)

4.2 值对象

在系统中，系统的客户端浏览器必然要与EJB交换数据。我们将会话Bean和实体Bean作为服务器端业务组件，而业务组件的一些方法可以向客户端返回数据，对业务服务对象的每个方法调用一般都是远程的。因此，在系统中需要远程调用使用网络层服务，而不关心客户端与Bean的直接交互。客户端需要调用业务对象的一次get方法以获得一个属性，调用业务对象的一次 set 方法以设置一个属性。如果要获得或设置业务对象的所有属性，则需要调用很多次get或set方法，而get、set方法会导致调用大大增加。随着网络远程方法使用的逐渐增加，将耗费大量系统资源，使应用程序的性能下降得很厉害，从而导致整个系统的效率减低，可用性不高。

因此，在本系统的设计中，大量使用了值对象。对需要在客户端与Bean之间传递的多个数据属性进行封装，这样客户端就可以把需要向 Bean 传递的多个属性值先填充到值对象类中再通过值对象传给Bean；反之Bean也可以此方式向客户端传递数据，从而实现一次远程方法调用多个属性，大大降低了系统网络开销。由于要求值对象类能在分布式系统的客户端和服务器端间进行传递，所以它必须是可串行序列化的Java对象。下面给出系统中的值对象类PaperVO的代码作为示例。

```
public class PaperVO implements Serializable {
private String PaperID = null;        //论文编号
private String PaperTitle = null;     //论文标题
public PaperVO () {
    super();
}
public String getPaperID() { //读取属性论文编号方法
    return PaperID;
}
public String getPaperTitle() { //读取属性论文标题方法
    return PaperTitle;
}
public void setPaperID(String string) { //设置属性论文编号方法
    PaperID = string;
}
public void setPaperTitle(String string) { //设置属性论文标题方法
    PaperTitle = string;
}
}
```

5 结束语

基于J2EE架构开发的科研管理系统，具有高性能、高可扩展性和高安全性的特点。由于采用了MVC设计模式，简化了系统的开发、管理和维护，提高了系统的开发效率，体现了J2EE架构的优势。本文阐述了如何设计与实现基于J2EE架构的科研管理系统。本系统经过近一年的试运行，取得了非常好的效果，达到了预期的设计目标，是J2EE应用开发的一个成功案例。

参考文献:

[1] Subrahmanyam Allamaraju,Karl Aveda.J2EE服务器端高级编程[M].北京:机械工业出版社,2001.

[2] 李红,曹永宁.基于J2EE的企业信息平台的设计与实现[J].计算机工程,2003,26(2):204-207.

[3] Ed Roman,Rima Patel Sriganesh,Gerald Brose.精通EJB[M].3版.北京:电子工业出版社,2005.

[4] Floyd Marinescu. EJB设计模式 [M]. 北京: 机械工业出版社, 2004.

[5] Joseph J Bambara,Paul R Allen.J2EE技术内幕[M].北京:机械工业出版社,2002.

[6] Pravin V Tulachan.EJB2.0组件开发指南[M].北京:清华大学出版社,2002.

[7] Kathy Sierra,Bert Bates.深入浅出EJB[M].南京:东南大学出版社,2006.

[8] IBM.WebSphere MQ manuals[EB/OL].www.ibm.com/software/mqseries/library/manualsa.