

# 基于 SpringSide 的可定制数据采集系统的设计与实现

廖 宁, 李振坤, 邓钊汉

(广东工业大学 计算机学院, 广东 广州 510006)

**摘 要:**当前计算机信息系统的根本出发点之一就是获取资源数据,而这些资源数据大多数都是非常灵活、多样的。为了适应待采集资源数据的多样性和灵活性,提出了一种采用可扩展架构技术,基于 SpringSide 架构的可定制资源数据采集系统的设计方案,介绍了该系统的数据库设计以及主要功能设计,以该系统其中一个小模块为例详细说明了在 SpringSide 架构下系统的实现过程。该设计方案已经应用在实际项目——E 平台实验动物资源库中,投入使用后的实际效果良好,并具有良好的灵活性、可扩展性和可维护性。

**关键词:**SpringSide; 可定制; MVC; Spring; Struts; Hibernate

**中图分类号:**TP274+.2

**文献标识码:**A

**文章编号:**1673-629X(2009)10-0234-04

## Design and Implementation of Customizable Resource and Data Acquisition System Based on SpringSide

LIAO Ning, LI Zhen-kun, DENG Zhao-han

(Faculty of Computer, Guangdong University of Technology, Guangzhou 510006, China)

**Abstract:** One of the current computer information system's essential starting points is gaining the resources and data, but most of these resources and data are very nimble and diverse. In order to adapt the multiplicity and the flexibility of the resource and data which needed to acquire, presents a design proposal of the customizable resource and data acquisition system which adopted a scalable architecture and based on SpringSide, and then introduces this system's database design as well as the major function design. Finally, take one small module of this system for example, explains the system's implementation process based on SpringSide. This design proposal already applied in the actual project—E—Platform Laboratory Animals Resources Library, after being operational the actual result to be good, and has good flexibility, extendibility and maintainability.

**Key words:** SpringSide; customizable; MVC; Spring; Struts; Hibernate

## 0 引 言

随着计算机技术、信息处理技术及网络技术的迅猛发展和不断普及,政府以及企事业单位在网络办公、信息的网络采集与反馈、信息智能处理等方面的需求也越来越大。各级政府(企业)在其信息化建设上都投入了大量的人力物力,各类信息系统应运而生,各自都有其自身的特点。而根据对大量信息系统的研究分析发现,大部分这些信息系统的根本出发点都是为了获取有效数据,然后再在其基础上进行更深层次的数据处理。所以,信息获取成为了各类信息系统都必须解决的问题之一。

而大部分的系统所需的信息都是多样化的,如果按照传统的电子表单方式,那每需要采集一类信息就得靠程序员以编写程序代码的模式来开发表单和获取信息。当表单的数据项不确定、表单格式经常变化时,继续采用这种方式就难于应付了,因此需要一种灵活的可定制资源数据采集系统。文中介绍了一种采用可扩展架构技术,基于 SpringSide 架构的可定制资源数据采集系统设计开发方案。

## 1 关键技术及系统架构

### 1.1 关键技术

SpringSide 是以 Spring Framework 为核心,它以 Ruby On Rails 的简约风格整合了 Java 当前众多的开源项目。Ruby On Rails 是一个使用 Ruby 语言写的开源网络应用框架,它是严格按照 MVC 结构开发的, Ruby On Rails 的简约风格即是努力使框架自身保持简

收稿日期:2009-02-24;修回日期:2009-05-27

基金项目:广东省自然科学基金资助项目(4107411)

作者简介:廖 宁(1983-),男,广东兴宁人,硕士研究生,研究方向为计算机网络与分布式系统;李振坤,教授,研究方向为软件项目管理、计算机网络与分布式系统。

单,来使实际的应用开发时的代码更少,并且使用最少的配置<sup>[1]</sup>。

SpringSide 采用主流的技术选型,最主要包括了当前较流行的 J2EE 轻量级开源框架,即 Struts, Spring, Hibernate 这三个开源框架(简称 SSH 框架)。在这个系统之所以不采用传统的 SSH 整合框架而采用 SpringSide 对 SSH 的整合,是因为 SpringSide 是由主流的 Java 框架以简约的方式组成,因此开发更加简单、方便和快捷。其中 SpringSide 与传统的 SSH 框架的主要区别如下:

(1) 在持久层中,Entity POJO 类是由 Hibernate Annotation 来注释 ORM 关系,而不是用 XML 文件来将 Entity POJO 类的属性和数据库表的数据项——映射。POJO 属性与数据库默认——映射,因此采用 Hibernate Annotation 所需的注释很少,配置文件也较少,便于后期维护。

(2) 在业务逻辑层,Manager 类采用 No Dao、No Interface 的紧凑模式,继承于强力的 HibernateEntityDAO 基类,并用泛型声明自己管理的 Entity Class。

(3) 在表示层,Struts Action 采用实用的新模式:其中 DispatchAction 的每个 Controller 响应一组相关操作;采用了 LazyValidatorForm 免去了 FormBean 的定义;页面采用 JSP2.0,并结合 Prototype.js、Dojo 以及 JSTL 来简化 Javascript 与 Ajax 应用<sup>[1]</sup>。

## 1.2 系统架构

整个系统架构如图 1 所示。

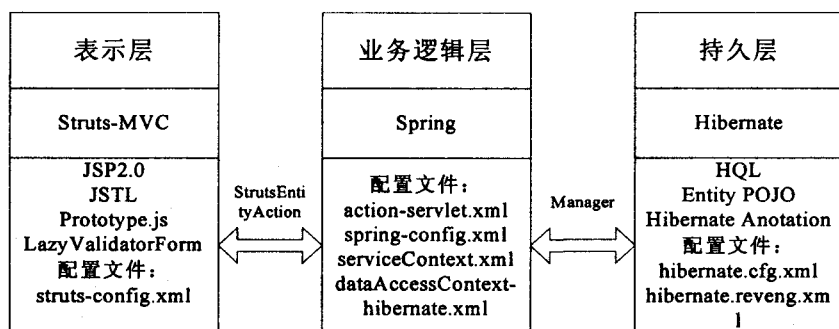


图 1 系统架构图

在这个架构中,Struts 致力于通过引用 MVC 模式<sup>[2]</sup>开发清晰明确的业务流程和用户交互,实现表现逻辑和业务逻辑的解耦。在 Struts 中模型是一个 StrutsEntityAction 类,开发者通过其实现商业逻辑。一个模型能为多个视图提供数据,这样就提高了可重用性。视图是由与控制器 Servlet 配合工作的一整套 JSP 以及 JSTL 标签构成,利用它们可以快速建立应用系统的界面。控制器主要是 ActionServlet,是模型与视图的联系纽带<sup>[3]</sup>。在 Struts 中对于业务逻辑的操作主

要由 Action、ActionMapping、Action—Forward 协调完成。Spring 提供了轻量级的管理业务组件的 IoC 容器,实现了对 EJB 技术的替代和更好的组件重用性<sup>[4]</sup>。Spring 根据配置文件,将其它对象的引用通过组件提供的 setter 方法进行设定。Spring 通过依赖注入机制,可以在运行期为组件配置所需要的资源,而无需在编写组件代码时就加以指定,从而在相当程度上降低了组件之间的耦合,实现了组件真正意义上的即插即用。而组件间依赖关系的减少也将极大改善代码的可重用性。这也是 Spring 最具价值的特性之一<sup>[5]</sup>。此外,应用的组件并不需要实现框架指定的接口,因此可以轻松地将组件从 Spring 中脱离,甚至无需任何修改,这在基于 EJB 框架实现的应用中是难以想象的。

而 Hibernate 作为数据库中间件使用,则实现了对关系数据库的对象化。它对 JDBC 进行了轻量级的对象封装,使 Java 程序员可以方便地使用对象编程思维来操纵数据库。Hibernate 不仅能够管理 Java 类(主要是 JavaBean,或者称之为 POJO)到 RDBMS 表的映射,还能够供应用查询和获取数据库中的数据。从开发效率上考虑,Hibernate 能够大大提高开发者使用 SQL 和处理 JDBC 与数据库交互的效率。目前,Hibernate 是事实上标准的 O/R 映射技术<sup>[6]</sup>。

## 2 系统设计

可定制资源数据采集系统最主要特点就在于可定制性,包括了可自定义数据项、自定义表单、可灵活设置数据项的显示格式以及计量单位等。要实现可定制性,其设计难点就在于系统的数据库设计方面。

### 2.1 数据库设计

本系统设计了一套编码规则,按照该编码规则对每一个数据项进行编码,并将所有数据项以该编码为主键保存在数据编码表里。数据编码表用于保存所有数据项的信息,包括各数据项的编码、数据项名称、数据类型、计量单位类别、所属类别、参考码表、备注说明等数据项的各种丰富特性。

表结构如表 1 所示。

表 1 数据编码表结构

编码	名称	数据类型	计量单位类别	所属类别	参考码表	备注说明
----	----	------	--------	------	------	------

其中数据项的编码规则如下:

数据项的编码用五层七位字符形式表示。其含义从左至右依次为:第一层为资源数据类别层,用一位字

母表示,如 A 表示为基本信息等;第二层为各个资源数据类别中的第一层类别,用一位字母表示;第三层为各资源数据类别中的第二层类别,用一位字母表示;第四层为各资源数据类别中的第三层类别,用一位字母表示;第五层:表示该数据项在以上四层类别下的顺序号,共三位数字,取值范围为 001~999。

除了数据字段编码表以外,该系统的数据库最主要的还有两个大表,一个是记录系统当前有哪些表单的系统表单信息表;另外一个表就是表单的数据项分配表,用来记录各个表单中含有哪些数据项的编码(该编码与数据编码表相关联),以及各数据项的计量单位及其表示样式等。而对于系统表单信息表中的每一张表单,在数据库中都有对应的一张表来保存该表单采集来的资源数据。

## 2.2 功能设计

本系统的主要功能主要是表单定制以及数据采集功能。其流程图如图 2、图 3 所示。

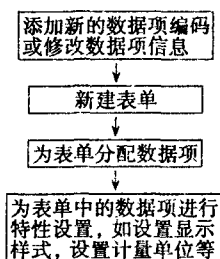


图 2 表单定制流程图

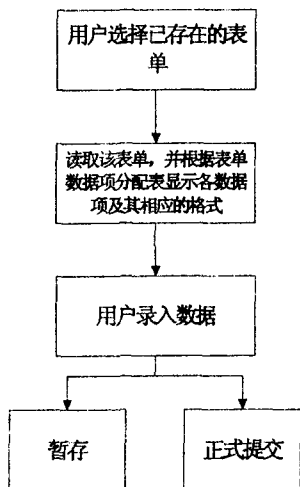


图 3 数据采集功能流程图

## 3 系统实现

下面以该系统表单定制模块中的数据编码表维护为例,说明在 SpringSide 框架下开发的过程。本系统是在 MyEclipse\_6.0.1 环境下开发,并需要下载并导入 springside-2.0-RC1-allinone 包以及安装 Hibernate Tools 插件。

### 3.1 持久层设计

首先在数据库中新建数据编码表 TBSJBM,然后用 Hibernate Tools 插件<sup>[7]</sup>直接连接数据库,选择数据编码表并生成带有 Hibernate Annotation 的 Entity POJO(即对应 TBSJBM 表的 JavaBean)。因为生成的 POJO 的默认类名与数据库表名,属性名与数据库列名相同,因此一般只需以 @Entity 注释类,以 @Id 和 @GeneratedValue 在 getter 上注释 id 列,表明主键及其生成策略即可。如下所示:

```

@Entity
@Table(name = "TBSJBM")
public class Tbsjbm implements java.io.Serializable {
    private String sjbm;
    private String sjmc;
    @Id
    @Column(name = "SJBM", unique = true, nullable = false, length = 8)
    public String getSjbm() {
        return this.sjbm;
    }
    public void setSjbm(String sjbm) {
        this.sjbm = sjbm;
    }
}
  
```

然后需要在配置文件 dataAccessContext-hibernate.xml 中为 hibernate sessionFactory 绑定该 POJO,即在 <sessionFactory> 节点的 <annotatedClasses> 节点增加 POJO,如下所示:

```

<property name="annotatedClasses">
<list>
    <value>com.labagd.eplatform.model.Tbsjbm</value>
</list>
</property>
  
```

### 3.2 业务层设计

在 SpringSide 框架下业务层不需要既编写 DAO 又为每个 DAO 编写接口,仅需要将两者合一,编写 Manager 类即可。Manager 类一般继承于基类 HibernateEntityDAO,该基类里已存在默认的 CRUD(增删改查)函数,因此子类只要用泛型语法声明自己管理的 Entity 类型即可。代码如下:

```

public class FieldCodingManager extends HibernateEntityDao<Tbsjbm> {
    private DataClassManager dataClassManager;
    public void setDataClassManager(DataClassManager dataClassManager) {
        this.dataClassManager = dataClassManager;
    }
    //...CRUD 以外的其它函数
}
  
```

在此可以将所有对数据编码表进行操作的函数都

写在 FieldCodingManager 中。如果 FieldCodingManager 要操作 Tbsjbm 外的其他 Entity POJO,则可以用 Spring 中的 setter 方法来注入另一个 Manager,如以上代码中的 dataClassManager,就是用来对数据类别表操作的 Manager。另外对每一个 Manager 类,均需在配置文件 serviceContext.xml 中定义,因为 Spring 2.0 作了统一事务配置<sup>[8]</sup>,加上 default-autowire 后,Manager 只需直接定义,代码如下:

```
<beans default-autowire="byName" default-lazy-init="true"
>
<bean id="FieldCodingManager"
class="com.labagd.eplatform.service.datamgr.FieldCoding-
Manager"/>
<bean id="dataClassManager"
class="com.labagd.eplatform.service.datamgr.DataClassMan-
ager"/>
</beans>
```

### 3.3 Controller 层设计

Controller 层接受来自浏览器的请求,并决定执行什么业务逻辑功能。本系统采用继承于 StrutsEntityAction 基类的 ActionServlet 类<sup>[9]</sup>来充当控制器。代码如下所示:

```
public class FieldCodingManagerAction extends StrutsEntityAction
<Tbsjbm, FieldCodingManager> {
    private FieldCodingManager fieldCodingManager;
}
```

FieldCodingManagerAction 中首先用泛型声明管理的 Entity POJO 是 Tbsjbm,即表示其为管理数据编码表的控制器,Manager 类型为 FieldCodingManager,再声明 fieldCodingManager 变量和它的 setter 函数,这样就可以拥有了默认的 CRUD 管理函数。

Controller 层的配置文件主要有两个,即 spring-config.xml, struts-config.xml。在 spring-config.xml 配置文件中,配置代码如下:

```
<!-- 管理后台的 Spring Servlet 配置文件 -->
<beans default-autowire="byName" default-lazy-init="true"
>
<bean name="/data/getField" class="com.labagd.eplatform.
web.datamgr.FieldCodingManagerAction"/>
</beans>
```

在 struts-config.xml 配置文件中,分别要进行 FormBean 定义以及 Action 定义,配置代码如下:

```
<struts-config>
<form-beans>
<form-bean name="FieldCodingForm" type="org.apache.
struts.validator.LazyValidatorForm"/>
</form-beans>
```

```
<action-mappings>
<action path="/data/getField" type="com.labagd.eplatform.
web.datamgr.FieldCodingManagerAction"
name="FieldCodingForm" scope="request" parameter="method"
validate="false">
<forward name="getFieldInfo" path="/data/getField.do?
method=getFieldCoding" redirect="true"/>
<forward name="showResult" path="/resdept/dataInput/re-
sultList.jsp"/>
</action>
```

按照以上配置代码, ActionServlet 在收到获取数据项编码的请求 getFieldInfo 后,然后由控制器 FieldCodingManagerAction 来负责调用指定的业务,即 getFieldCoding 函数,调用该函数以后再将查询结果返回到结果页面,即 resultList.jsp 页面。

## 4 结束语

根据信息系统中需要采集资源数据的多样性特点,提出了基于 SpringSide 框架的可定制资源数据采集系统的设计实现方案,在一定程度上解决了基于 Web 的数据采集系统会因需求的变化而不得不进行源代码级的修改问题。该方案已经应用在实际项目——E 平台实验动物资源库中,投入使用后的实际效果良好,并具有良好的灵活性、可扩展性和可维护性。

### 参考文献:

- [1] SpringSide 官方网站. SpringSide 2.0 概述[EB/OL]. 2007-05. <http://wiki.springside.org.cn/display/springside/Overview>.
- [2] 隋 永,周家纪. MVC 在 J2EE 框架中的应用研究[J]. 计算机技术与展,2006,16(12):125-127.
- [3] 孙卫琴.精通 Struts:基于 MVC 的 Java Web 设计与开发[M]. 北京:电子工业出版社,2005:383-403.
- [4] Walls C, Brdidenbach, Ryan. Spring in Action[M]. [s.l.]: O'reilly & Associates Inc, 2005.
- [5] 罗时飞.精通 Spring[M]. 北京:电子工业出版社,2005:29-48,191-224,303-309.
- [6] 孙卫琴.精通 Hibernate:Java 对象持久化技术详解[M]. 北京:电子工业出版社,2005:1-42.
- [7] Hibernate org. Hibernate Reference Documentation[EB/OL]. 2007. [http://www.hibernate.org/hib\\_docs/v3/reference/en/html/](http://www.hibernate.org/hib_docs/v3/reference/en/html/).
- [8] 范会联,张玉芳.基于 JSF + Spring + Hibernate 集成架构的 Web 信息系统设计[J]. 计算机技术与发展,2007,17(3):29-31.
- [9] Struts org. The apache software foundation[EB/OL]. 2008-05. <http://struts.apache.org/>.