

文章编号: 1005-8451 (2011) 04-0031-03

# 基于轻量级 Java EE 架构的教师考核系统的设计与实现

宋成明, 李彦军

(山西大学商务学院 信息工程学院, 太原 030031)

**摘要:** 结合院校信息化建设的实践和需求, 引入轻量级 Java EE 架构体系, 可以快速构建一个多层的可复用、可扩展、实用的教师考核系统。详细阐述开源框架 Struts2、Hibernate、Spring 的相关技术, 以及如何基于开源框架来简化系统开发的复杂性, 提高开发效率, 提出了系统的解决方案。并把这种基于 S2SH 的轻量级 Java EE 架构, 在教师考核系统中得到了实现。

**关键词:** 轻量级; Struts2 框架; Spring 框架; Hibernate 框架; 教师考核系统

中图分类号: TP311.12 文献标识码: A

## Design and implementation of Teacher Examination System based on light-weight

### Java EE framework

SONG Cheng-ming, LI Yan-jun

(School of Information Engineering, Business College of Shanxi University, Taiyuan 030031, China)

**Abstract:** Combined with the practice and needs for the informationization construction in colleges, used the light-weight Java EE framework. The Teacher Examination System could be quickly built which was reusable and multi-layered, scalable and practical. It was discussed the related technology of Struts2, Hibernate, and Spring, elaborated that how to use the open-source framework to simplify the complex system development, improved the efficiency of systems development proposed solutions. S2SH light-weight Java EE-based framework for assessment in Teacher Examination System was implemented.

**Key words:** lightweight degree; Struts 2 framework; Spring framework; Hibernate framework; Teacher Examination System

教师考核是各大院校管理工作的重要组成部分。现有考核方法的缺点是<sup>[1]</sup>, 教师必须回忆本学年在教学、科研等各方面工作的完成情况, 手工填写各项内容。并计算出工作量标时、综合记分等, 这项工作十分繁琐且容易出错, 各部门的审核难度大, 消耗大量的人力, 难以实现部门之间的信息共享与沟通。

采用轻量级框架技术, 可以快速构建教师考核系统<sup>[2]</sup>。使得对教师教学科研等各项工作的量化考核方便快捷。教师在客户端可随时输入自己已经完成的工作数据(如: 发表的论文题目, 作者排名, 出版物), 由系统按学校的考核办法自动计算出应该获得的考核积分。

## 1 轻量级的 Java EE 框架

轻量级 Java EE 软件架构主要由主流的开源框架 Struts2、Spring、Hibernate 根据其各自的应用

特性进行整合而成, 选择以 Spring 框架为核心并整合 Struts2 和 Hibernate 的框架组合。

### 1.1 Struts2 框架

Struts2 是实现了 MVC 模式的框架, Struts2 将一个 Web 系统的程序分为模型、视图和控制器 3 个部分。模型由 JavaBean、EJB 等组件构成; 视图由 JSP 文件、POJO 对象组成; 控制器由 Action 来实现。

### 1.2 Spring 框架

Spring 框架是一个开源且基于 POJO 的轻量级 J2EE 应用框架, 它是为了解决企业应用程序开发复杂性而创建的。Spring 是一个以依赖注入(DI) 原则和面向方面编程思想(AOP) 为基础的轻量级框架。

### 1.3 Hibernate3 技术

Hibernate 本质上是一个面向 Java 开发环境的 O/R 数据库映射工具。借助于 Hibernate 的 O/R 技术, 可以非常轻松地将以面向对象思想封装对象的各个状态映射到基于 SQL 的关系模型数据结构中去。

## 2 教师考核系统设计与实现

### 2.1 功能分析

根据系统应用功能需求，将教师考核系统设计为成果管理、人员管理、院系考核、考核管理、报告管理和系统管理模块，通过这6个功能模块，达到系统的要求。

(1) 成果管理：各项教学科研成果的录入，科研处对成果进行审核、评定。

(2) 人员管理：对本院系教职工信息管理。

(3) 院系考核：院系考核组人员查看本院系教职工的考核信息，并对隶属于该院系的教职工进行院系内考核，形成初步考核结果。

(4) 考核管理：包括制定与修正考核细则、发布考核事项。其中，学校考核组成员可对院系考核结果进行审核，形成最终考核结果。

(5) 报表管理：包括人员考核信息报表、成果统计报表、院系考核结果报表、学校审核报表。

(6) 系统管理：包括院系机构管理、系统用户组管理、基础数据管理。

### 2.2 系统总体架构设计

本系统基于Java EE平台开发，采用MyEclipse6.0为开发工具、SQL Server为数据库系统、Tomcat为Web服务器。参照了Java EE分层设计模型，将整个系统设计分为持久、业务逻辑层和表现层<sup>[3]</sup>。这种分层方式使整个系统变得更加清晰，切入点的分离使得每个部分的责任独立，层与层通过接口的方式通信，这也遵循了针对接口编程的原则。教师考核系统架构如图1。

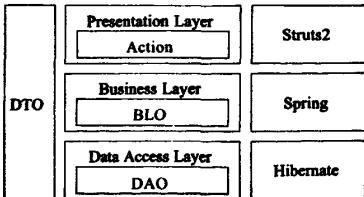


图1 教师考核系统软件架构图

教师考核系统软件架构平台中采用了如下的整合策略：

(1) Struts2中的Action和数据逻辑层的数据操作bean交由Spring容器来管理，在Struts2中采用伪Action来替代真实的Action，真实的Action位于Spring容器中。

(2) 层与层之间通过DTO进行数据传输。

(3) 业务逻辑bean与数据访问bean中的POJO对象由Spring注入。

(4) 数据访问bean仅针对POJO对象操作，而POJO对象与数据库中数据的映射、数据的关系逻辑及数据的持久化工作由Hibernate来实现。

### 2.3 持久层的设计与实现

持久层的目标是为系统提供一个统一、安全、并发性良好的数据持久化机制。它能有效降低业务层和数据层的耦合程度，使得开发者更加专注于构建业务模型和实现业务层的业务逻辑。本系统采用Hibernate作为持久层的ORM框架，并且使用数据访问对象(DAO)模式。部分代码如下：

```
import org.springframework.orm.hibernate3.support.HibernateDaoSupport;
.....
public class EppEmpDAOImpl extends HibernateDaoSupport implements EppEmpDAO {
    public void createInfo(EmpData data) throws DataAccessException {
        try {
            eppEmpEntity.create(data);
        } catch(Exception sql){
            SQLException sqle = (SQLException)sql.getCause();
            System.out.println(sqle.getErrorCode());
        }
    }
    .....
}
```

### 2.4 业务逻辑层的设计与实现

在业务逻辑层，我们使用了Spring框架，它不仅可以很方便地组织业务逻辑，还可以在事务管理、AOP等很多层面上发挥作用。在本系统中，采用setter依赖注入的方式，利用IOC容器，可通过XML的方式很容易地组织bean。

我们在业务层主要完成下面2项工作。

#### 2.4.1 在applicationContext.xml中配置事务

为了避免在业务逻辑层中重复编写大量的事务处理代码。我们在教师考核系统中，把事务管理定义在transactionManager配置上，这样开发人员

只须关注业务逻辑，而将事务管理完全交给配置文件，这样做的好处是既节约开发时间又可以提高开发效率。配置文件 applicationContext 的部分配置如下：

```
<!-- === 事务管理配置 ==-->
<bean id="transactionManager" class="org.springframework.orm.hibernate3.HibernateTransactionManager">
    <property name="sessionFactory">
        <ref local="sessionFactory" />
    </property>
</bean>
```

#### 2.4.2 结合 Spring 和 Hibernate

利用 Spring 和 Hibernate 的整合，使得业务逻辑层仅需要通过持久层来访问底层数据。而且，由于使用了 IOC，业务层不需要自己生成 DAO，仅需要声明 DAO 类型。DAO 的具体实现类型在 Spring 的 XML 文件中配置，这样就更加有效地降低了业务层和持久层的耦合。任何组件都可以最大程度地得到重用。当我们开发一个业务逻辑组件时只需要使用 setter 方法进行依赖注入，便可轻松实现组件间的引用。

#### 2.5 表现层的流程设计与实现

##### 2.5.1 表现层的流程设计

基于 Struts2 MVC 的表现层架构模型如图 2。

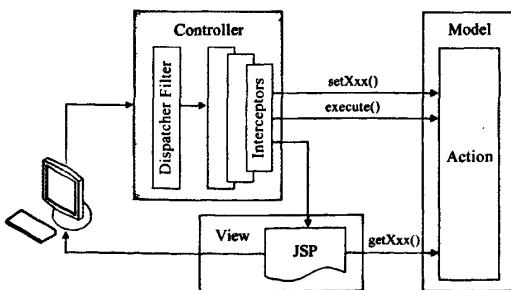


图 2 基于 Struts2 MVC 的表现层架构模型

图 2 描述了 Struts2 架构中的模型、视图和控制器。

(1) 从图 2 中可以看出客户端提交一个 (HttpServletRequest) 请求，请求先后被提交到一系列 (主要是 3 层) 的过滤器，如 ActionContext-CleanUp、其他过滤器 (SiteMesh 等)、FilterDispatcher。

(2) FilterDispatcher 是控制器的核心，就是 MVC 的 Struts2 实现中控制层 (Controller) 的核心。

(3) FilterDispatcher 判断 ActionMapper 是否需要调用某个 Action 来处理这个 (HttpServletRequest) 请求。

(4) 一旦 Action 执行完毕，根据 struts.xml 中的配置，转到对应的结果视图。

MVC 模式把应用分为输入、处理、输出 3 部分，从而减弱了 Action 控制类和业务逻辑接口之间的耦合。

##### 2.5.2 表现层的实现

限于篇幅，这里只给出与添加考核细则页面相关的 QualifyItemAction 类的部分实现：

```
public String createQualifyItem () {
    qualifyItemBLO.create(item);
    return SUCCESS;// 添加成功!
}
```

从以上代码可以看出，表现层中的数据表示是 Action 类的中对象属性，表现层与业务逻辑层进行通信时使用 POJO 进行。

### 3 结束语

教师考核信息化是学校信息化进程的战略步骤之一，其质量对以后的信息整合工作有着较大影响。本系统集成 Struts、Spring 和 Hibernate 3 种框架技术，快速实现了一个 B/S 结构的教师考核系统。简化了系统的开发、管理与维护，提高了开发速度，有利于系统的集成和扩展。

#### 参考文献：

- [1] 秦峰. 基于 Web 技术的教师考核系统的实现 [J]. 安徽工业大学学报, 2003, 20 (3): 119-223.
- [2] 杨兴春, 谭石, 董文, 等. 基于轻量级 J2EE 构架的高校教务管理系统的设计与实现 [J]. 计算机系统应用, 2007 (3): 42-43.
- [3] Spring Framework Organization. The Spring Framework - Reference Documentation version 2.5[EB/OL]. 2008-10-31. <http://static.springsource.org/spring/docs/2.5.x/reference/index.html>
- [4] 张俐, 张维玺. 基于 JavaEE 的固定资产管理系统的设计与实现 [J]. 计算机工程与设计, 2009, 30 (16).