基于 SSH 的渠道管理系统的设计与实现

沙先军[1],王爱平[2],魏博诚[1]
(1.安徽大学 计算机学院,安徽 合肥 230039;
2.安徽大学 计算与信号处理教育部重点实验室,安徽 合肥 230039)

摘 要:结合电信行业渠道管理信息化建设的现状和需求,采用基于 Struts+Spring+Hibernate(SSH)的轻量级分层技术架构的解决方法,设计和开发了一套 B/S 架构的渠道管理系统。实验结果表明,SSH 框架的使用不仅简化了系统的开发过程,而且提高了系统的可扩展性和可维护性。

关键词:Struts;Spring;Hibernate;耦合;渠道管理系统
中图分类号:TP311 **文献标识码**:A **文章编号**:1674-7720(2011)04-0089-04

Design and implementation of SSH-based channel management information system

Sha Xianjun[1], Wang Aiping[2], Wei Bocheng[1]
(1.Computer Science and Technology School, Anhui University, Hefei 230039 China;
2.Key Lab.of Intelligent Computing & Signal Processing, Ministry of Education, Anhui University, Hefei 230039, China)

Abstract: Combining with the status and requirements for the information construction of channel management system of telecommunications industry, this paper uses layer technology of lightweight architecture for Struts, Hibernate and Spring(SSH), the B/S channel management information system is designed and developed. The findings show that the application of the SSH framework can not only simplify the development of Java web system, but also improve the expansibility and maintainability of the system.

Key words: Struts; Spring; Hibernate; coupling; channel management system

渠道是电信行业直接面向客户进行营销、销售和服务的载体,随着电信行业竞争格局的形成和运营形式的多样化,已经有越来越多的渠道加入了电信行业。渠道的作用功不可没,不仅发展了电信行业而且还为电信运营商降低了运营成本,但目前渠道管理仍然面临着不少问题。首先由于历史原因,渠道信息系统版本较多,维护和管理比较困难;其次系统流程不畅;而且由于系统功能不健全,数据不全面,不能为业务部门提供全面和准确的支撑报表统计功能。本文基于这样的业务需求驱动,采用能够快速开发出跨平台、可重用、可扩展、分布式系统的 SSH 架构,设计实现了一个能够较好地适应业务需求变化的渠道管理信息系统。

1 SSH 技术集成框架

使用可重用的、成熟稳定的框架可以构建健壮的、可重用的、可扩充的、易维护的 Web 应用程序。目前,Java 开源世界里有不少优秀的框架,本文选用目前业界主流的 Java 开发体系 Struts+Spring+Hibernate 实现整个系统。从图 1 可以看出 SSH 集成架构主要由页面表现层、业务逻辑层、数据持久层构成,并且各层之间通过域对象[1](Domain Objects)做为载体进行通信。

1.1 页面表现层

页面层由 Struts 实现,Struts 是基于 MVC 模型的框架。MVC 模型将一个 Web 应用分割成为模型(Model)、视图(View)和控制器(Controller)三个部件,这三个部件既相互独立又能协同工作,通用的控制组件 ActionServlet 接收来自客户端的 HTTP 请求,根据 Struts-config.xml 配置文件,把请求转发给相应的 Action 对象,然后 Action 类实现业务逻辑和动作处理,通过流程跳转将处理结果返回给客户端[2]。

1.2 业务逻辑层

由 Spring 实现业务组件的组装关联与管理,Spring 是个流行的轻量级容器,是一个开源的并且普遍兼容的非强制性的框架。它通过 IoC[3](Inversion of Control,又称 DI,Dependency Injection)、AOP(Aspect-Oriented Programming)

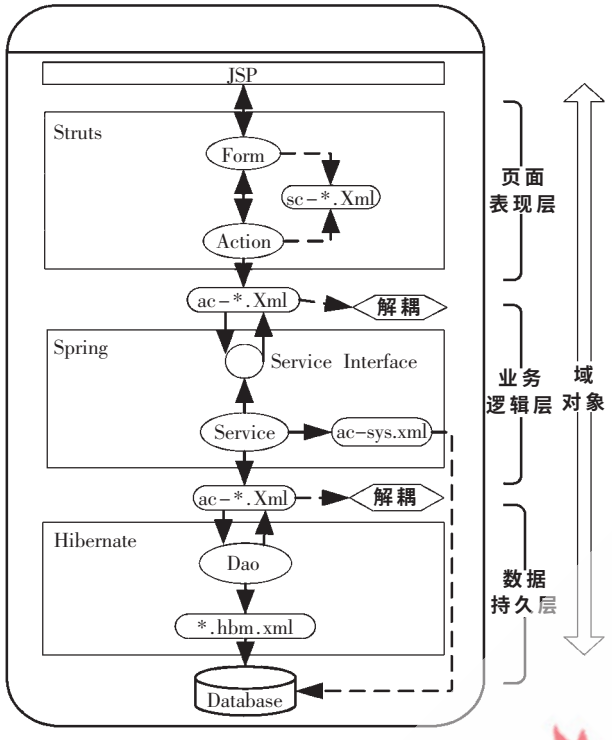

图 1 SSH 集成架构

的应用、使用面向接口的编程最大限度地降低业务组件之间的耦合度，增强系统兼容性和可扩展性。

1.3 数据持久层

借助开源框架 Hibernate 对 JDBC 进行轻量级的对象封装，将数据库表与对象进行关联，实现数据库访问性能优化和与数据库交互的常用操作 CRUD。Hibernate 封装了数据库访问、事务管理、数据缓存等工作，可以大大提供开发效率。

将这三种技术有机结合起来构建的 SSH 技术框架，不但可以有效提高系统开发效率，而且在系统安全性、稳定性和健壮性上都有良好的改进。域对象在各个层之间移动，为表示层提供所需要的数据源，为持久层提供对象，使得各个层以一种松散耦合的方式彼此作用而无需考虑低层的技术细节，进而构建出一个完整的 Web 开发框架。

2 系统简介及分层实现

2.1 系统简介

渠道是电信运营商与客户进行交互的具体途径，是向客户销售产品并提供差异化服务的载体。渠道管理主要承载渠道运营过程中的管理支撑功能，包括面向渠道的规划建设和所涉及人员、费用等方面的基础管理功能，以及绩效考核、服务管理等辅助管理功能。最终通过系统达到为渠道业务提供服务、加强渠道管控水平、提高各类渠道商的素质及实力、培育营销渠道系统的核心竞争力，并引导各渠道商积极配合电信运营商推行相关市场政策。

图 2 为某省渠道管理系统的功能结构图，主要包括渠道规划建设管理、业务管理、费用管理、运营考核管理、积分管理、酬金管理、系统管理、资源管理、支撑服务管理、统计报表等十大功能模块。该系统服务器操作系统采用 Unix(solaris)，关系型数据库采用 Oracle，应用服务器根据资金投入选择了 Oracle 的 Weblogic。客户机的操作系统选用 Windows2000、WindowsXP 等版本，安装 IE6.0 以上的浏览器。

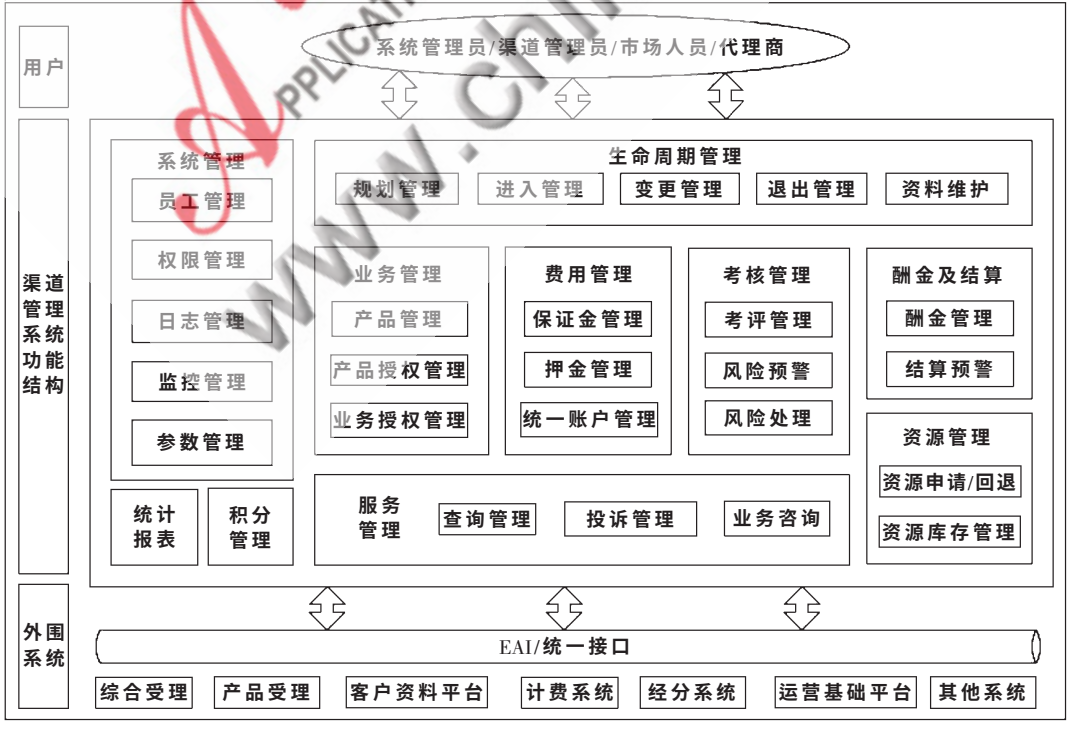

图 2 渠道管理系统功能结构

2.2 SSH 架构分层实现

2.2.1 数据持久层的搭建

持久层 Persistence 主要完成数据的访问，它操作底层的数据库表，使用 DAO 组件封装具体的操作细节，为业务层提供接口，使业务逻辑与数据持久化分离。在 Hibernate 中，通过配置相应的 XML 文件（*.hbm.xml）来完成对象与表、对象属性与表字段的“O/R 映射”关系。Hibernate 运行时，会自动读取 XML 映射文件，然后按照该文件指定的规则动态构建 Java 类，以便管理数据在数据库与 Java 程序之间的转换。

持久层的访问通过 DAO 组件完成，下面以渠道组织机构信息表(Dchngroupmsg)为例介绍 DAO 组件建立的过程。

首先使用自动化工具生成 JAVA 的 VO(Value Obects)对象 DchngroupmsgVo 和 *.hbm.xml 映射文件。Dchngroup-msgVo 类的属性与映射文件 Dchngroupmsg.hbm.xml 中的字段是一一对应的，它完成了对象与表、对象属性与表字段的“O/R 映射”。同时所有映射文件需要在 Spring 框架的 application-Context.xml 中配置，这样在 Hibernate 启动时才能根据该映射文件真正完成对象/关系的映射。

其次是 DAO 组件的实现与配置。DAO 组件继承了 HibernateDaoSupport 类，是 Spring 中整合了 Hibernate 的支持类，正是由于 Spring 对 Hibernate 的良好整合，调用 Hibernate 进行数据操作时只需要简单地继承 Spring 的 HibemateDaoSupport 类，然后在需要调用的方法中通过 getHibemateTemplate( )提供的方法就可以方便地操纵 Hibernate。

DAO 组件实现后，将它们配置在 Spring 容器中，让 Spring 容器为其注入 SessionFactory 的引用，并将 DAO 组件注入到业务逻辑组件中。通过这种依赖注入，可以提供应用各组件之间的良好解耦。

2.2.2 业务逻辑层的构建

在 SSH 架构中 Spring 是最核心的框架，Spring 主要应用于业务层来管理其他组件，充当了管理容器的角色。负责处理应用程序业务逻辑、业务校验和事务管理[4]；同时管理业务层的对象依赖；在表示层和持久层之间增加了一个灵活的机制，使得它们没有直接联系，借助 Spring 的 IoC、AOP 应用、面向接口编程，能降低业务组件之间的耦合度，增强系统扩展性。

构建 Spring 业务层主要完成以下两方面的任务：

（1）对 Spring 容器进行初始化与配置：Spring 提供一个 ContextLoaderListener 类用作 Spring 容器的初始化。Spring 容器初始化之后，需要创建 ApplicationContext 实例，Spring 有两个核心接口 BeanFactory 和 ApplicationContext，其中 ApplicationContext 是 BeanFactory 的子接口，增强了 BeanFactory 的功能，提供系统架构服务。

（2）业务逻辑实现：使用面向接口的编程，调用持久层定义好的接口为表示层提供业务接口，而无需关心接口的具体实现细节，先定义业务层的接口 SGroupMsgSvcI：

接口 SGroupMsgSvcI 的实现类 SGroupMsgSvcImpl：

```
public class SGroupMsgSvcImpl extends BaseService
                                    implements SGroupMsgSvc
{
    public DchngroupmsgVo getDchngroupmsgVo(String id)
                                                throws Exception
    {
        DchngroupmsgVo gmvo = new DchngroupmsgVo();
        DchngroupmsgDAO gmdao =(DchngroupmsgDAO)
        this.getBean("dchngroupmsgdao");
        return  gmvo = gmdao.get(id);
    }
}
```

以上的业务逻辑很简单，只是通过简单的调用 DchngroupmsgDAO 的接口方法来完成。DchngroupmsgDAO 引用是在前面持久层中配置的，这个配置指示 Spring 去动态注入 DchngroupmsgDAO 到 SGroupMsgSvcImpl 中，实现渠道组织信息的获取。

2.2.3 页面表示层的建立

Struts 实际上是 Servlet 技术的一个扩展，它用一个 ActionServlet 来接收浏览器的请求，用于系统的集中控制，然后在相应的 Action 类中调用业务逻辑，最后进行流程跳转。

表示层的建立主要是使用标签编写 JSP 页面，定义 Struts 的 Action 类及相应的配置文件。首先要定义 ActionForm，用于收集 JSP 页面传来的数据，供 Action 中调用业务逻辑使用。为了使用 Struts 的 Validator 框架来做客户端的表单验证，ActionForm 继承了 org.apache.struts.action.ActionForm 的子类 ValidatorForm。ActionForm 中定义的成员名称要与 JSP 页面表单中的域名称一致，这样在提交数据的时候，Struts 会自动把表单中的数据封装到继承的 ActionForm 中，避免了以往用 request.getParameter 获取参数的繁琐。

然后编写 Action 类，所有的 Action 类都继承了自定义的 BaseAction 类，BaseAction 类是 org.apache.struts.actions.DispatchAction 的子类，同时在 BaseAction 类中实现了一些公共方法，例如令牌验证判断是否重复提交、生成下拉列表、创建上下文 ApplicationContext 实例等。Action 类从 ActionForm 中提取数据，调用业务逻辑，然后根据返回结果转向相应的页面。

```
public class SGroupMsgAction extends BaseAction
{
    public ActionForward queryGroupMsg(ActionMapping
                                                mapping,
        ActionForm actionForm, HttpServletRequest request,
        HttpServletResponse response)throws Exception{
        SGroupMsgSvcI groupMsg_chn = (SGroupMsgSvcI)
                                this.getBean("SGroupMsgSvc");
        String group_id = request.getParameter
```

```
                                        ("GROUP_ID");
        DchngroupmsgVo dvo = groupMsg_chn.getDchn-
                                    groupmsgVo(group_id);
        ActionForward forward = mapping.findForward
                                        ("SUCCESS");
        return forward;
    }
}
```

所有的 Struts Action 类都继承自 BaseAction，基类 BaseAction 完成 Spring 上下文 ApplicactionContext.xml 的加载，提供一个公共的服务定位器方法 getBean()，这里 SGroupMsgAction 是 BaseAction 的子类，继承了父类的 getBean 方法，所以只要通过传入参数“SGroupMsgSvc”即可查找 Spring 的 Bean 资源，“SGroupMsgSvc”正是前面业务层在配置文件中指定的 bean。这样根据给定 bean 的 id 就能返回配置文件中指定的类。

Struts 的控制器 ActionServlet 接收用户查询渠道信息的 URL 请求“/sGroupMsg.do?operate=queryGroupMsg”，根据该请求的 URL 查找 struts-config.xml 配置文件来决定该请求是否处理 SGroupMsgAction，SGroupMsgAction 的 queryGroupMsg 方法接收页面数据，通过服务定位器查找名为“SGroupMsgSvc”的 Bean 资源，返回业务类的接口 SGroupMsgSvcI，并以域模型 DchngroupmsgVo 对象为参数调用业务接口的 getDchngroupmsgVo 方法来处理业务逻辑，若查询成功则返回一个自定义逻辑名称“SUCCESS”的 ActionForward 对象，最后 ActionServlet 把流程转向 ActionForward 中定义的 JSP 页面（success.jsp），从而完成一次请求/响应过程。

至此，基于 Struts、Spring+Hibernate 框架开发的渠道组织机构查询功能开发完成。

本文所设计的 SSH 集成的 Web 开发框架基于良好的应用程序分层和成熟的开源项目，具有结构清晰、松散耦合、可扩展和可维护性好的特点，已在电信运营商的各个省份（如北京、湖南、安徽、山西、陕西、四川，黑龙江、新疆等）的渠道管理系统中得到了非常成功的应用。目前系统采用的 SSH 集成架构已成为最为理想和成熟的 J2EE Web 应用框架，而且这种开发模式将会被越来越多的程序员所接受，在实际开发中得到广泛应用。

参考文献

[1] 郝彬，陈朔鹰.利用框架技术构建 Web 应用.计算机工程与设计[J].2007,28(1):8-13.

[2] 谌湘倩，狄文辉，孙冬.基于轻量级 J2EE 框架的网络教学系统[J].计算机工程，2008，34(6):266-268.

[3] 林信良.Spring 2.0 技术手册[M].北京：电子工业出版社，2007.

[4] 李刚.整合 Struts+Hibernate+Spring 应用开发详解[M].北京：清华大学出版社，2007.

（收稿日期：2010-10-24）

作者简介：

沙先军，男，1985 年生，硕士研究生，主要研究方向：软件技术与数据库。