

基于轻量级 J2EE 框架的网络教学系统

孙小锥, 上官右黎, 文福安

(北京邮电大学网络技术研究所, 北京 100876)

摘要: 针对国内现有网络教学系统的局限性, 在分析轻量级 J2EE 架构基础上, 分离了表示层、业务逻辑层和数据持久层, 集成并扩展 Spring, Hibernate, Struts 开源框架, 开发了一套基于国产基础软硬件(Linux+NC)的网络教学系统。该系统具有良好的复用性、扩展性和可维护性。

关键词: 轻量级; Spring 框架; Hibernate 框架; Struts 框架; 网络教学

Network Teaching System Based on Lightweight J2EE Framework

SUN Xiao-zhui, SHANGGUAN You-li, WEN Fu-an

(Institute of Cyber Education Technology, Beijing University of Posts and Telecommunications, Beijing 100876)

【Abstract】 Considering the limitation of the domestic existing network teaching system, and by analysing lightweight J2EE framework, the Web layer, business logic layer and data persistence layer are separated, while Spring, Hibernate and Struts are integrated and extended to develop a network teaching system based on domestic foundation software and hardware (Linux + NC).

【Key words】 lightweight; Spring frame; Hibernate frame; Struts frame; network teaching

我国的网络教育大多建立在 Wintel(Windows 操作系统和 Intel CPU)平台上, 现有的部分基于 J2EE 框架的教学软件虽然具有跨平台性, 但须运行在昂贵的商业应用服务器和商业数据库中, 限制了网络教育的发展。鉴于此, 本文分析了实际需求和国内现有系统, 构建并开发了一个多层 B/S 的网络教学系统。

1 J2EE 框架

传统 EJB 应用代码有依赖容器、启动缓慢、部署复杂、资源占用高、对服务器要求高等缺点, 轻量级框架与 EJB 框架不同, 它简化了编程模型并且是非入侵性的, 使应用程序代码对框架的依赖最小化^[1]。

1.1 Struts 框架

Struts^[2]是基于 MVC 设计模式的框架。MVC 设计模式由控制器(controller)、模型(model)、视图(view)3 部分组成。在 Struts 中, 控制器由 Struts 框架提供 ActionServlet 类和 Action 类来实现; 模型是应用程序状态的封装; 视图用来显示用户请求, 主要使用 JSP 来完成表示逻辑。ActionServlet 接收来自客户端的 HTTP 请求, 根据 Struts 配置文件, 把请求转发给相应的 Action 对象, 然后 Action 的处理结果决定页面跳转, 将处理结果返回给客户端。而 VO 和 ActionForm 则实现了 Model 和 View 之间的数据交互。表示层处理请求流程见图 1。

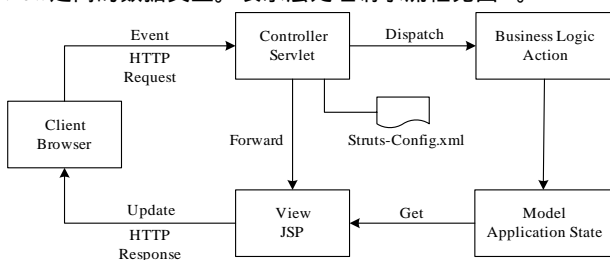


图1 表示层处理请求流程

1.2 Hibernate 框架

Hibernate 是基于对象/关系映(Object/Relational Mapping, ORM)的持久化框架, 它对 JDBC 进行了轻量级的对象封装, 使编程人员可以使用面向对象的编程思维来操纵数据库。Hibernate 不仅管理 Java 类到数据库表的映射, 还提供了数据查询和获取数据的方法, 可以大幅度减少开发时人工使用 SQL 和 JDBC 处理数据的时间。同时, 它提供了一种类似于 SQL 的语言——Hibernate 查询语言(HQL)。和 SQL 不同, HQL 是一种面向对象的查询语言, 可查询以对象形式存在的数据。

1.3 Spring 框架

Spring 框架^[3]是优秀的轻量级 J2EE 应用框架, 它提供了控制反转(IoC)容器和面向方面编程(AOP)的功能, 解决了很多 J2EE 开发中的常见问题, 并能方便地把各种组件和框架整合进来, 形成一个连贯的整体构架。Spring 的轻量级的 bean 容器为业务对象(business objects)、DAO 对象和资源对象(如 JDBC 数据源或者 Hibernate SessionFactory 等)提供了 IoC 类型的装配能力, 并支持事务管理、远程调用以及对多种 O/R 映射工具。

2 系统整体架构及分层实现

2.1 系统整体架构

在本网络教学系统中, 分别建立校级网络教学门户和远程公共门户, 用于访问本地资源和公共资源, 从而更大程度上实现网络教育资源共享。校级网络教学门户包括教学管理、智能答疑、网上学习、课程点播、资源服务等模块。远程公共门户包括教师指导、新闻公告、网上交流、远程课堂、资

基金项目: 国家“十五”科技攻关基金资助项目(2005BA115A01)

作者简介: 孙小锥(1982 -), 女, 硕士研究生, 主研方向: 网络交互, 多媒体应用技术; 上官右黎, 教授; 文福安, 教授

收稿日期: 2007-03-26 **E-mail:** sxz0324@gmail.com

源服务、远程支持等模块。

2.2 系统分层实现

本系统分离了表示层、业务逻辑层和数据持久层。其中，业务逻辑层使用 Façade^[4](外观总管 Manager) 模式为表示层提供一个统一的接口；持久层使用 DAO 模式为业务层提供数据抽象层接口，从而保证了表示层与业务逻辑层、业务逻辑层与持久层间的松散耦合。系统整体框架实现如图 2 所示。

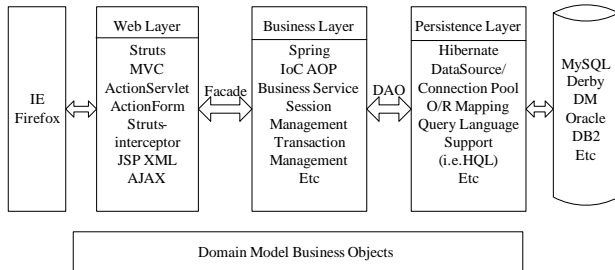


图 2 系统框架

2.2.1 表示层

表示层使用 Struts 框架。前端控制器 ActionServlet 通过查找 Struts 配置文件决定把请求(*.do 等，具体在 Web.XML 中配置)交给哪个 Action 处理，Action 再调用相应业务逻辑的服务组件，并根据处理结果调用 Forward 对象所指定的响应返回。数据验证机制使用了可插座的 Validator 框架，提高了复用性并保证了验证规则和应用程序的松耦合。与 Spring 的集成采用将 Struts Action 管理委托给 Spring 框架的方法，使用 Spring 的 Context Loader-Plugin 即 Struts 的 ActionServlet 来装载 Spring 应用程序环境。

根据实际需要，对 Struts 框架进行了扩展。BaseAction 中使用 strategy 模式和回调模式，定义所有 Action 的一些公共逻辑，从而提高代码的可重用性。使用自己创建的 Struts 拦截器 struts-interceptor，其中融入了 AOP 的思想，将一些横切性(crosscutting)问题模块化。前端机制中采用了 AJAX(Asynchronous JavaScript + XML)技术，使 Web 应用程序不必完全依赖服务器重新载入页面来向用户呈现更改。

2.2.2 数据持久层

持久层封装了数据访问细节，为业务逻辑层提供服务。在系统中，持久层采用了 Hibernate 框架。创建需要持久化的实体类和映射文件(*.hbm.xml)。Hibernate 通过映射文件 *.hbm.xml 建立对象与数据库表的关联，并把对持久化对象的操作转化为对数据库表的操作。

数据库访问使用 DAO 模式，抽象和封装对数据源的操作。创建了持久化类接口 studyDao 及其实现 studyDaoImpl。Spring 对 Hibernate 的 DAO 有良好支持，提供 HibernateDaoSupport 类，可以方便地调用 save(), delete(), update() 等 Hibernate 的 Session 的操作。通过让 studyDaoImpl 扩展 HibernateDaoSupport 类，获得 HibernateTemplate 的引用。

此外，Hibernate 有很好的跨数据库可移植性，对当前流行的数据库都提供了相应的 Dialect 进行优化操作，提高了系统效率。

2.2.3 业务逻辑层

系统中，为学习模块中的业务层定义了一个接口 studyManager 及接口的实现类 studyManagerImpl，相关组件则基于接口 studyManager 定义。采用这种面向接口而非面向类的编程，可以减少组件间的耦合度并且极大提升开发过程

中测试的灵活度。

(1)使用 IoC 模式管理业务组件的依赖关系。须声明由容器管理组件之间的依赖关系，把对组件之间依赖关系的控制进行了倒置。这样，组件间就不存在硬编码的关联，任何组件都可以最大程度地得到重用。Spring 通过 setter 方法注入依赖关系(即 Type 2 IoC)，并在配置上下文文件 application-Context.xml 中便配置可轻松实现组件间的引用，指定之间的依赖关系。如下：

```
<bean id="studyManager" parent="txProxyTemplate">
  <property name="target">
    <bean
class="com.bupticet.local.study.service.impl.StudyManagerImpl">
      <property name="studyDao">
        <ref local="studyDao" />
      </property>
    </bean>
  </property>
</bean>
```

IoC 将组件“依赖关系”的职责移交给容器来处理，通过 1 个 bean 声明，把业务对象及其依赖的 DAO 组装起来。

(2)由于事务管理有一个很明确的横切概念，因此 Spring 的 AOP 最适合。进行声明式事务管理，从而避免在众多方法中重复编写大量的事务处理代码。系统中，把事务管理定义在 studyManager 方法上，这样核心代码就只须关注业务逻辑，而将事务管理完全交给配置文件。事务处理的 AOP 配置如下：

```
<bean id="transactionManager"
class="org.springframework.orm.hibernate3.HibernateTransaction
Manager">
  <property name="sessionFactory">
    <ref local="sessionFactory" />
  </property></bean>
<bean id="txProxyTemplate" abstract="true"
class="org.springframework.transaction.interceptor.Transaction
ProxyFactoryBean">
  <property name="transactionManager">
    <ref bean="transactionManager" />
  </property>
  <property name="transactionAttributes">
    ...
  </property>
</bean>
<bean id="studyManager" parent="txProxyTemplate">
  <property name="target">
    ... </bean>
```

首先，定义事物管理器 transactionManager，它使用 Spring 提供的 HibernateTransaction-Manager。Spring 启动时会自动实现“依赖注入”，将 sessionFactory Bean 注入。然后，定义 txProxyTemplate，并在其中注入事务管理器。最后，txProxyTemplate 中定义了一些基本的事务策略，主要通过 transactionAttributes 属性实现，这样就能以通配符的方式进行匹配声明，把以 save 和 remove 等开头的方法纳入事务管理范围。若这些方法在执行中出现 Unchecked Exception，则 Spring 将当前事务回滚；若正常结束，则提交事务。业务接口“study-Manager” bean 同“txProxyTemplate” bean 是父子节点关系，它继承了 txProxyTemplate 的事务策略。也可根据实际需要，

(下转第 270 页)