

基于轻量级 J2EE 平台的政府采购系统的设计与实现

朱少敏¹, 刘建明²

(1. 中国电力科学研究院, 北京 100085; 2. 国电信息中心, 北京 100761)

摘 要: 目前, J2EE 体系结构已经成为开发电子政务系统的主流平台。剖析了 J2EE 重量级与轻量级开发平台的优缺点, 对目前流行的开源框架 WebWork、Spring、Hibernate 进行分析, 结合政府电子采购系统的实际特点, 采用 J2EE 标准三层体系结构, 设计基于 WebWork+Spring+Hibernate 的 J2EE 轻量级开发平台。通过政府电子采购系统中具体实例的实现, 详细说明如何应用此种轻量级框架构建 J2EE。

关键词: 电子政务; 政府电子采购系统; 开源框架; 三层体系结构; 轻量级 J2EE 平台

中图分类号: TP315 **文献标识码:** A **文章编号:** 1000-7024 (2007) 24-5945-04

Design and realization of government procurement system based on lightweight J2EE framework

ZHU Shao-min¹, LIU Jian-ming²

(1. China Electric Power Research Institute, Beijing 100085, China;

2. China Electric Power Information Center, Beijing 100761, China)

Abstract: J2EE framework is becoming prevalent developing platform in electronic government system platform. By analyzing J2EE development platform advantages and disadvantages between heavyweight with lightweight and analyzing prevalent open resource frameworks such as WebWork, spring, hibernate, combining with government electronic procurement system characteristics, a lightweight J2EE framework is designed based on WebWork+spring+hibernate using standard three-tier J2EE system architecture. Using an explicit government electronic procurement system example, how to explore J2EE application is detailed illuminated.

Key words: electronic government; government electronic procurement system; open resource frameworks; three-tier system architecture; lightweight J2EE platform

0 引言

政府电子采购系统是政府电子政务建设的一个非常显著应用, 是“金财工程”的重要组成部分。目的是利用先进的信息技术通过网络化实现在线的跨部门、跨地域的政府采购, 营造一个低成本、高效率、统一的、互动的电子采购平台。实现政府采购信息化、运作规范化、运营实体化。为政府采购信息化建设探索了一种企业参与、政府监管、社会监督、公共使用的新模式。

J2EE(Java 2 platform enterprise edition)体系结构为构建具有安全性高、易扩展性、可维护性强的电子政务系统提供了良好的机制, 已经成为开发电子政务系统的主流平台。目前以 EJB (enterprise JavaBean) 等技术为核心的重量级 J2EE 系统开发平台已经显示出其在电子政务开发方面的局限和不足^[1]。轻量级 J2EE 平台的出现使得人们有了新的解决方案, 日益得到广泛关注和应用。

1 J2EE 平台的比较分析

1.1 重量级开发平台的问题

以往在构建 J2EE 应用的时候, 首先会选用复杂的 EJB 解决方案。这种重量级平台在应用中存在着以下不足:

(1)EJB 应用服务器价格昂贵: EJB 的使用要求有支持 EJB 的应用服务器支持。目前常用的有 IBM 的 WebSphere 和 BEA 公司的 WebLogic 等, 但是它们的应用服务器系统价格也比较高; 同时使用不同的应用服务器, 应用服务器的部署描述需要重新书写, 这使得在不同应用服务器下移植企业级应用会比较麻烦。

(2)EJB 使用过于复杂: EJB 是一种功能齐全、复杂性很高的重量级企业级组件, 开发人员很难在短时间内掌握精通。如果水平不够的开发人员采用 EJB 技术, 开发的系统性能将得不到保证。

(3)EJB 自身性能问题: 由于 EJB 框架考虑了大量的系统

收稿日期: 2006-12-19 E-mail: zhushaomin8888@sina.com

作者简介: 朱少敏 (1983-), 男, 安徽庐江人, 博士研究生, 研究方向为网络与信息安全; 刘建明 (1955-), 男, 山东荣成人, 教授级高级工程师, 博士生导师, 研究方向为网络与信息安全、流媒体技术。

事务、安全等方面问题,需要创建和管理多个对象实例和相关资源,消耗了服务器大量处理能力和内存。因而 EJB 提供强大功能同时也付出了效率的代价。

(4)EJB 测试复杂:软件项目的开发需要进行严格的测试。但由于 EJB 的内聚性较强,比如数据持久层管理、事务管理、生命周期管理全部都是由 EJB 容器管理,并且有要求苛刻的 API,使得开发者必须依赖于 EJB 容器进行开发。EJB 使用的复杂性使得测试也变得相对复杂和困难。

1.2 轻量级开发平台的优势

所谓的轻量级并不是功能弱、设计简陋、实现粗糙,主要是和以 EJB 为核心技术的复杂框架对比而言,致力于提供最简单的组件来构筑 Web 应用系统。主要有以下几个优点:

(1)开发层次结构清晰、开发简单化:轻量级架构提供简单化编程。软件的开发、测试、发布、维护等环节更加容易。同时框架接口实现部分可以根据情况配置,实现不同层次之间相对独立,层次结构清晰。

(2)面向对象的空间增大:EJB 通常不是真正的对象,因为它受限于 EJB 组件模型特性。例如^[2],EJB 必须是相当粗粒度的,即使本地 EJB 也是如此。这限制了正规的面向对象的设计。但是轻量级容器强加给对象的限制要少的多,所以通常它们可以获得更大的面向对象空间。

(3)测试的简单:无论是单元测试或是集成测试,轻量级框架非常容易进行。轻量级框架各个层次的组件藕合性小,利于测试工作。一方面开发人员可以在 JUnit 测试类中轻松导入和使用所有被管理的 Bean,保证测试类和程序可以进行正常交互。另一方面测试时不用绑定它们之间的依赖关系,软件测试因此变得简单。

2 轻量级 J2EE 开发平台设计

2.1 基于 MVC 设计模式的 WebWork 框架

WebWork 是由 OpenSymphony 组织开发,致力于组件化和代码重用的 MVC 模式 J2EE Web 框架。

WebWork 通过一个 Dispatcher 对象把客户端发送的 URL 请求映射到一个命令对象(由 Xwork 里的 Action 对象实现)中。该命令对象在后台完成与系统内部模型交互,并根据处理后的返回结果把控制权映射到某一具体视图(通常用 JSP 表示)^[3]。最后视图层根据表达式语言 OGNL 对更新后结果进行显示。

在表示层设计中,传统的方法是采用 Struts 框架进行设计。相对于 Struts 框架,WebWork 主要有以下优点:

(1)在线程模型安全方面:Struts 仅允许一个 Action 去处理所有的请求,所以需要同步 Action 中涉及的所有资源,这就存在线程安全的问题。而在 WebWork 中每个请求单独对应一个 Action,因此没有线程的安全问题。

(2)在对 Action 的执行控制方面:控制 Struts 创建的 Action 的执行顺序是不容易实现的。但是 WebWork 可以用拦截器实现对 Action 执行顺序的控制。

(3)在页面的组成方面:WebWork 支持表达式语言,页面可以不含有繁琐的 request.getParameter()方法和大量复杂的标签。

2.2 基于 O/R 映射的 Hibernate 框架

当前,Hibernate 架构已成为最为流行的对象/关系映射解

决方案。它是一个基于 Java 的开放源代码的持久化中间件。不仅提供 ORM 映射服务,还提供数据查询和数据缓存功能^[4]。Java 开发人员可以方便地通过 Hibernate API 来操纵数据库。

Hibernate 采用运行生成机制,在系统启动时生成 SQL 语句,对象实施持久化管理。应用程序通过 Hibernate 与数据库发生关系,完成对数据进行操作。Hibernate 自身通过属性和类映射文件将类映射到数据库。因此 Web 应用可以通过持久化的对象类直接访问数据库,而不必使用 JDBC 和 SQL 语句进行数据的操作。当系统底层数据库进行移植时,只需简单地修改配置文件 Hibernate 属性即可,而不再修改代码和映射文件。此外,Hibernate 专门提供一个对象查询语言 (HQL) 来查找和操纵对象,由 Hibernate 进行统一的管理和优化,方便程序员对数据持久层的访问。另外,Hibernate 还通过底层管理的 JNDI、JDBC、JTA 对外提供事务接口,满足用户事务管理需求。

2.3 轻量级的 Spring 框架

Spring 框架是轻量级的 J2EE 应用框架,核心基础是一个轻量级的容器。它提供了 Bean 的配置基础、面向方面编程 (aspect oriented programming, AOP) 支持、控制反转 (inversion of control, IoC) 模式等功能^[5]。Spring 的目标是实现一个全方位、功能完备的整合框架。在此框架下实现多个子框架的灵活组合,这些子框架既可以彼此独立,也可以切换成其它框架。Spring 希望实现的目标是一站式服务的框架整合方案。

IoC 是面向对象领域新兴的编程思想,也是 Spring 的精髓所在,就是指由容器来控制程序之间的关系,而不是传统实现中由程序代码进行操控。其本质思想就是将调用者与被调用者进行分离,最大程度上降低彼此之间的藕合度。IoC 将控制创建的任务放入框架中,从而实现与应用程序的分开。当使用 Spring 的 IoC 容器时只需指出组件需要的对象,在运行时 Spring 的 IoC 容器会根据 xml 配置数据提供给它^[3]。

2.4 轻量级 J2EE 开发平台的构建

从对各个框架的分析可知,WebWork 架构实现表示层设计,并对表示层进行 MVC 划分,但是缺乏对模型的支持。Hibernate 提供模型部分中持久化数据与关系数据库的映射,但是还是比较紧耦合实际的业务逻辑^[3]。综合分析 WebWork、Hibernate、Spring 这 3 个开源框架性能和特点后,政府电子采购系统所采用的轻量级 J2EE 开发平台以 Spring 框架为核心,WebWork 框架应用于表示层,数据持久层选择 Hibernate 技术。整个 J2EE 轻量级开发平台采用 WebWork+Spring+Hibernate 的开发架构,如图 1 所示。

3 轻量级 J2EE 平台在政府电子采购系统中的实现

3.1 功能说明

政府电子采购系统由外网 Internet 应用和内网政务专网两部分组成。以内网政务专网中的一个子系统政府电子采购计划管理系统的查询用款计划功能的实现为例,详细说明如何利用 Spring 框架整合 WebWork 和 Hibernate 构建轻量级 J2EE 平台。采购人查询采购用款计划将实现一个简单的采购用款计划管理功能。包括采购人录入、查询、查看、删除采购用款计划等操作。

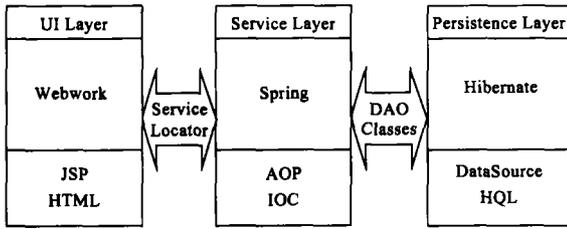


图1 轻量级J2EE开发平台体系结构

3.2 表示层

WebWork 框架应用于表示层,提供了简单灵活的MVC模型。主要功能^[7]:①接收和处理用户的请求,并将请求信息发送到业务逻辑层进行处理;②将业务逻辑层处理的结果传递到相应的视图并显示给用户;③处理应用程序的异常。

首先利用 WebWork 标签和 Javascript 技术构建表示层的录入采购计划页面 stockProjectInput.jsp,代码如下:

```
<ww:form name = "inputStockProject" >
.....
<td width = "30%">采购计划录入</td>
<td width = "30%"><table>
<ww:textfield name = "stockProjectModel.proName"/></table>
</td>
.....
<input type = "button" value = "保存" onclick = "javascript:go-Submit();">
.....
```

当用户提交用款计划的表单时,其中包含一个对象:input-StockProject.action。用户提交表单后,系统会根据 WebWork 的 xwork.xml 找到此 action 的名称和对应实现类。

```
<xwork>
<action name = "inputStockProject" class = "stockProjectAction">
<result name = "success" type = "dispatcher">/WEB-INF/stock-ProjectEdit.jsp</result>
</action>
</xwork>
```

3.3 业务逻辑层

业务逻辑层位于整个轻量级结构的中间部分。忽视了这一层的作用,将导致了程序代码之间的耦合度变大。Spring 作为轻量级的核心部件,主要负责完成以下任务^[7]:

- (1)对程序进行业务校验和业务逻辑处理;
- (2)设计与其它各层之间相交互的接口;
- (3)将程序开发涉及的业务逻辑和对象通过xml联系起来。

3.3.1 Spring 与 WebWork 的整合

轻量级 Spring 框架可以很好与表示层的 WebWork 框架进行整合。在对 WebWork 配置的时候只要添加 Spring 的配置,就可以在使用 WebWork 的同时对 Spring 进行初始化。将 WebWork 对动作的管理转交给 Spring 框架来处理。具体来说就是通过 Spring 设置的 xml 文件(如下所示)来对 action 添加相应的业务处理对象的注入。

```
<beans>
<bean id = "stockProjectAction" class = "StockProjectAction"
```

```
parent = "baseAction" singleton = "false">
<property name = "stockProjectService">
<ref bean = "stockProjectService"/>
</property>
.....
</beans>
```

在 action 对应实现类中,只要声明业务逻辑类的读属性和写属性,就可以调用相关业务逻辑^[7]。

```
public class StockProjectAction {
StockProjectService stockProjectService; //业务逻辑对象
public StockProjectService getStockProjectService() {
return stockProjectService; }
public void setStockProjectService(StockProjectService stock-ProjectService) {
this.stockProjectService = stockProjectService;}
public String inputStockProject () throws Exception{
return SUCCESS; }
.....
}
```

3.3.2 Spring 与 Hibernate 的整合

Spring 与 Hibernate 整合需利用业务逻辑类 StockProject-Service 去调用数据访问对象去访问数据库。由于使用了反转控制机制,业务逻辑类不需要直接生成数据访问类,只要声明数据访问类的属性即可,由 Spring 框架通过配置 xml 文件将数据实现类的具体实现付给其属性,有效地降低了数据访问层与业务逻辑层的耦合度。业务逻辑类的实现是通过接口来完成的。这样做的好处是当业务逻辑发生变化时如业务逻辑增加或者减少,需要改动的代码量就少。

业务逻辑类接口:

```
public interface StockProjectService extends Service {
//录入采购计划
public void inputStockProject (StockProjectModel stockPro-jectModel);
.....
}
```

业务逻辑类的具体实现:

```
public class StockProjectServiceRealization implements Stock-ProjectService {
//数据访问对象
StockProjectDAO stockProjectDao;
public StockProjectDAO getStockProjectDao() {
return stockProjectDao;}
public void setStockProjectDao (StockProjectDAO stockPro-jectDao) {
this.stockProjectDao = stockProjectDao;}
//计划录入功能
public void inputStockProject (StockProjectModel stockPro-jectModel){
stockProjectDao.inputStockProject (stockProjectModel);}
.....
}
```

```
}
```

Spring 框架在业务逻辑层中主要实现业务逻辑的处理,同时通过配置文件进行相关的事务管理。

```

<beans>
//业务逻辑对象
<bean class = " StockProjectServiceRealization">
bean id = "stockProjectService" parent = "txProxyTemplate">
<property name = "target">
    <property name = "stockProjectDao">
    <ref bean = "stockProjectDAO"/>
    </property>
</property>
</bean>
//声明式事务管理
<bean id = "stockProjectDAO" class = "StockProjectDAORea-
lization" parent = "transactionManager"/>
</beans>

```

3.4 数据访问层

数据访问层位于 J2EE 平台的末端。通常是程序最不容易控制的地方,并占用大量的调试时间。在数据访问层中使用 Hibernate 框架实现对象和关系之间的映射,完成普通对象与数据库的操作之间的相互转换。首先使用普通 Java 对象 StockProjectModel 模型进行持久化。属性通过 getter 和 setter 方法访问,对外隐藏了具体内部实现的细节。

```

public class StockProjectModel {
//采购单位名称
private String stockprojectname;
.....
public String getName() {
    return stockprojectname;}
public void setName(String stockprojectname) {
    this.stockprojectname = stockprojectname;}
}
.....
}

```

其次, Hibernate 通过 xml 文件来建立对象和关系数据库的映射。映射文件包含了对象/关系所需要的元数据,元数据包含持久化类的声明和属性到数据库的映射。需要注意的是各种关联,例如一对一关联、一对多关联、多对一关联等^[4]。

```

<hibernate-mapping >
<class name = "StockProjectModel" >
<property name = " stockprojectname" />
.....
</class>
</hibernate-mapping>

```

最后为了使得业务逻辑层与数据访问层之间的耦合度进一步减小, StockProjectServiceRealization 使用数据访问接口 StockProjectDAO 而不是具体的类。

```

数据访问接口:
public interface StockProjectDAO extends DAO {

```

```

public void inputStockProject (StockProjectModel stockPro-
jectModel);
.....
}
数据访问类的具体实现:
public class StockProjectDAORealization extends DAOHib-
ernate implements StockProjectDAO{
    public void inputStockProject (StockProjectModel stock-
ProjectModel){
        this.doCreateObject(stockProjectModel);}
.....
}

```

在政府电子采购系统中采购单位、采购项目、采购资金等都是容易发生变化的。为了保证在系统变化时,尽量减少代码的修改量,数据访问类需要使用接口的方式。但是在不易发生变化的系统中就没有必要一定使用接口的方式,这样会使得程序过于复杂。是否使用接口方式应该根据具体开发的系统特点来进行判定。

4 结束语

通过对政府采购过程的全面了解和析,针对政府采购过程中的采购计划申报、招标管理、过程监管、文档备案等提出的信息化应用解决方案的实际特点,提出系统开发的轻量级 J2EE 平台设计与实现方法。采用简单灵活的 MVC 设计框架 WebWork 作为表示层;选用 Spring 轻量级框架代替 EJB 模式,作为业务逻辑层处理的核心部件;选择目前最流行的对象/关系映射方案 Hibernate 构造数据访问层。构建了一个安全可靠、易于扩展、开放性、可维护性好的政府电子采购平台^[9]。不仅为政府电子采购系统的顺利实施提供了有力的保障,同时也对其它电子政务系统的设计与开发提供了有益的借鉴。

参考文献:

- [1] 黄达安,何明昕,黄战.基于轻量级 J2EE 框架电子政务系统的设计与实现[J].计算机工程与设计,2006,27(2):228-231.
- [2] 张进坤.轻量级容器中开发 J2EE web 应用的研究[D].四川大学,2005:8-9.
- [3] 丁杰,李治柱,陆舟.利用 Spring 框架进行 WebWork 与 Hibernate 的整合与开发[J].微型电脑应用,2006(2):30-36.
- [4] LightBody P,Carreira J.Webwork in action[Z].Manning Publications Co,2005.
- [5] 孙卫琴.精通 Hibernate:Java 对象持久化技术详解[M].北京:电子工业出版社,2005.
- [6] 郭圣伟,金炜东.基于 Spring 轻量级 Web 框架的统考系统的设计[J].科技资讯,2006(2):46-47.
- [7] 牟军,吕立.使用轻量级框架进行 J2EE 应用开发[J].小型微型计算机系统,2006(6):1149-1152.
- [8] 黄华.框架技术在 Web 系统开发中的应用[J].微机发展.2005(5):77-80.
- [9] 刘军,戴金山.基于 SpringMVC 与 iBATIS 的轻量级 Web 应用研究[J].计算机应用,2006,26(4):840-843.