

基于 Struts 框架的 Web 报表展示的设计与实现

Designing and Realizing of Web Report Form Based on Struts Framework

林 琴 谭骏珊 (中南林业科技大学 计算机科学学院 长沙 410004)

摘要: Struts 是目前非常流行的基于 MVC 的 Java Web 框架。Web 报表的开发,各个企业也提出了不同的解决方案。在 Web 报表上有效展示大量数据的问题,仍是值得关注的问题。为解决在 Web 页面上大量数据的展示问题,本文设计一种基于 Struts 框架的 Web 页面上数据展示的方案,该方案从优化表格元素和处理特定图像元素入手,有效提高了展示效率及效果。本文首先介绍了 Struts 的工作原理和结构,就 Web 报表展示中 Bean 的优化设计进行讨论,最后在 Struts 框架上对报表的 Web 展示进行实现。

关键词: Struts 框架 Web 报表展示 Bean

1 引言

报表在各个部门的应用非常广泛,近年来,随着数据仓库、OLAP 分析、数据挖掘技术的发展,报表的展示都成了每个企业产品的必要部分。Web 报表展示,各个企业都提出了各自的解决方案^[1],Web 报表产品层出不穷。然而,对于数据库中的海量数据,如何以一种高效的方式在 Web 页面上展示,仍是 Web 报表展示需要解决的问题。本文设计了在页面上展示大量数据的方法,同时对于 Web 的展示,如果以使用编写 JavaBean 并将其嵌入到 jsp 页面的形式,将使整个 jsp 页面存放大量的代码,不但不能提高整体效率还会增加维护的难度。本文从如何应用 Struts 框架对 Web 报表展示进行优化设计并实现进行讨论。

2 Struts 简介

现代软件开发对开发周期要求越来越迅短,如果要自己开发一套框架已经不太现实了。所以我们可以选择比较成熟的软件框架。而 Struts 对于 Java 的 Web 开发就是一个不错的选择,它实现了模型——视图——控制器 (Model——View——Controller) MVC 设计模式,其广泛的适应性和高度的扩展能力加上对生产效率的极大提高成为了目前 Java 的 Web 开发所逐步认可的标准^[2]。

2.1 Struts 实现的 MVC 框架

(1) 视图。视图是通过一组 JSP 程序实现的。这

些 JSP 程序中不能包含业务逻辑,也不能包括模型的信息,模型的信息是通过控制器传递的。在 Struts 中 ActionForm 也被看成是视图的一部分。

(2) 模型。模型是程序的状态表示,其状态包括:消息中传递的状态、共享状态或一次回话中的状态以及持久状态。操作这些状态的流程也同样包括在模型中,这些模型就是业务总体运作流程和数据。

(3) 控制器。控制器部分由 ActionServlet 和 Action 来实现。ActionServlet 也就是 Struts 框架用来和 Web 容器交互的接口,该类是 Struts 框架的核心组件,ActionServlet 继承了 HttpServlet 类,它在 MVC 模型中扮演中央控制器的角色,当 Web 容器一接收到客户请求就会交给 ActionServlet。该类根据 struts - config. xml 文件的配置信息选择合适 Action 来处理请求,并传递请求过程中的数据,这些数据是通过 ActionForm 传递的。Action 本身也并不包含业务逻辑的处理代码,它只是负责调用业务的处理代码的。

2.2 Struts 工作原理

Struts 继承了 MVC 的特性,并根据 J2EE 特点做了相应变化与扩展。图 1 是其工作原理图。由图可以看出,它实现了模型——视图——控制器的分离。

2.3 Struts 框架的优势

- (1) 良好的架构和设计;
- (2) 可重用,模块化,扩展性好;

(3) Open source。

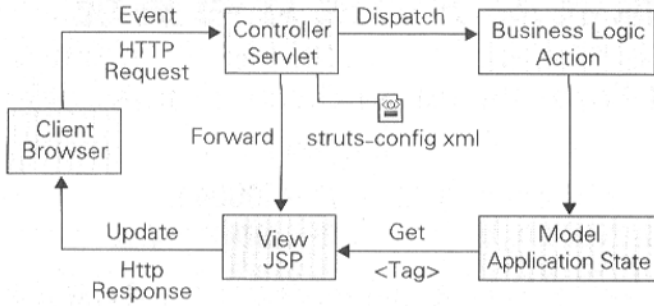


图 1 Struts 工作原理图

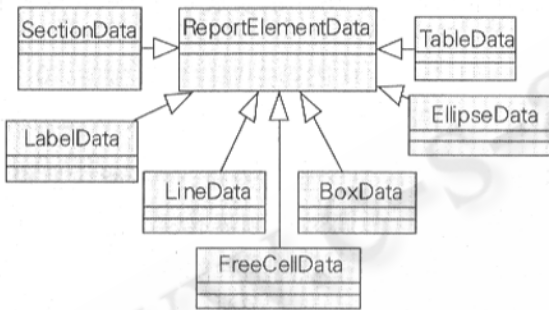


图 2 各数据结构的类图

3 Web 报表展示的设计

设计包括两部分内容：主页面报表元素展示设计和目录树结构设计。

3.1 主页面报表元素展示设计

数据库中存储的是各个数据结构的描述信息。不同的数据结构有着不同的特点，如样式、位置等，要求报表的 Web 展示就是根据这些数据结构的特点，从数据库里面调出相应的信息显示在页面上，这跟普通的 Web 页面显示是有区别，普通 Web 页面是自己定义页面的样式风格以及数据的显示方式，而报表展示则要求按照数据库存储的数据结构的信息原貌的显示在 Web 页面上。由于每种数据结构封装成一个类，这就要求通过编写 JavaBean，对类提供的方法的访问来访问数据库。因此，我们不需要知道数据库里的信息，也不要了解类是如何实现的，只需要调用类对外提供的方法就可以了。这些数据结构包括：表格、直线、椭圆、矩形、自由单元格、标签、节点，图 2 是各数据结构的类图^[3]。

图 2 中，ReportElementData 类是这些报表元素的

基类。设计一个 ReportDocGenerator 的 Bean，主页面调用这个 Bean 的输出信息，来完成页面元素显示。在 ReportDocGenerator 这个 Bean 中，通过判断 ReportElementData 类是属于何种具体的数据结构，来分别调用相应的数据结构处理单元，其形式如下：

```

if ( reportElementData instanceof SectionData ) {
    /* *
     * @todo 对 SectionData 进行处理。
     * /
}
else if ( reportElementData instanceof TableData ) {
    /* *
     * @todo 对 TableData 进行处理。
     * /
}
//略
    
```

我们这里对一些主要页面元素数据结构的处理作介绍：

(1) 表格元素的处理。在数据库中，表格元素 (TableData) 以单元格信息的形式存放，每个单元格都有四个边框，因此每个单元格包括边框的样式 (TdProperties) 以及显示的字体的样式 (FontProperties)，因此应该设计两个类把这些样式的属性都取出来，并生成一个样式表，嵌入到 jsp 页面便可以显示了。这里碰到一个设计的问题，数据库中的表格是很多的，每个表格就有大量的单元格，如果每个单元格都是生成一个边框样式表和字体样式表，那么整个生成的 jsp 页面代码将会非常的庞大，这样不仅消耗了大量内存还会大大降低了执行效率。为了解决上面的问题，我们进行了这样的优化设计：即设计一个 List 容器，用来存储单元格边框样式和字体样式，再用两个二维数组分别对应于表格中各单元格的位置序号，一个存放相应的边框样式在 List 中的序号，另一个存放字体样式在 List 中的序号。读取单元格样式的时候，先比较样式，把不相同的样式依依存放到 List 容器中。List 容器中先存放不同的边框样式，接着存放不同的字体样式。图 3 展示了其关系。

(2) 图形元素的处理。图形元素包括直线、椭圆、矩形。如何把数据库中的图形元素显示在页面中，我们认为一种是把图片打印到内存中然后显示出来；另

一种方法就是如普通页面设计那样,在要显示图形的地方添加图片地址。在微软的 Excel 中画图形,结束后保存为 .html 文件,发现跟随文件保存下来的还有一个 .files 文件,里面包括该图片文件和一些 .xml 的文件。我们采用微软的做法,通过编码生成图片,然后再插入到页面中。

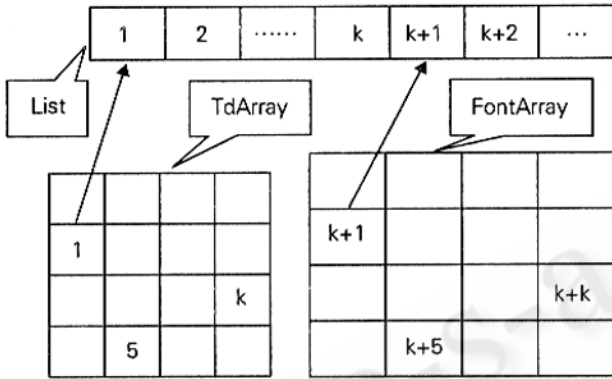


图 3 单元格样式存储关系图

在 Web 页面中,浏览器支持的格式有 JPG、GIF 和 PNG。PNG 是网景公司开发的支持新一代 WWW 标准而制定的较为新型的图形格式,它综合了 JPG 和 GIF 格式的优点,支持 24bit 色彩 (256 * 256 * 256), 压缩不失真并支持透明背景和渐显图像的制作,被称为传统 GIF 的替代格式。因此这里我们选择生成 PNG 图片。由于 Java 的图形处理得到的是一个 AWT 图像,必须依靠开源软件包 JAI 来对该图形文件按 PNG 格式编码。其代码如下:

```

OutputStream bo = new BufferedOutputStream
(out); // OutputStream 输出流
ParameterBlock pb = new ParameterBlock().add
(tag);
PlanarImage src = (PlanarImage) JAI.create("
awtimage", pb);
JAI.create("encode", src, bo, "PNG", null);
//PNG 格式编码
bo.close();
    
```

图片生成后,插入到页面中,还将碰到一个问题,那就是图片背景色透明度的问题。图片的背景色和页面的背景色不一致,影响了显示效果,这是由于我们采用 BufferedImage 类生成的图片是 RGB 模式的,它可以通过调节透明度来改变透明效果,但是却不能够完全透

明。RGB 模式要转换成索引模式才行,但是这个需要编码。我们在本次设计实现的时候,采用了一种简单可行的方法,那就是利用 CSS 中样式的 Chroma 属性,它可以设置一个对象中指定的颜色为透明色。我们可以在生成图片时,把图片的背景色设置为白色,然后在 jsp 页面中应用该属性,就可以轻松的过滤图片的背景色,而不影响显示效果。经过多次试验证明,不管页面背景色更改为什么颜色,显示的图片背景色都是透明的。代码如下:

```

<IMG src = "line.png" style = "FILTER: chroma
(color:#000000)" runat = "server" >
    
```

3.2 目录树结构设计

目录树设计采用了 Struts 框架。目录树结构^[4]中有三个对象 group, form, formdefinition, 表示目录树结构中的节点和叶子。group 对象是一个节点,它下面可能还有它的子 group 对象,而 form 对象是特殊的节点,它的下面不再有节点只有叶子,它的属性和 group 类似。Group 和 form 都有展开和合拢功能,Formdefinition 对象是叶子,点击它可以直接在主页面中显示前面设计的各元素了。

在设计中,我们共设计了 4 个以 .do 作为请求的服务名,他们分别对应四个 FormBean 和 ActionBean。表 1 是其功能描述。

这里限于篇幅,我们就对其中的 groupAction.do 服务,通过 UML 语言中的类图表示出来,见图 4。

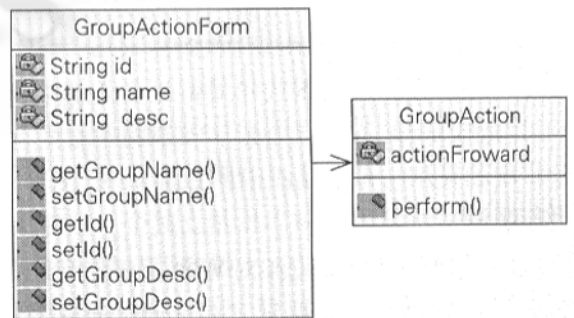


图 4 groupAction.do 服务的类图

所有的 *.do 请求均被指向 ActionServlet,每个服务都通过 Struts - config.xml 来配置,Struts 的核心是 ActionServlet, ActionServlet 的核心是 Struts - config.xml。这样,所有的消息传递都可以通过 Struts - config.xml 来进行管理配置,一目了然,方便了开发和后期的维护,与在 JSP 中嵌入 java 代码相比,提高了执行效率。

表 1 各服务功能描述

	FormBean	ActionBean	描述
formAction.do	formActionForm	formAction	完成 form 对象的消息处理
groupAction.do	froupActionForm	groupAction	完成 group 对象的消息处理
reportDcoAction.do	reportDocActionForm	reportDocAction	通过 formdefinition 对象的调用完成前面设计的页面元素的消息处理
resourceAction.do	ResourceActionForm	resourceAction	完成目录树显示消息的处理

我们通过如上设计,并在 Jbuilder 中开发,由 Tomcat 服务器发布^[5]。得到 Web 报表展示的某一页面如图 5 所示。

参考文献

- 1 李云亮、李相枢, NET 环境下两种 Web 报表解决方案的

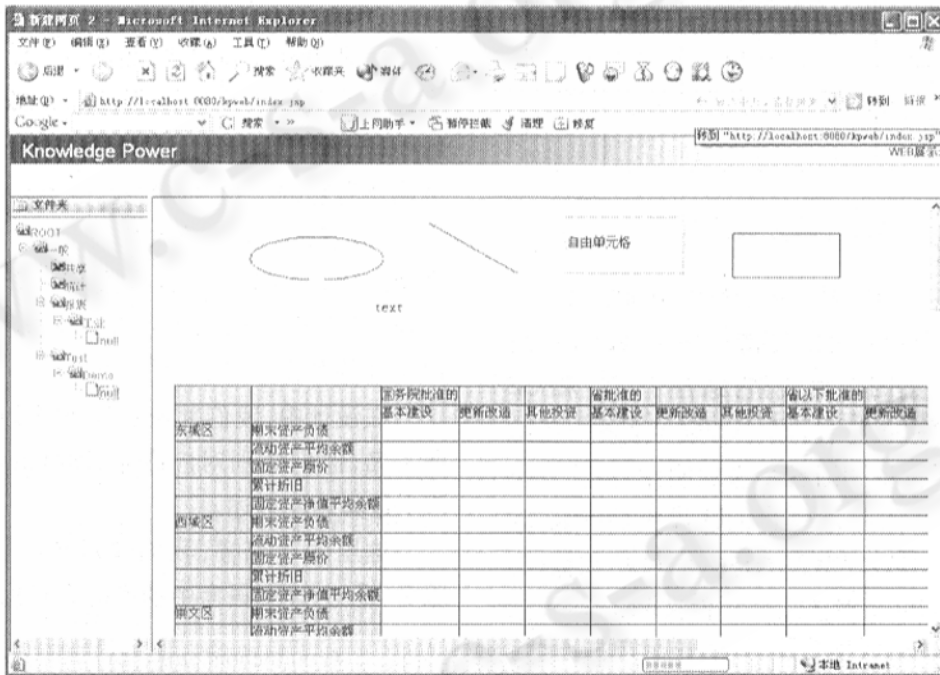


图 5 Web 报表展示图

4 结束语

本文通过利用 Struts 框架对 Web 报表展示进行优化设计与实现,并向大家展示了在设计过程中的若干问题。这个实现还只是一个模型,若真正应用到业务层面,在执行效率方面还可以有进一步的提升,主要通过添加 cache 缓存的单元,来有效提高访问速度。因此,接下来的工作应该是针对其实际应用做进一步的优化。

对比分析,计算机应用研究,2004,(06):212-214。

- 2 英若波译,实战 Struts[M],北京:机械工业出版社,2005。
- 3 GRADY BOOTH, JAMES RUMBAUGH, IVAR JACOBSON. The Unified Modeling Language User Guide [M]. MA: Addison Wesley Longman, 1999.
- 4 李树贤、范太华、韩永国, WEB 模式下借助树形目录实现动态报表[J]计算机与数字工程, 2005, (09):122-124。
- 5 孙卫琴、李洪成, Tomcat 与 Java Web 开发技术详解 [M],北京:电子工业出版社,2004。