

# 基于 JSF + Spring + Hibernate 集成架构的 Web 信息系统设计

范会联<sup>1,2</sup>, 张玉芳<sup>1</sup>

(1. 重庆大学 计算机学院, 重庆 400044; 2. 涪陵师范学院 计算机科学系, 重庆 408003)

**摘 要:**利用 DAO 模式集成 Spring 与 Hibernate, 用 Service Locator 模式集成 JSF 与 Spring, 使得它们优势互补, 形成一个统一的架构, 有助于快速构建基于 Web 的信息系统。论述了集成这三个架构的原理和方法, 介绍基于 JSF, Spring 和 Hibernate 架构实现的网上选课系统的设计, 以说明利用这个集成架构开发功能强大而又灵活的信息系统的高效方法。

**关键词:**Spring; Hibernate; JSF; 面向接口编程; 选课系统

中图分类号: TP311.52

文献标识码: A

文章编号: 1673-629X(2007)03-0023-03

## Web Information System Based on Integrated Architecture of JSF, Spring and Hibernate

FAN Hui-lian<sup>1,2</sup>, ZHANG Yu-fang<sup>1</sup>

(1. College of Computer Science, Chongqing University, Chongqing 400044, China;

2. Department of Computer Science, Fuling Normal University, Chongqing 408003, China)

**Abstract:** Integrating Hibernate and Spring framework with DAO pattern, and integrating Spring framework with Service Locator pattern, which not only avoid their disadvantages while retain their advantages, but also make them more powerful. The mechanism and the way to integrate the three architectures were discussed, and an efficient way was brought forth to develop a powerful and flexible information system by explaining the development of a selecting courses management system based on the integrated architectures.

**Key words:** Spring; Hibernate; JSF; interface-oriented programming; selecting courses system

## 0 引 言

随着 Internet 技术的发展和 Web 信息系统的广泛应用, 基于 MVC 的多层架构设计思想已成为解决 B/S 系统体系结构的成功模式, 因此在 Web 前端开发、业务逻辑管理、数据持久层等诸多领域涌现出大量实用工具和框架, 其中在前端开发领域出现基于 UI (User Interface) 组件构建 Web 界面的 JSF (Java Service Faces), 基于对象/关系数据库映射 (Object/Relational Mapping, ORM) 解决数据持久层问题的 Hibernate 和提供 IoC (Inverse of Control)、AOP (Aspect-Oriented Programming) 服务的 Spring Framework 成为目前 Java 社区的讨论热点。通过整合以上技术, 可以得到一个构建具有灵活性、低耦合及易于维护的 Web 信息系统

的完整解决方案。

## 1 基于 JSF, Spring, Hibernate 的 Web 信息系统的架构

Hibernate 是一个对象/关系数据库映射工具, 把用对象模型表示的对象映射到基于 SQL 的关系模型结构中去。Hibernate 不仅管理 Java 类到数据库表的映射, 而且还提供数据查询和获取数据的方法以及延迟加载等重要功能, 并且 Hibernate 拥有功能非常强大的查询语言 (HQL, Hibernate Query Language), HQL 最重要的特点是完全面向对象, 查询得到的是持久对象, 而不是数据库的记录。因此, 数据持久层采用 Hibernate 作为解决方案, 首先写出基本的 DAO 接口, 然后基于 Hibernate 生成的 POJO 类为访问对象给出 DAO 实现 (同时还包含 POJO 与数据库之间的映射文档), 供业务对象访问<sup>[1]</sup>。

Spring 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

收稿日期: 2006-07-03

基金项目: 重庆市教育科研项目 (04-GJ-063)

作者简介: 范会联 (1971-), 男, 重庆人, 硕士研究生, 讲师, 研究方向为网络信息系统、软件工程; 张玉芳, 博士, 副教授, 硕士生导师, 研究方向为远程教育、数据挖掘、信息网络与信息系统。



Hibernate Session Factory 等)提供了 IoC 类型的装配能力。Spring 使用 XML 格式的应用配置文件为开发者提供一种通过解析定制的属性文件来手动管理单实例对象或者工厂对象的选择性。由于 Spring 将非入侵性作为一个重要的目标,因此由 Spring 配置管理的 bean 对象均不需要依赖 Spring 自有的接口及类就可以完成对它们的 bean 属性的设置。因此,按类似 Java Bean 的方式实现系统的业务类,业务类对 DAO 类的依赖性通过 IoC 基于设值(setter-based)方法注入,即在业务类中设置 DAO 类型的私有变量,建立该私有变量的 set × × × ()方法,然后通过 Spring 配置文件设置属性名为私有变量名,其值为 DAO 类<sup>[2]</sup>。

JSF 侧重于用户界面的创建和管理,具有一套预制的 UI 组件和基于事件驱动编程模型。利用 JSF 所提供的组件模型可以方便地创建用户接口,同时,也可执行组件中封装事件处理、输入验证等操作。JSF 所提供的运行环境可以很容易地将客户端事件与服务器端的事件处理程序绑定,将 HTTP 请求映射为具体组件的事件处理,从而改变了以往基于 Java Web 应用的 request-response 处理机制。同时它还提供可扩展的组件模型,此组件模型包含了一个重要功能——渲染组件(把组件所包含的信息转换成客户端能够接受的表达形式,比如对于 Web 浏览器而言就是 HTML 页面,而对于手机等无线移动设备所使用的 WAP 浏览器而言就是 WML 页面,每个组件都具有各自的渲染组件功能)<sup>[3]</sup>。所以,选用 JSF 构建 Web 系统表现层,创建受 JSF 管理的 Java Bean 用于封装应用层数据和对事件的响应,事件的处理通过调用 Spring 管理的业务类 Bean 完成,对 Spring 管理的业务对象的查找通过 Service Locator 模式实现,从而达到应用程序逻辑和表示的完全分离<sup>[4]</sup>。

图 1 为以上 3 个架构的集成结构。

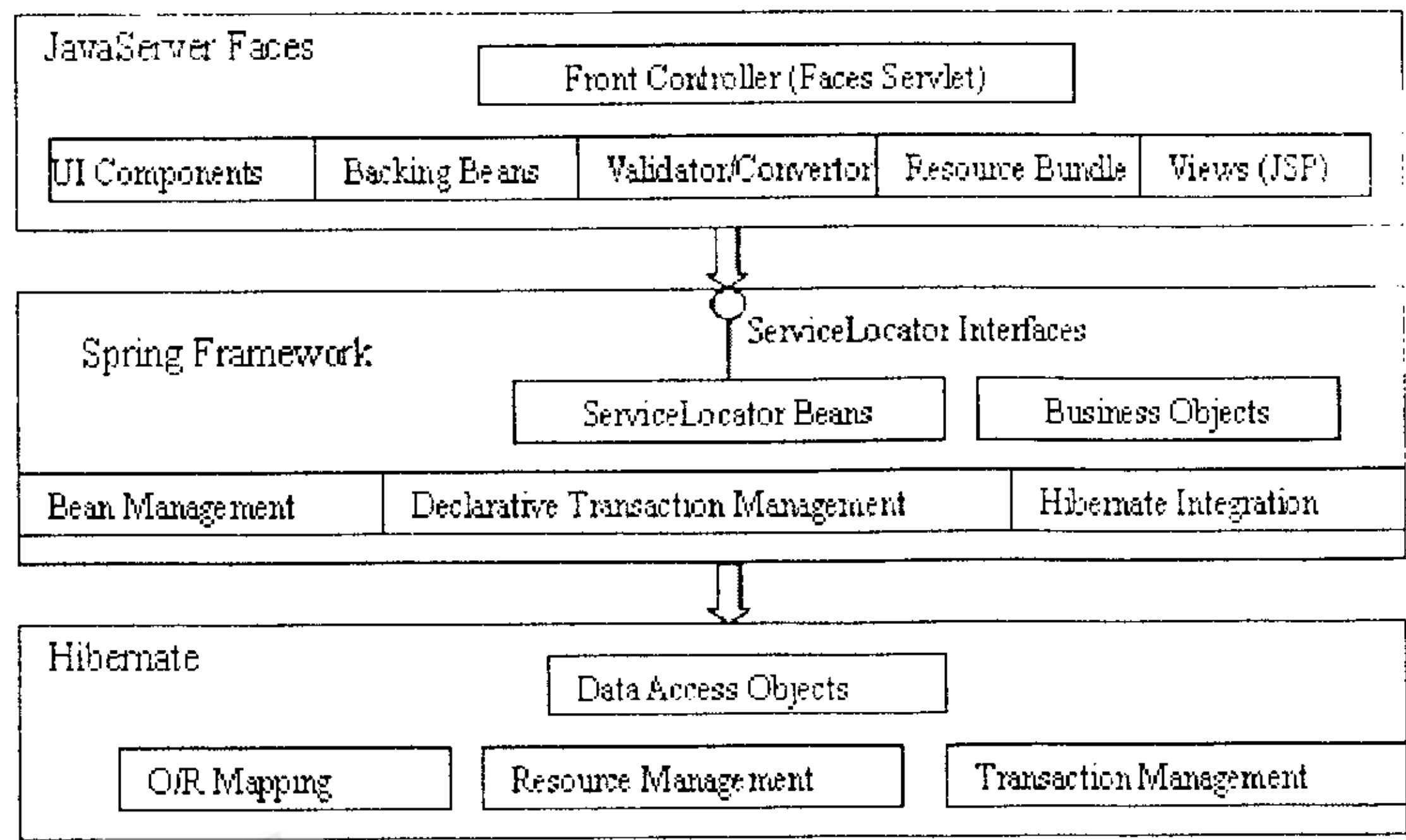


图 1 基于 JSF + Spring + Hibernate 的 Web 信息系统的集成架构

## 2 面向接口的系统设计

保持系统架构稳定性的一种方法是坚持面向接口编程的设计思想。由于系统功能相对稳定,而功能的实现可以有不同方法,所以在分层模式中,用接口对类按功能进行分组,分离功能与实现,使位于不同层次的程序员只需要清楚接口发布的方法就可以了。图 2 展示了网上选课系统按不同框架管理的部分类和面向接口的设计方法。

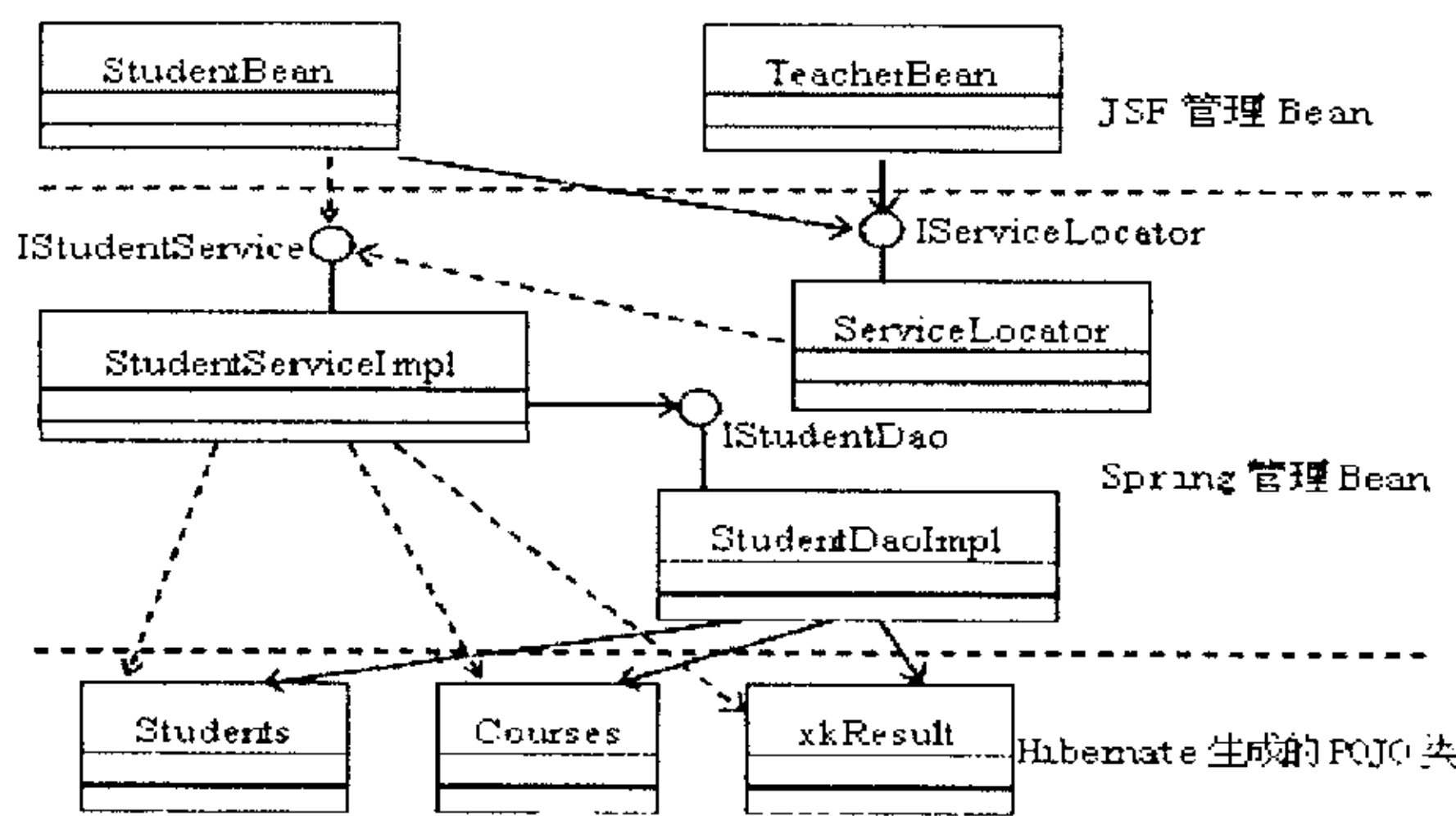


图 2 分层管理类及面向接口设计示例

## 3 基于 JSF + Spring + Hibernate 集成架构的 Web 信息系统设计

### 3.1 DAO 与 Hibernate 架构

选课系统的数据库逻辑设计如图 3 所示,映射出的部分 POJO 类见图 2。Hibernate 提供面向对象的设计与传统关系型数据的整合,对实体进行面向对象的封装,实现对象到关系型数据库的持久化服务。Hibernate 利用 Hibernate 自带的 Hibernate-Extentions, MiddleGen, Hibernate-Doclet 等辅助工具,采用从数据模型到类模型的开发方式,即按数据库定义→HBM 映射→POJO(Plain Ordinary Java Object)类的步骤进行。首先用 E-R 工具完成对数据模型的定义,然后顺序使用 MiddleGen 和 hbm2java 这两个工具分别生成 Hi-

bernate 映射文件和 POJO 类。POJO 类是不包含逻辑代码的普通对象,但从数据层面上看,POJO 类是作为数据实体的对象化表现形式。POJO 类不但把数据表的主键及其他字段映射成类中的属性,而且还自动把数据表间的关联映射为类中集合类型的属性,并且在需要的时候能自动加载关联实例,达到延迟加载的目的<sup>[5]</sup>。

通过对 JDBC 的封装,Hibernate 自动把对 POJO 类对象的操作转化为对数据库的操作,因此,DAO 层的 CRUD(Create, Retrieve, Update, Delete)操作从针对数据库中的实体记录转为面向 POJO 类。这样,DAO 开发所



需要的代码数量就大为减少。

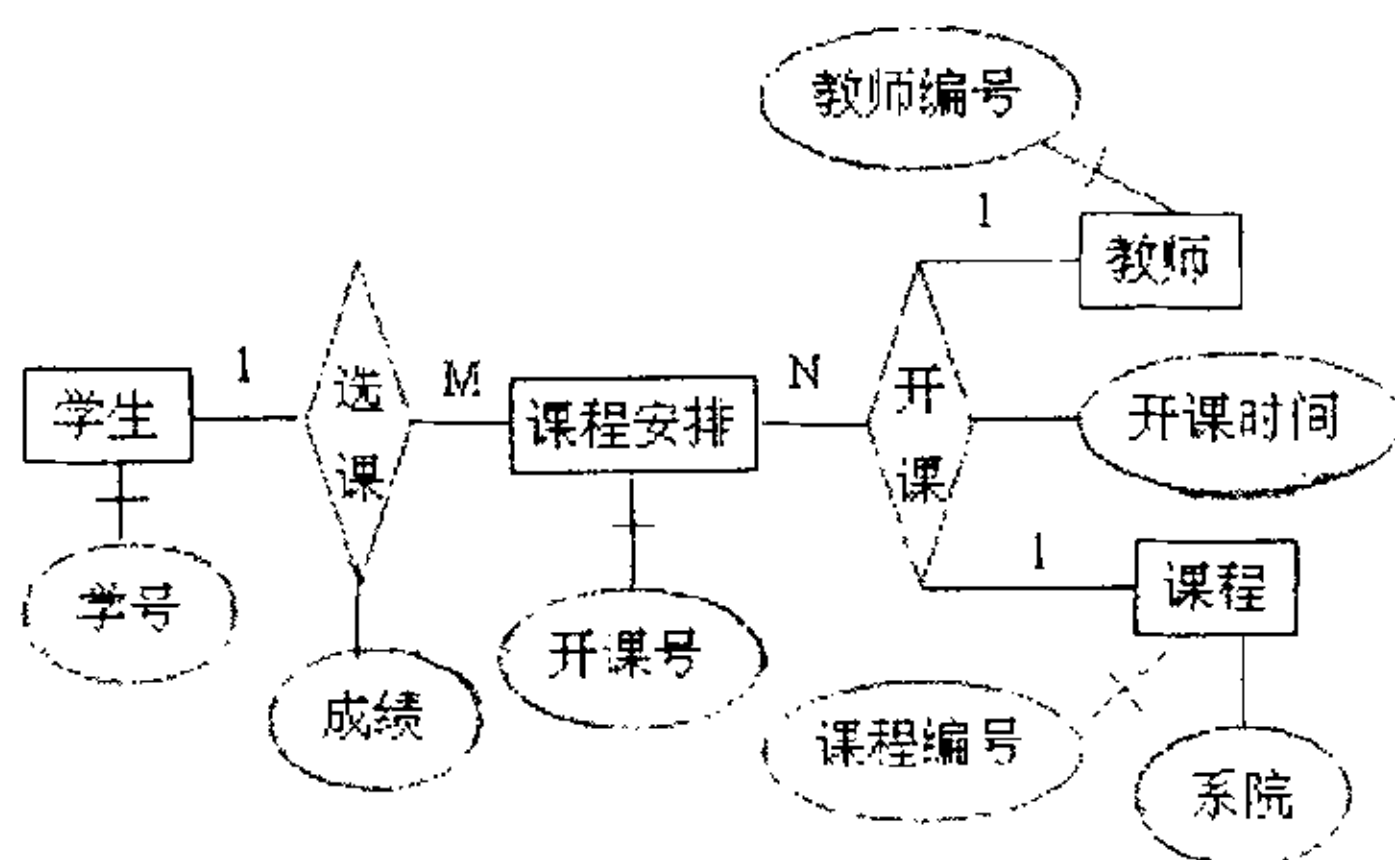


图3 选课系统数据库逻辑设计

### 3.2 集成 Spring 与 Hibernate

Spring 框架提供的控制反转 (IoC) 和面向方面编程 (AOP) 的插件式架构降低了应用组件之间的依赖性,借助于 XML 配置文件,能够在运行时连接不同的应用组件。IoC 是 Spring 框架的基础, Spring 框架在基于 IoC 的基础之上,提供对 Hibernate 的优秀支持,整合了许多 IoC 的方便特性。并且 Spring 框架对 Hibernate 的支持符合 Spring 通用的事务和数据访问对象 (DAO) 异常层次结构。

Spring 通过在配置文件中定义 Hibernate 资源,配置 SessionFactory 和 HibernateTemplate,从而实现 Hibernate 资源的依赖注入。配置文件中将如 JNDI 或 JDBC DataSource、Hibernate SessionFactory 等资源的查找分别定义为一个 Bean。应用对象对资源的访问只需要通过 Bean 引用。通过实现 HibernateTemplate 的 HibernateCallback 回调接口,可以有效地进行 Hibernate 数据访问,并且在任何情况下都由 HibernateTemplate 来管理 Session 的开闭以及自动的多方事务。模板实例是线程安全和可重用的,因此它们可以做为其他类的变量。对于简单的单步动作,象 find, load, saveOrUpdate 或者 delete 的调用, HibernateTemplate 提供更为便利的选择以代替回调方式的执行。此外, Spring 提供了一个方便的基本类 HibernateDAOSupport 类,该类提供 setSessionFactory 方法来接受一个 SessionFactory,同时提供 getSessionFactory 和 getHibernateTemplate 方法供其继承类使用。因此,网上选课管理系统中的 DAO 类都继承自 HibernateDAOSupport 类,并借助 Spring IoC 容器注入对 HibernateTemplate 的依赖性。

Spring AOP 提供关注系统级服务 (如:系统日志、事务、安全性) 的手段,弥补了 OO (面向对象) 技术在用于系统级服务设计时的不足。Spring 把事务管理抽象为 AOP 中的 Aspect (方面) 来看待,支持各种事务管理器 (比如 JTA 事务、Hibernate 事务等),通过提供 TransactionProxyFactoryBean 类,实现类似 EJB 的声明性事务,事务行为和隔离层次的变化可以通过配置文

件来改变而不会对业务对象的实现造成影响。在选课系统设计中,业务逻辑包含在 StudentService (提供学生登陆、选课、查询等功能)、TeacherService (教师成绩管理、查询等功能)、AdminService (管理员的功能) 等 Java 类中。比如 StudentService 类含有学生选课方法,该操作涉及新增或更新学生的选课结果表,同时还要更新对应开设课程的可选人数和已选人数。通过对 StudentService 类引入 TransactionProxyFactoryBena 代理工厂,借助 IoC 注入对相应 DAO 类的依赖,然后为 transactionAttributes 属性指定适宜的事务策略,从而达到方便灵活配置对业务类事务管理的目的。并且, Hibernate 的延迟加载功能在事务中也可得到很好支持。

### 3.3 JSF 在系统设计中的应用

JavaServer Faces 负责与客户端设备交互,为基于 Java 的 Web 应用提供一种用户界面开发框架以及应用逻辑和业务逻辑的连接。它通过一个控制器 FacesServlet 提供 Web 应用程序生命周期管理,而且能够在服务器上处理客户端事件、输入验证等用户界面管理问题<sup>[6]</sup>。

选课系统在设计层设计了 StudentBean、TeacherBean 等受 JSF 管理的 Java Bean (Managed Bean),这些 Bean 由表示层数据和相关行为或事件组成 (这在 Struts 中需要分别用 ActionForm 收集表示层数据, Action 构成行为),用值绑定表达式和方法绑定表达式来把 JSF 组件与应用层的 Managed Bean、验证器类、转换器类等对象联系起来。Managed Bean 中的方法调用业务逻辑层中定义的 Bean (受 Spring 管理) 中相关方法来完成,而行为或事件的处理结果可作为页面导航的依据。另外,在 JSF 层的 Managed Bean 查找 Spring 中受管 JavaBean 可以通过以下两种的方法实现:一种是使用 Spring 框架或者 JSF-Spring 项目提供的变量解析器来替换 JSF 自身的变量解析器;另一种是不替换 JSF 变量解析器,而是利用 Spring 框架提供的 FacesContextUtils 在系统中构造一个 ServiceLocator 类来统一定位 Spring 中的受管 Bean。

缓存对于系统性能的提升具有非常重要的作用,网上选课系统除了利用 Hibernate 提供的纵向两级缓存 (Session Level 和 SessionFactory Level) 外,还另外建立应用程序范围 (Application) 和会话范围 (Session) 两种缓存。其中在 Application 范围内缓存了系统的常用属性、本学期所有课程、任课教师信息和开课记录,用 HashMap 按“关键字→HashMap (而 HashshMap 又由页码→当前页的记录链表 List 组成)”的方式进行组织,

(下转第 108 页)



4 层小波特征来识别,用 20 个小波描述子系数表示形状,这些数据已证明能基本精确表达形状。

在进行了一些最初的试验后,多层感知网络的结构确定如下:20-10-4(20 个输入接点、10 个隐藏神经元和 4 个译码被分为 1 或 0 的输出神经元)。并且在训练阶段,向原始的样本进行旋转、平移和放大并且加入了噪音<sup>[10]</sup>产生了新的学习模式,在实验中用了 300 个学习模式进行训练,达到了满意的效果。这样就增加了神经网络的一般化识别能力。在实验中信噪比值是从 40dB 到 3.6dB。40dB 的噪音形状几乎没有相应的影响,充满噪音的形状和有 3.6dB 噪音的形状意味着形状完全被噪音覆盖。训练后的系统通过对 100 个模式进行识别得到表 1 的数据,它表明了受干扰形状识别的结果。

表 1 加入噪音的识别数据

信噪比(dB)	识别率(%)
40	100
30	100
18	100
15	97.9
10	89.1
5	60.0

4 结 论

文中介绍了小波描述子和神经网络相结合的形状识别方法。通过实验表明该方法能够更加鲁棒地识别

和区分具有旋转、平移和尺度变化的物体。由于神经网络具有良好的容错性能与泛化能力,因此本方法对于噪声不敏感,对于信噪比大于 15dB 的形状有很好的识别能力。

参考文献:

[1] 丁险峰,吴 洪,张宏江,等.形状匹配综述[J].自动化学报,2001,27(5):678-694.

[2] Chuang Gene C H, Kuo C C J. Wavelet descriptor of planar curves: theory and application[J]. IEEE Trans Image Processing, 1996, 5(1):56-70.

[3] 杨翔英,章毓晋.小波轮廓描述符及在图像查询中的应用[J].计算机学报,1999,22(7):752-757.

[4] Pandya A S.神经网络模式识别及其实现[M].徐 勇,荆涛译.北京:电子工业出版社,1999.

[5] Navi R, Stantanu C, Subhashis B. Recognition of partially occluded objects using neural network based indexing[J]. Pattern Recognition, 1999, 32:1737-1749.

[6] 高 隽,张维勇,蒋建国,等.基于神经网络的形状识别系统及优化[J].系统工程与电子技术,1999,21(3):52-55.

[7] Vapnik V. The Nature of Statistical Learning [M]. New York: Springer, 1995.

[8] 刘宏申,秦 锋.确定轮廓形状匹配中形状描述函数的方法[J].华中科技大学学报,2005,33(4):13-16.

[9] 朱同林,彭嘉雄.轮廓形状匹配的函数小波特征方法[J].自动化学报,2001,27(6):855-859.

[10] Matsuoka K. Noise injection into inputs in backpropagation learning[J]. IEEE Trans Syst Man Cybern, 1992, 22: 436-440.

(上接第 25 页)

以便分页显示。在 Session 范围内缓存登陆用户个人信息。JSF 框架提供了方便的用户缓存实现方式,通过分别创建 ApplicationBean、SessionBean 并纳入 JSF 管理,在配置文件中设置 managed-bean-scope 属性为 application 或者 Session 实现数据缓存。

4 结束语

JSF 在基于 Java 的 Web 应用开发领域提供了一个可重用、可扩展、基于组件、工具友好的服务器端 UI (User Interface) 框架;Hibernate 完成对象和关系数据之间的映射;Spring 框架将业界广泛使用的技术集成进来,为解决 J2EE 平台技术间的使用隔阂起了重要作用。将该架构应用于网上选课系统的设计和开发,大大降低了表现层、控制层和数据访问层之间的耦合性,增强了系统的可扩展性和可维护性,同时利用缓存机制,减少了对数据库的直接访问次数,使系统具有良

好的性能。

参考文献:

[1] Hibernate org. Hibernate Reference Documentation[EB/OL]. 2005. <http://www.hibernate.org/hib-docs/v3/reference/en/html/>.

[2] Springframework. org. Spring Framework Documentation [EB/OL]. 2005. <http://www.springframework.org/documentation>.

[3] 朱庆生,葛 亮.新一代 Web 应用框架 JSF[J].计算机科学,2005(7):224-225.

[4] SHEN DEREK YANG. Put JSF to wrok[EB/OL]. 2004-06. <http://www.javaworld.com/javaworld/jw-07-2004/jw-0719-jsf-p.html>.

[5] 冯国仕,李志蜀.基于 Struts 与 Hibernate 集成架构的项目管理系统[J].计算机应用,2005(8):1884-1885.

[6] GEARY D, HORSTMANN C. JavaServer Faces 核心编程 [M].王 军,马振萍,等译.北京:电子工业出版社,2005.