

# 基于 Struts 与 Hibernate 框架的企业报表组件

甘寿全,袁道华,游强华,曾祥洪,彭俞霖

(四川大学 计算机学院,四川 成都 610065)

**摘要:**在企业级 WEB 应用中,会涉及到大量的报表,由于表对象的不同,模式层开发者须针对每个报表开发业务逻辑,使得代码复用率低,并且造成了极大的人力浪费。针对此问题提出并实现一个基于 Struts 和 Hibernate 框架的通用企业报表组件,自动生成企业报表。开发者只需给出查询的表对象、查询的条件、显示的类型(只读、可编辑),就能自动生成报表,不再需要针对报表编写任何的代码。

**关键词:**Struts; Hibernate; 企业报表; 组件

**中图分类号:**TP317.3

**文献标识码:**A

**文章编号:**1673-629X(2007)04-0136-04

## Enterprise Report Groupware Based on Struts and Hibernate Frame

GAN Shou-quan, YUAN Dao-hua, YOU Qiang-hua, ZENG Xiang-hong, PENG Yu-lin

(Department of Computer Science, Sichuan University, Chengdu 610065, China)

**Abstract:** There are a lot of different report forms in enterprise WEB applications. The developers working at the model layer have to develop business logic on each report form because of the different form objects. Obviously the reuse ratio of code is very low and it wastes much human resource. According to this problem this paper represents enterprise report form groupware based on Struts and Hibernate frame, which can build different report forms automatically. The developer only need give the searching object, searching condition and display format (read only, read/write), then it can build report form automatically. There is no need to write any code.

**Key words:** Struts; Hibernate; enterprise report form; groupware

### 0 引言

在面向对象开发方法成为主流的今天,开发企业级 WEB 应用时,基于 Struts 和 Hibernate 数据持久层框架的开发形式也越来越流行。开发企业报表时,通常做法是针对该报表编写一个 JavaBean,在该 Bean 中编写通过查询条件返回表记录对象 List 的方法。然后在报表所在 JSP 文件中嵌入一段 Java 程序,在该程序段中实例化报表的 JavaBean,并收集选择条件,然后调用该 JavaBean 中的查询函数,取出该报表所需显示的记录对象列表。使用 for 循环语句,根据记录的每列是什么 HTML 控件显示格式,编写对应的控件代码,并填充数据,使记录在页面上一条一条地显示出来。但在企业应用中,有大量的报表,如果还是按照通常的做法,针对每个报表开发业务逻辑,就使得程序开发者工作量非常大,而且开发的代码间复用率极低。在开

发企业报表过程中,笔者发现生成报表的实现方法在逻辑上是相同的,只是因为操作的表对象不同而造成了代码的不同,因此设想构建一个基于 Struts 和 Hibernate 架构的针对企业报表的通用组件,利用它,开发者只需给出查询的表对象、查询的条件、显示的类型(只读、可编辑),就能自动生成报表,不再需要针对报表编写任何的代码。

### 1 Struts 和 Hibernate 的框架体系结构简介

Struts 是一个基于 MVC 模式的应用架构,它清晰地系统划分成了控制器、模式和外观视图三个部分,让开发者遵循这个统一的模式进行设计编码。视图实际上就是 JSP 文件,模式由实现业务逻辑的 JavaBean 或 EJB 组件构成,控制器主要由 ActionServlet 和 Action 组成。

Action 类是整个体系的核心,它在客户请求、界面表示和业务逻辑之间起到一个桥梁的作用。每一个 Action 都用来处理某一项任务,或者进行一个业务操作<sup>[1]</sup>。Struts 虽然解决了模式、控制和视图层的分离,但是对数据持久层却没有提供灵活的框架支持。Hi-

收稿日期:2006-06-05

**作者简介:**甘寿全(1980-),男,四川富顺人,硕士研究生,研究方向为分布式处理与网络计算,数据库与信息系统等;袁道华,教授,研究方向为分布式处理与网络计算,移动计算,数据库与信息系统等。

Hibernate 作为当今非常流行的面向 Java 环境的 O/R Mapping(对象关系映射)框架,提供了 Java 类到数据库中表的映射和数据查询、恢复等机制,实现了数据层对象的持久性。它对 JDBC 进行了非常轻量级的对象封装,使得 Java 程序员可以使用面向对象编程思想来操作关系数据库,大大减小了操作数据库的工作量<sup>[2]</sup>。把 Struts 和 Hibernate 优势互补,把它们整合在一起,就可以实现真正意义上的 MVC 设计模式,控制器的功能依然由 ActionServlet 完成,它接收来自客户端的请求,把请求映射到相应的 Action。模式层由 Action、业务逻辑模块和 Hibernate 组成,Action 通过调用业务逻辑模块的接口,实现业务逻辑操作<sup>[3]</sup>。而业务逻辑模块则通过 Hibernate 穿过持久层映射到具体的数据表<sup>[4]</sup>。View 由 JSP 和 Struts 自定义标记组成。

## 2 企业报表组件 report 的设计和实现

传统的在 JSP 中嵌入 Java 程序段,取出记录列表,然后使用循环语句针对每条记录中列的显示形式编写相应 HTML 控件的办法可以实现报表的自动生成,也能很容易地编写出处理新增、删除、修改等业务的代码。缺点在于针对不同的表对象,都得建立返回值是该对象类型的记录列表的 JavaBean,以便通过这些记录生成报表。即开发企业报表时,只是因为操作对象的不同,开发者不得不开发了大量的业务逻辑。因此只要能实现需要操作的对象的统一定义,以及通过该变量和查询条件,从数据库中取出记录列表,就能根据报表每列的显示形式生成相应的 HTML 控件,也就实现了通用的企业报表的自动生成。在报表上实现新增、修改操作,很容易想到通过 formbean 传递数据,把请求提交到 Struts 的一个 Action,然后调用相应的业务逻辑来进行处理。但是 formbean 传递的是单条数据,而企业多数情况下都是进行的多条记录的插入和修改,使用这种方式显然不符合操作习惯,而且会因频繁操作数据库,影响系统性能。而选用 pageContext 来传递记录,能实现多条数据记录的值传递,而且一条记录数据实际上就是一个业务对象类的实例,由此想到可以把报表的行也抽象表示成一个类,这个类的属性就包含一个 HTML 控件,控件的值就是记录对应属性的值。

在报表上进行多条记录的插入,实际上就是添加了多个同样的 HTML 控件序列。因此可以采用这些控件的默认值,预先生成一条记录,专门用于新增操作。即用户提出新增请求后,触发 Javascript 函数,调用 HTML 所提供的“mergeAttributes”函数复制它,然后插入到报表中。通过这种方式,可以快速实现新增

操作,但数据并未提交到数据库,还得编写一个保存逻辑用于处理数据的提交。客户在页面上进行了多条记录的新增、修改操作后进行保存,保存业务逻辑在进行处理的时候就得分出哪些记录是新增的,哪些是修改的,哪些是没有变化的,以便调用 Hibernate 提供的 API 进行相应业务处理。为此,为每行数据设一个名叫“row\_tag”的状态标志,用“new”表示新数据,“old”表示没有变化,“old\_change”表示修改过的。同理,把要删除的“row\_tag”置为“del”。通过对每行记录设置状态标志后,在把数据提交到数据库的时候,就可以预先选择出新增对象、修改对象、删除对象,然后调用 Hibernate 的 API,进行批处理。

根据上面的设计思路,整个组件分成两大功能模块,即显示模块和业务处理模块<sup>[5]</sup>。

### 2.1 显示模块

创建 HTML 控件类。根据面向对象编程思想和设计方法,针对报表中可能出现的 input, select, textarea 等 HTML 控件分别创建具有它们属性的类,并增添 display 属性用于生成该字段的标题,增加 validate 属性用于验证输入数据的格式,增加 defaultValue 属性用于生成新增记录。然后创建一个控件工厂 eFactory 用于控制生成 HTML 控件,并把这些类封装到 view 包中。因为采用页面内容来传递数据,因此针对整个报表创建一个 PageData 类,它的属性包括显示的 HTML 控件列表 viewList、每页显示的记录条数 pagesize、当前页码 pageid 以及显示类型(只读、编辑)等页面信息。最为重要的,它定义了业务对象类 beanName、选择条件 where、排序条件 orderBy,以便利用它们从数据库中取出数据。

然后通过继承 net.sf.hibernate 的子包,构建一个 HibernateUtil 类,实现通过 session 操作数据库。在这个类中包含了 Hibernate 会话 session、Hibernate 语句 hql、数据源名字 DB\_NAME、数据库连接 conn、hibernate 事务 transaction 这些私有属性。同时还在其中定义了建立连接以及通过 session 对数据库进行查、插、删、改等操作的方法。

由于本组件是基于 Struts 的,业务逻辑的实现都会转发到 Action 再调用相应的逻辑处理,因此构建一个继承 Action 的动作类 DataAction。

还要构建一个专门用于报表业务的类 DataUtil,在其中引入了上面定义的 PageData 类、HibernateUtil 类和 eView 包,并定义一个 show 方法用于返回页面生成代码。在 show 方法中,首先调用从数据库取数据的 mappingData() 方法,该方法根据 PageData 传过来的 BeanName、Where、OrderBy 等条件,以及页面大小、页

数等信息,取出当前页的数据,代码如下:

```
.....
String hql = "from" + pageData.getBeanName(); //从页面类中取出业务对象,构成 hql 语句
if (! pageData.getWhere().equals(""))
    hql = hql + "where" + pageData.getWhere(); //添加条件
if (! pageData.getOrderBy().equals(""))
    hql = hql + "order by" + pageData.getOrderBy(); //添加排序条件
Query q = session.createQuery(hql); //执行查询语句
q.setFirstResult(pageData.getFromcount()); //设置 Hibernate 首个返回对象的位置
q.setMaxResults(pageData.getPagesize()); //限制 Hibernate 返回的对象数
java.util.List ls = q.list(); //得到查询对象列表
.....
```

然后根据取出的数据和 PageData 类中定义的 viewList 生成报表。在报表生成代码中,首先添加生成 dataform 的字符串。该 form 的 Action 为进行数据库更新的动作类 DataAction,使得用户提交页面数据后,ActionServlet 能正确转向 DataAction,进而进行相应的处理。接着添加一些隐藏的 input 控件,用于提交关键数据,比如当前页面的 url。根据要预先生成用于“新增”操作的考虑,添加一个隐藏的 EventTable,在其中使用 PageData 中定义的 eView 的默认值生成记录,同时在把该行数据的标志置位 new,以便在用户提交数据后根据该标志调用专门处理新增的业务逻辑进行处理。然后添加一个生成 ShowTable 的字符串,在其中根据在 PageData 中定义的 eView 列表和由 mappingData()方法得到的表数据,生成报表。最后添加用于在页面底部生成新增、删除、提交的 input 控件和触发业务处理的 js 函数名以及页面的页数信息、翻页控件的字符串,实现整个报表的显示。

## 2.2 新增、删除、修改业务处理模块

新增业务需要 js 函数和 JavaBean 共同来完成。Javascript 函数实现复制 EventTable 的一行记录插入到 ShowTable 中,JavaBean 实现把新增对象插入到数据库中。由于对每行记录设置了 row\_tag,可以很容易分辨出哪些记录是新增的、哪些是修改的,因而不需提交 ShowTable 中的所有记录,只需在进行数据库更新前,分离出新增、修改对象列表,然后调用 HibernateUtil 预先定义的新增、修改对象列表的函数进行处理。处理单表关键代码如下:

```
.....
int size = ParamUtil.getInt(request, "row_length", 0); //得到页面上的行数
final Class c = Class.forName(classname); //装载对象
final Method[] m = c.getDeclaredMethods(); //得到该 bean 中定义的方法
for (int b = 1; b < size; b++) {
    Object instance = c.newInstance(); //实例化
    row_tag = request.getParameterValues("row_tag")[b]; //得到行标志
    for (int i = 0; i < m.length; i++) {
        name = m[i].getName();
        if (name.indexOf("set") != -1) {
            String paramName = name.substring(3, name.length()).toUpperCase();
            paramName = classtag + "." + paramName; //根据 Hibernate 命名规则得到参数名称
            Class param = m[i].getParameterTypes(); //得到参数类型
            Object paramValue = new Object[1];
            if (param.getName().equals("java.lang.String")) {
                paramValue = request.getParameterValues(paramName)[num]; //根据参数类型取得数据
            }
            m[i].invoke(instance, paramValue); //调用 set 函数赋值
        }
    }
    if (row_tag.equals("new")) {
        newObj.add(instance); //把新增的记录添加到新增对象中
    } else if (row_tag.equals("old_change")) {
        oldObj.add(instance); //把修改过的记录添加到老对象中,以便更新数据库
    }
}
.....
```

同理,可以在 DataUtil 中编写删除业务逻辑。

因此整个组件包含一个 view 包、一个 PageData 类、一个 HibernateUtil 类、一个继承 Struts 的 Action 类 DataAction、一个 DataUtil 类和一个 js 文件 report.js。组件的系统框架如图 1 所示。

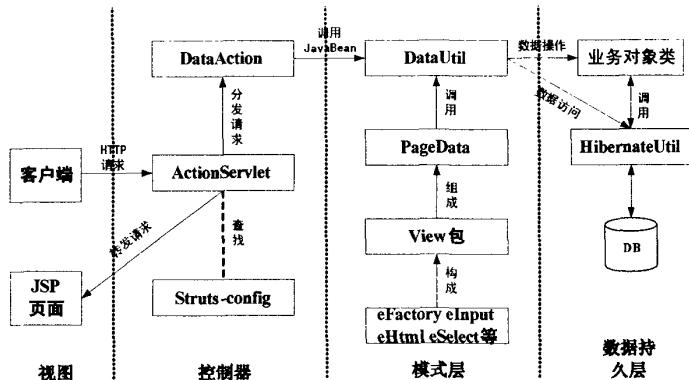


图 1 企业报表组件系统框架图

### 3 组件使用

建一个部门表 department, 具体字段如下:

department: dept\_id dept\_name remark

以这两个表来介绍该组件的使用流程:

(1) 搭建应用环境。

由于本组件是基于 Struts 和 Hibernate 框架的, 因此首先得下载 Struts 包及其相关配置文件, 并根据其配置规则创建一个 WEB 系统框架。然后下载 Hibernate 包, 创建 Hibernate 的配置文件 hibernate. cfg. xml 和 hibernate. properties, 因为 SessionFactory 通常只需建立一次, 所以可以利用 Struts 的 plugin 机制在系统初始化时建立它。

(2) 根据 Hibernate 的表映射规则, 编写 Department. hbm. xml, 然后建立对应的 Java 映射类 Department。

(3) 在 hibernate. cfg. xml 中建立数据库连接和 SessionFactory, 并申明上面的两个 XML 映射文件。

(4) 对于单张表 Department 进行报表生成列表形式, 如图 2 所示。

### 4 结论

根据面向对象编程思想和设计方法, 把企业报表抽象成一个统一的类, 而它又由 HTML 控件类所组成, 通过 JavaBean 与 js 文件的配合使用, 在 Struts 和 Hibernate 框架中实现了一个可以自动生成的通用企

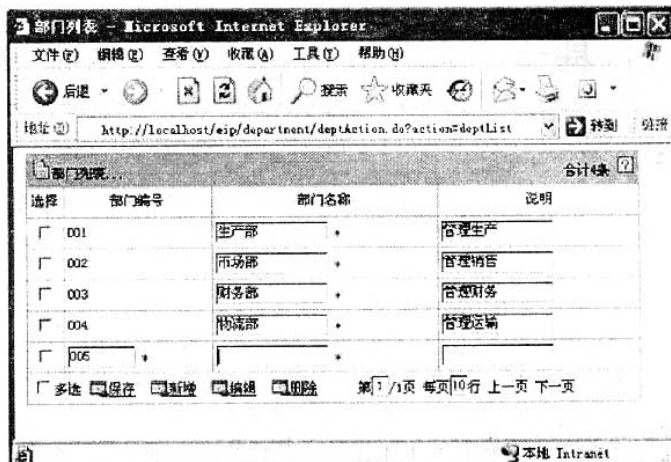


图 2 生成部门列表

业报表, 使得开发者在开发企业报表时, 只需申明需要操作的表的 Hibernate 对象类、页面显示字段、页面显示类型, 大大减小了工作量。

#### 参考文献:

- [1] 孙卫琴. 基于 MVC 的 JavaWeb 设计与开发[M]. 北京: 电子工业出版社, 2004.
- [2] 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.
- [3] 方巍, 孙涌, 张书奎. 整合 Struts 和 Hibernate 的 Web 系统应用[J]. 计算机与现代化, 2005(12): 39-41.
- [4] 高昂, 卫文学. 基于 Hibernate 与 Struts 框架的数据持久化应用研究[J]. 计算机应用, 2005, 25(12): 2817-2820.
- [5] 阎宏. Java 与模式[M]. 北京: 电子工业出版社, 2002.

(上接第 88 页)

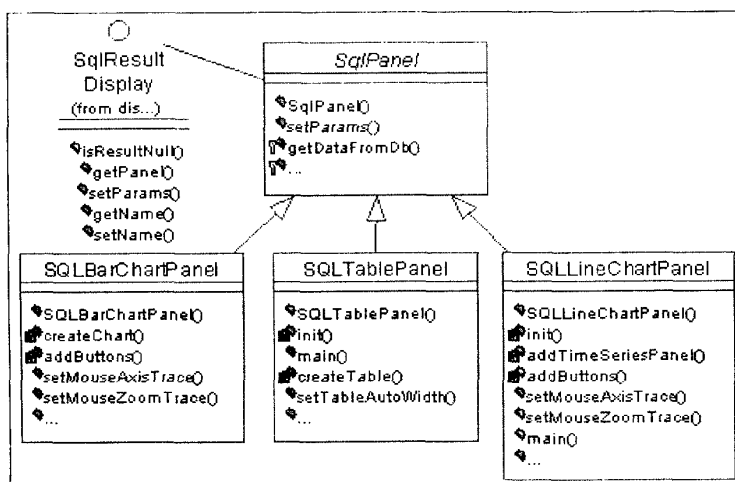


图 4 查询结果展现涉及的接口和类

的主要接口, 是基于树的 API, 它在内存中创建了文档树结构, 通过遍历这颗树来解析到 XML 中存储的数

据。文中着重介绍了 XML 解析在数据存储方面的应用, 对推广应用 XML 和 DOM 技术有一定的实用意义。

#### 参考文献:

- [1] 李勇军, 冀汶莉, 马光思. 用 DOM 解析 XML 文档[J]. 计算机应用, 2001(8): 103-105.
- [2] 邱俊. 如何有效利用 DOM 处理 XML 文档[J]. 重庆工学院学报, 2002(4): 38-41.
- [3] 朱前飞, 高芒. XML 解析技术研究[J]. 电脑开发与应用, 2004(11): 26-28.
- [4] 逮鹏. XML 文档解析技术及其应用[J]. 郑州大学学报, 2002(12): 17-19.
- [5] 孙霞, 程宏斌. 基于 JAVA 的 DOM 解析技术[J]. 计算机时代, 2004(7): 1-3.