

# 基于 SSH 框架的社区管理信息系统设计

杨钟鸣

(东华理工大学信息与电子工程学院,江西 抚州 344000)

**摘要:**SSH 适用于涉及业务较多的中大型企业级开发。对于一些小型的项目、简单的应用,使用 JSP 以及 Servlet 等技术便足以完成。但对于中大型项目,其需求变更频繁、业务流程复杂、要求的维护性能高。根据这些特点设计了社区管理信息系统,并从视图层、数据层和业务层分别探讨了系统实现的关键技术。该系统的后台设计通过 Spring 对 Struts 和 Hibernate 的集成完成,前台 Web 页面运用具有 Ajax 技术的 Ext 的 js 框架完成。得出的结果充分验证了 SSH 对这类系统的可扩展性以及可维护性,也更加肯定了 SSH 对于此类项目的适用性。

**关键词:**SSH;企业级开发;可扩展性;可维护性;适用性

**中图分类号:**TP393 **文献标识码:**A **文章编号:**1674-3504(2010)04-379-05

随着 J2EE 的发展,J2EE 已经成为主流的企业级应用解决方案,但是 J2EE 编程正变得越来越复杂。为了降低成本,加快应用程序的开发和设计,J2EE 平台提供了一种基于组件的方法。对于一个应用模型,根据功能将其应用逻辑划分成为组件。其中包括 Struts, Spring, Hibernate 等架构,通过这些架构整合起来可以得到一个开发灵活、低耦合及易于维护的 Web 信息系统的完整解决方案。本系统正是运用 Struts1.3, Spring2.0 和 Hibernate 3.0 以及前台的 Ext js 框架。Struts 用于从页面表单获取数据并调用业务层进行数据的各种业务处理;Hibernate 用于 dao 层与数据库进行交互,对数据进行增删改查;Spring 则是封装 Hibernate 和 Struts,以便对二者进行管理,提高应用效率(谌湘倩等,2009)。

## 1 系统结构

B/S 架构的社区管理信息系统以基本人口和房屋数据为基础,通过消息传递机制、事物管理机制、工作流程等建立起一个可扩展、集成数据和应用的系统结构。它将社区各机构的多项工作集成起来,以进行更为有效的业务处理。

系统分为市政管理、社会稳定、市民呼声等功能模块(图1)。其中,市政管理模块包括对商铺小区、市场摊位、市政设施等进行相应信息管理,街道

内各个居委对本地街面环境、房屋搭建的问题反映;社会稳定又分为综治和信访稳定等子模块。综治模块是对某些高危人群、小区、四小行业房屋租赁等特定人群和地点进行监控管理;信访稳定模块则是通过网络办公实现信访办的日常工作;市民呼声模块是对市民通过各种渠道向政府提出意见的收集,其中包括短信、电话、邮件以及网络意见收集。整个系统对各类人群、企业单位等方面进行综合统一管理,使其更加具有“综合服务”的特性。

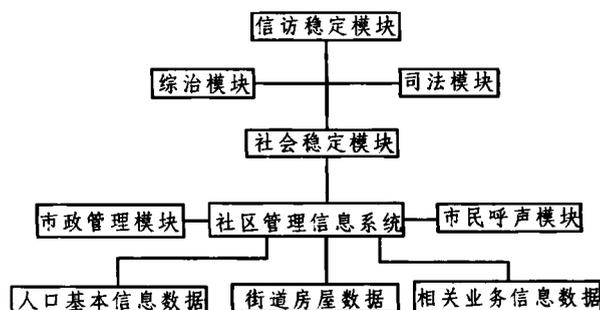


图1 系统结构图

Fig. 1 Picture of system structure

## 2 系统主要使用的技术

社区管理信息系统是一个应用平台,也是一个基于 JSON 和 Web Services 数据交换体系。后台以 SSH 框架管理各层,前台以 ExtJs 框架将 html, css 以及 Ajax 进行封装。

### 2.1 SSH (struts + hibernate + spring) 框架简介

Hibernate 是一个非常优秀的 ORM (Object-Relation Mapping, 简称 ORM) 轻量级框架。众所周知,

收稿日期:2010-06-24

作者简介:杨钟鸣(1985—),男,工学硕士,主要研究方向:分布式数据库与数据挖掘。

内存不能永久保存数据。要将数据永久保存只有将数据放入数据库中进行持久化。目前关系数据库仍然是使用最为广泛的数据存储库。关系数据库是非面向对象的。OOP是当今的主流, JAVA是一种非常强大的面向对象的语言。因此将两种思想结合起来将大大提高软件开发的效率。而Hibernate框架提供了这种将面向对象的程序设计(OOP)与关系数据库相互映射的机制。它把对象模型表示的对象映射到基于SQL的关系模型基础上,在JDBC的方式上进行轻量级对象封装。同时Hibernate还提供数据查询和获取数据的方法,减少使用SQL和JDBC访问数据库的时间(孙卫琴,2005)。

Hibernate对JDBC做了一定的封装,使得DAO层的代码得到充分简化(董洪彬,2004)。但对于某些特定的代码,如session的生成、事务的打开等,调用次数依然频繁。但在Spring管理下,Hibernate又得到了进一步的封装。而且spring对Hibernate是轻量级、非侵入式的封装,使得编程效率更加提高(朱庆伟等,2004)。

Spring是一个开源框架,是为了解决企业应用程序开发复杂性而创建的。框架的主要优势之一就是其分层架构,分层架构允许您选择使用哪一个组件,同时为J2EE应用程序开发提供集成的框架。使应用程序的各个部分之间尽量形成一种松耦合的结构,使其有更大的灵活性(高昂等,2005)。

Struts框架是基于MVC模式的框架,是一个免费开源的Web层的应用框架,主要采用JSP与Servlet技术实现,把Servlet、JSP、自定义标签和信息资源整合到一个统一的框架中,关注于控制器的流程而开发人员只需开发相应的Action类和相应的JSP组件等,就可以套用Struts框架,进行项目的开发。

## 2.2 ExtJs 框架简介

ExtJs是一种主要用于创建前端用户界面、用javascript编写的、与后台技术无关的前端ajax框架。ExtJs初期仅是对Yahoo! UI的对话框扩展,后来逐渐有了自己的特色,深受网友的喜悦。发展至今,Ext除YUI外还支持Jquery,Prototype等多种JS底层库,让大家自由地选择。该框架完全基于纯Html/CSS+JS技术,提供丰富的跨浏览器UI组件,灵活采用JSON/XML数据源开发,使得服务端表示层的负荷真正减轻,从而达到客户端的MVC应用(卫军等,2009)。

## 2.3 三层架构

在软件体系架构设计中,分层式结构是最常见,也是最重要的一种结构(杨湧等,2008)。通常意义上的三层架构就是将整个业务应用划分为:表现层、业务层和持久层(图2)。区分层次的目的是为了实现在“高内聚,低耦合”的结构特点。

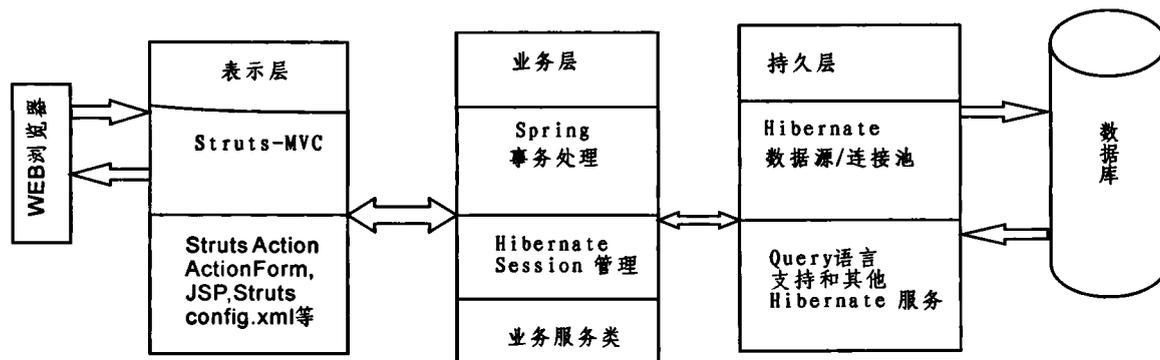


图2 三层架构图

Fig.2 picture of three-tier architecture

(1)表示层用于显示数据和接收用户输入的数据,为用户提供一种交互式操作的界面。

(2)业务层是针对具体问题的操作,也可以说是对数据层的操作,对数据业务逻辑处理。

(3)持久层所做事务直接操作数据库,即针对数据的增添、删除、修改、更新、查找等。

从图2中可以看出Struts作用在表示层。ActionFormBean接收网页中表单提交的数据,然后通过Action进行处理,再转发到对应的网页。Hibernate主要负责持久层的工作,以对象的形式对数据库进行增加删除修改以及用其特有的Query或其它查询语言进行查询操作。采用Hibernate作为持

久层技术的最大的好处在于:可以完全以面向对象的方式进行系统分析、系统设计。Spring 负责业务层管理,为表示层提供统计的调用接口,封装持久层的 DAO,与 POJO 进行无缝的链接。此外, Spring 同时对这三层进行一个无缝的良好的整合。而这种功能的实现也正是 Spring 这种框架核心思想——控制反转的一种体现。

### 3 系统分析

#### 3.1 业务流程的解析

系统内各种业务不尽相同,以烟草专卖商户管理为例。当业务人员录入一个烟草专卖商户时,输入其身份证号时,系统会调用人口库进行查找。若确有其人,则调出此人的基本信息给该业务人员查看,以便其继续录入烟草商的其它信息。若无此人,则将其存储于人口临时表中。具体业务流程如图 3。当信息录入完毕后,数据经前台 form 表单转到由 Struts 控制的视图层。在视图层,由 MyEclipse 提供的 Hibernate 反转机制生成的并与数据库关联的 POJO(plain object Java object)类 ToNoCardSalerInfo 将表单信息保存并带入业务层。业务层只关心数据业务的操作,对数据库的操作则是调用相应的 DAO 层来实现。DAO 层则是完全与数据库进行交互。当 DAO 层进行数据库的操作时,利用 Spring 提供的 HibernateDaoSupport 类管理 DAO 层,建立 DAO 层的实现类 ToNoCardSalerDaoImpl,最终由 Hibernate 本身的机制生成 SQL 语句,将数据保存到数据库中。

在录入烟草专卖商户数据的同时,相关领导既可查看所有烟草商户的信息,也可通过不同的查找条件,找出某个或某些商户的信息。查找的条件也经表单进入视图层、业务层,最终进入 DAO 层。由 Hibernate 本身特有的 HQL 语句,将结果查询出来。生成的结果保存在相应的集合中,传入前台,显示在列表中。

#### 3.2 实现技术及其分析

##### 3.2.1 实现技术

业务层是 Spring 通过控制反转将二者的接口注入至 Struts 的 Action 中。在 Spring 的配置文件 applicationContext.xml 中,在 Action 中注入业务层的代码如下:

```
<bean name = "/tobacco" class = " com. zongZhi.
```

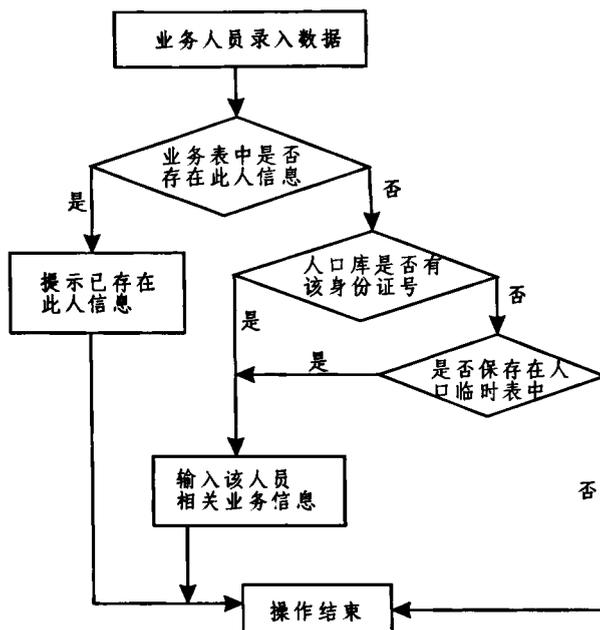


图 3 业务流程图

Fig.3 picture of Business Process

```
zhiAn. action. TobaccoAction" scope = " prototype" >
  < property name = " toNoCardSalerInfoAgentI"
  ref = "toNoCardSalerInfoAgentI" / >
</bean >
```

而业务层当中要调用了 DAO 层,即在业务层中也注入了 DAO 层的代码:

```
<!-- 配置业务层-->
< bean id = " toNoCardSalerInfoAgentI" class = "
com. zongZhi. zhiAn. agent. agentImpl. ToNoCard-
SaleInfoAgentImpl" >
  < property name = " toNoCardSalerInfoDaoI" ref
= "toNoCardSalerInfoDaoI" / >
</bean >
```

在 DAO 层中,由于 DAO 层继承了 HibernateDaoSupport 类,使用 getHibernateTemplate() 方法对数据的增删改查等操作。因此在 Spring 的配置文件 applicationContext.xml 中, DataSource 和 SessionFactory 在配置完毕后又注入到 DAO 层中。而 DataSource 当中的 hibernate. cfg. xml 文件则是 Hibernate 对 JDBC 驱动的封装。

```
<!--配置 sessionFactory -->
< bean id = " sessionFactory" class = " org. spring-
framework. orm. hibernate3. LocalSessionFactoryBean"
>
  < property name = " configLocation" >
```

```

    < value > classpath:hibernate.cfg.xml </
    value >
  </property >
</bean >
<!--配置 DAO 层-->
<bean id = " toNoCardSalerInfoDaoI" class = " com.
zongZhi.zhiAn.dao.daoImpl.ToNoCardSalerDaoImpl"
>
  <property name = " sessionFactory" ref = " ses-
sionFactory"/ >
</bean >

```

### 3.2.2 技术分析

从配置文件中可以看出,程序中的依赖关系为:Action 依赖业务层,业务层依赖 DAO 层,DAO 层依赖 SessionFactory,SessionFactory 依赖 DataSource。Spring 能有效地组织 J2EE 应用各层的对象。不管是控制层的 Action 对象,还是业务层的 Service 对象,还是持久层的 DAO 对象,都可在 Spring 的管理下有机地协调、运行。Spring 将各层的对象以松耦合的方式组织在一起,Action 对象无须关心 Service 对象的具体实现,Service 对象无须关心持久层对象的具体实现,各层对象的调用完全面向接口。当系统需要重构时,代码的改写量将大大减少。

从中可以看出,Spring 贯穿表现层、业务层、持久层,以高度的开发性与它们无缝整合。这体现了 Spring 的一种非常重要的思想——控制反转 (Inversion of Control),也称为依赖注入 (Dependency Injection)。

IOC 的思想来源于一种好莱坞式的原则——“Don't call us. We will call you.”即“不要来找我们,我们会主动来找你的”。IOC 充分的体现了“好莱坞”原则。其核心思想在于提供一个更加简单的机制来规定组件之间的依赖关系(一般涉及到对象的合作),并且在它们生命周期中对依赖关系进行管理。IOC 提供了这样的服务,使一个组件能够在它的生命周期中访问它的依赖和服务。当然,IOC 思想的前提是养成面向接口编程的习惯,而非面向实现类。

IOC 是一种设计模式,系统中通过引入实现了 IOC 模式的 IOC 容器,即可由 IOC 容器来管理对象的生命周期、依赖关系等,从而使得应用程序的配置和依赖性规范与实际的应用程序代码分开。其

中一个特点就是通通过文本的配件文件进行应用程序组件间相互关系的配置,而不用重新修改并编译具体的代码,使得业务组件可复用,也降低了业务和框架的耦合。

可以把 IOC 模式看做是一个升华的工厂模式,把 IOC 看作是一个大工厂,只是这个大工厂里要生成的对象都是在 XML 文件中定义的,然后利用 Java 的“反射”编程,根据 XML 中给出的类名生成相应的对象。从实现来看,IOC 是把以前在工厂方法里写死的对象生成代码,改变为由 XML 文件来定义,也就是把工厂和对象生成这两者独立分隔开来。这么做的目的就是提高灵活性和可维护性 (Walls et al., 2006)。

在小型 Web 应用当中, JDBC 就已经很适用了。而 SSH 三层架构并不是十分适合此类项目。因为 SSH 是以适当牺牲运行效率的代价,来换取程序的可扩展性以及可维护性。因而越大型越复杂的项目,SSH 就越适合。而 Spring 当中的 IOC 思想,则更加证明了这点。因为它使得主体与客体的关系由直接改为间接,且增加一个第三方来直接控制主体和客体之间的关系。

## 4 结束语

社区管理信息系统以 Spring 为核心,并通过 Spring 对 Hibernate 与 Struts 进行依赖注入,将两者很好地集成在 Spring 中。SSH 框架的综合使用,使得开发人员可以编写简洁明了的代码。在大型 Web 系统项目中,合理利用各种框架技术能够有效的减少系统各模块的耦合性。采用框架技术,不但可以方便大型项目程序员之间的分工协作,减少程序代码的耦合性,还可以促使开发人员养成良好的开发习惯,培养团队协作能力。

### 参考文献

- 湛湘倩,狄文辉,孙冬. 2009. 基于 SSH 框架与 AJAX 技术的 Java Web 应用开发[J]. 计算机工程与设计, 30 (10): 37-42.
- 董洪彬, 袁延平. 2004. 利用 Hibernate 的 J2EE 数据持久层的解决方案[J]. 计算机工程, (30): 36-39.
- 高昂, 卫文学. 2005. 基于 Hibernate 与 Struts 框架的数据持久化应用研究[J]. 计算机应用, 25(12): 2817-2819.
- 孙卫琴. 2005. 精通 Hibernate: Java 对象持久化技术详解[M]. 北京: 电子工业出版社: 10-13.
- 卫军, 夏慧军, 孟腊春, 等. 2009. ExtJS Web 应用程序开发指南[M]. 北京: 机械工业出版社: 9-14.

杨湧,付江辉,李丽华. 2008. 基于 MSP430 的嵌入式智能防盗监控系统的设计[J]. 东华理工大学学报:自然科学学报,32(12): 192-196.

电子科技(1):34-36.

Walls C, Breidenbach R. 2006. Spring in Action 中文版[M]. 北京: 人民邮电出版社:76-80.

朱庆伟,吴宇红. 2004. 一种对象/关系映射框架的分析和应用[J].

## Design of Community Management Information System Based on SSH Framework

YANG Zhong-ming

(Electronic Engineering, East China Institute of Technology, Fuzhou, JX 344000, China)

**Abstract:** SSH applied in cases involving much business in the large-scale enterprise development. For some small projects and simple application, which use JSP and Servlet technologies will be enough to complete. But for medium and large projects, their needs change frequently, complex business processes, required to maintain high performance. According to these features the design of management information platform for the community, and from that layer, business layer and data layer are discussed and the key technology. The background of the system design of Struts and Hibernate through Spring on the integration of complete, front Web page using Ajax technologies with the js framework Ext completed. Results fully verify the SSH on the scalability of such systems, and maintainability, and more certainly the SSH application for the applicability of such projects.

**Key Words:** SSH; enterprise development; scalability; maintainability; adaptability