

B / S 结构下复杂报表实现技术的研究

吴 雷, 袁兆山, 李 超
(合肥工业大学 计算机与信息学院, 安徽 合肥 230009)

摘 要: 提出了 B/S 结构下基于 XML 的报表实现方案, 解决了 Web 应用系统中报表样式不能灵活生成的问题。通过对实际项目中报表的结构进行分析, 将报表的组成为样式和数据两个部分, 并用自定义的标记按照 XML 规范描述出这两部分信息, 服务器端的解析器获得这两个 XML 文件后将报表信息解析出来并传给客户端进行显示。

关键词: B/S; XML; 报表; 样式; 数据

中图法分类号: TP311. 5 文献标识码: A 文章编号: 1001-3695(2006) 05-0083-03

Study on Complicated Report Technique in B/S Model

WU Lei, YUAN Zhao-shan, LI Chao
(School of Computer & Information, Hefei University of Technology, Hefei Anhui 230009, China)

Abstract: This paper puts forward a XML-based technique to design report in B/S model and solves the problem which report 's style cannot be designed flexibly in most Web application systems . By analyzing the framework of reports in the project, reports can be constituted by two parts : style and data , and they can be described by defined marks. Server gets these two XML-based files and parses them. At the end , client gets parsed information about reports and show it in the explorer.
Key words: B/S; XML; Report; Style; Data

随着 Internet 技术的兴起, B/S 结构已经成为目前开发 Web 应用系统时首选的体系结构, 基于 Web 的信息管理系统也更广泛地走进了企事业单位的日常工作中, 给这些单位的信息管理、统计分析工作带来了很大的方便。但是, 大多数信息管理系统的报表生成方式不灵活, 显示的报表格式固定、可复用性差, 一旦用户对报表样式的需求发生改变, 就会给软件维护带来许多不必要的麻烦。通过结合实际项目, 提出了 B/S 结构下报表灵活生成、打印问题的一个通用的解决方案。

1 基于 XML 的报表生成模型

1.1 报表分析

通过对项目中的报表样式进行分析, 可以将报表样式大致分为两大类: 一类是固定长度的报表, 这类报表所包含的数据记录数是确定的, 该类报表的典型实例为个人详细信息表等; 另一类是不定长度的报表, 这类报表所包含的数据记录条数是未知的, 报表的长度由其需要显示的数据记录的条数来决定, 该类报表的典型实例是各种类型的统计表、台账等。

每张报表中都有需要填充数据的空格部分, 在这里定义为表格域。报表中的表格域可以分为三类: 静态文字域, 这类表格中的文字是给定的, 不需要从数据库中读取; 动态文字域, 这类表格中的文字是动态生成的, 需要从数据库中读取; 斜线域, 这类表格带有斜线, 此类表格在报表中也很常见。

图 1 所示的就是不定长度的报表, 通过对该报表的结构进行分析, 可将该报表分为四个部分。这四部分在图 1 中已经被

标注出来, 从上到下依次为表标题、表头、表身、表尾。在这张报表中, 表身部分是不定长度的, 可以随着数据记录长度的变化而变化。对于固定长度的报表, 由于其不含有包含未知长度数据记录集的表身部分, 因此在描述表格时可将其表身部分省略, 看成由表标题、表头、表尾三个部分组成。

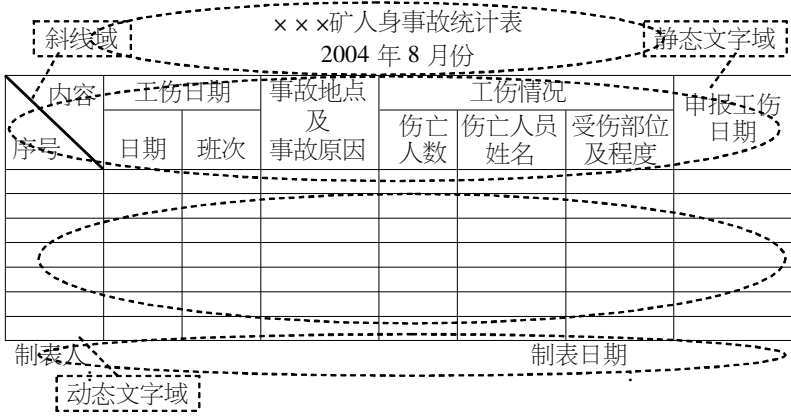


图 1 不定长度的报表

1.2 用 XML 语言描述报表

XML(eXtensible Markup Language) 是一种扩展性标识语言, 它将 SGML 的丰富功能与 HTML 的易用性结合到 Web 的应用中, 具有良好的数据存储结构、高度可扩展性、高度结构化, 而且便于网络传输。正由于其在这些方面的优越性, 决定了 XML 可以为报表的灵活生成提供合适的解决方案^[1]。

通过上面对报表结构的分析可以看出, 只需将报表的样式和动态数据信息描述出来就可以实现报表的显示, 因此, 在报表生成模型中定义了一系列的 XML 标记用来描述上述两种文件。描述报表样式时, 需要按照标题、表头、表身、表尾四个部分依次描述, 描述这四个表中的每一个时, 按照先行后列、从上到下、从左到右的顺序依次描述; 描述报表数据时, 也是按照描述

样式时所采用的顺序。需要说明的是,描述样式时需要清晰地描述表格域内文字的样式、表格域的大小,而数据描述不需要。

现在以不定长度报表为例介绍如何使用 XML 语言描述报表的样式。将组成报表的四大部分看作四个子表,标记分别定为 <titletable>, <headtable>, <bodytable>, <tailtable>,可以用高度 height,宽度 width,边框宽度 border 三个属性来描述这四个子表。对于每个子表,首先用标记 <row>描述行,然后从左到右按列依次描述行中的每个表格 <col>。在描述表格域时,可以用表格域类型 type,高度 height,宽度 width,所包含的行数 rownum,所包含的列数 colnum 等几个属性。最后要描述的是表格内的文本,用标记 <textvalue>表示。可以使用的描述属性有字体 font,字大小 fontsize,字颜色 fontcolor,水平对齐方式 align,垂直对齐方式 valign 等。这里需要说明的是,如果表格域是斜线类型,描述 <col>的属性时,需要增加一个斜线属性类型 slalinetype 用来说明是正斜线还是反斜线,并在 <textvalue>下增加两个子节点 <lefttextvalue>和 <righttextvalue>用来描述斜线左右的文本字段^[2]。

只描述报表样式是不够的,报表中还有动态数据部分,这些数据是从数据库中读取的。报表的数据描述文件与样式描述文件类似,由 < titledataset >, < headdataset >, < bodydataset >, < taildataset > 四部分组成。在描述各个部分时,仍先描述行 <row>,然后依次描述该行的每一列 <col>。

下面是报表样式描述文件和报表数据描述文件的 DTD^[3]:

```
<? xml version = " 1. 0" encoding = " UTF-8"? >
<! ELEMENT reportstyle titledataset?, headtable?, bodytable?, tailtable?) >
<! ELEMENT titletable ( row + ) >
<! ATTLIST titletable
height CDATA#IMPLIED
width CDATA#IMPLIED
border CDATA#IMPLIED >
<! ELEMENT headtable ( row + ) >
<! ATTLIST headtable
height CDATA#IMPLIED
width CDATA#IMPLIED
border CDATA#IMPLIED >
<! ELEMENT bodytable ( row + ) >
<! ATTLIST bodytable
height CDATA#IMPLIED
width CDATA#IMPLIED
border CDATA#IMPLIED >
<! ELEMENT tailtable ( row + ) >
<! ATTLIST tailtable
height CDATA#IMPLIED
width CDATA#IMPLIED
border CDATA#IMPLIED >
<! ELEMENT row ( col + ) >
<! ELEMENT col ( slantline?, textvalue?) >
<! ATTLIST col
type ( 0 |1 |2) #REQUIRED
height CDATA#IMPLIED
width CDATA#IMPLIED
colnum CDATA#IMPLIED
rownum CDATA#IMPLIED
<! ELEMENT slantline ( lefttextvalue?, righttextvalue?) >
<! ATTLIST slantline
slalinetype ( 0 |1) #REQUIRED >
<! ELEMENT lefttextvalue EMPTY >
<! ATTLIST lefttextvalue
fontsize CDATA#IMPLIED >
<! ELEMENT righttextvalue EMPTY >
```

```
<! ATTLIST righttextvalue
fontsize CDATA#IMPLIED >
<! ELEMENT textvalue EMPTY >
<! ATTLIST textvalue
align( center | left | right) " center"
valign ( top | middle | bottom) " middle"
fontsize CDATA #IMPLIED >
<? xml version = " 1. 0" encoding = " UTF-8"? >
<! ELEMENT reportdataset ( titledataset?, headdataset?, bodydataset?,
taildataset?) >
<! ELEMENT titledataset ( row + ) >
<! ELEMENT headdataset ( row + ) >
<! ELEMENT bodydataset ( row + ) >
<! ELEMENT taildataset ( row + ) >
<! ELEMENT row ( col* ) >
<! ELEMENT col EMPTY >
```

1.3 报表生成模型体系结构介绍

图 2 展示了报表生成模型的体系结构。数据库中有一张数据表专门负责存储业务中需要用到报表的样式描述文件。Web 服务器中的 EJB 是一个有状态会话 Bean, 它的 createStyleXml() 方法用于从数据库中读出所需要的报表样式描述文件, 产生符合 XML 规范的报表样式描述文件; 该 EJB 中的另外一个方法 createDataXml() 用于从数据库中读取报表中需要动态生成的数据, 并将这些数据按预先定义的 XML 标记组织成符合 XML 规范的报表数据描述文件^[4]; 服务器中的 Servlet 则负责解析 EJB 读取数据库后产生的 XML 格式的文件, 将报表样式描述文件和报表数据描述文件解析为可以显示报表内容的 HTML 标签并传送给客户端浏览器^[5]。

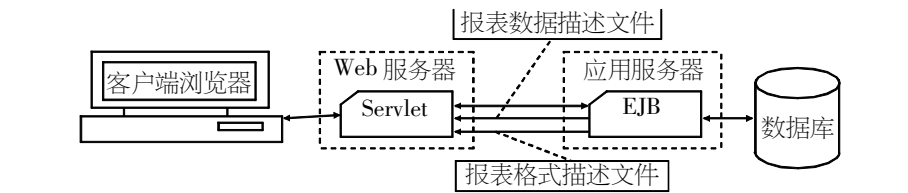


图 2 报表模型结构示意图

该模型的工作流程如下:

- (1) 将应用程序中需要实现的报表样式都按照预先定义的 XML 标记描述出来, 并给以一个唯一的标志存到数据库中, 以完成对报表样式描述文件的存储; 对应于每张应用程序中需要显示的报表, 都需要一张报表数据描述文件来实现其动态表格域的显示, 因此, 在描述完报表样式后, 需要在会话 Bean 中编码实现一个方法来完成该报表的数据描述文件的生成。
- (2) 当用户点击显示报表的按钮时, 客户端浏览器将所需要显示的报表 ID 号传给 Servlet。
- (3) Servlet 的 doPost() 方法接收到报表生成请求后, 调用会话 Bean 的 createStyleXml() 方法从数据库中获取相应的报表样式描述文件, 然后根据参数 ID 从会话 Bean 中找到对应该报表的 creatDataXml() 方法, 以便从数据库中读取数据后创建符合 XML 格式的报表数据描述文件^[6]。
- (4) Servlet 获得两种描述文件后, 通过 ParseXml() 方法解析这两个描述文件: 首先解析数据描述文件, 将会话 Bean 读取数据库后获得的数据描述文件按行存到相应的表标题、表头、表身、表尾数组中, 然后按行解析报表样式文件, 如果遇到静态文本域, 就产生 <td> </td> 标签并将文本嵌入其中, <td> 中文本的样式属性与样式描述文件中所描述的静态文本属性一致; 如果遇到的是动态文本域, 将相应数组中的数据嵌入 <td> </td> 中, 并将数组的指针后移; 如果遇到的是斜线域, 则调用 createSlaline() 方法产生一个与样式文件中描述的

斜线域大小、样式相同的图片嵌入到 <td> </td> 中。

(5) 将解析后得到的 HTML 文件传送到客户端浏览器即得到需要显示的报表。

1.4 实例应用

该报表模型的应用背景为煤矿安全信息管理系统。在实际项目中,该项目的用户——煤矿安全监察局需要经常对煤矿的“三违”、“隐患”、“工伤”、“安全质量标准化”、“职工安全培训”等信息进行统计分析,因此,涉及到大量复杂的报表,而且有对报表样式进行更改的需求。该模型的实现有效地解决了报表的复杂性以及报表样式变更给软件开发、维护所带来的困难,在很大程度上提高了软件的可维护性。表 1 给出了使用一张某矿人身事故统计表的显示示例。该报表是不定长度的报表,而且报表中有斜线域,也有嵌套表,是典型的复杂报表。该表用上述的报表生成模型实现后在客户端的显示结果见表 1。

表 1 某矿人身事故统计表							
2004 年 8 月份							
内容 序号	工伤日期		事故地点 及 事故原因	工伤情况			申报 工伤 日期
	日期	班次		伤亡 人数	伤亡人员 姓名	受伤部位 及程度	
1	2004-8-3	中班	在 568 石门两道风门间推车时 不慎被挤伤	1	王刚	肋部 轻伤	2004-8-9
2	2004-8-4	早班	在 528 工作面被顶板掉落的矸石砸伤	1	李明	手掌 轻伤	2004-8-11
3	2004-8-5	晚班	在 329 工作面联跑时被片帮煤砸伤	1	张华	右膝 轻伤	2004-8-13
4	2004-8-6	中班	在 346 备用面装面时被弓形梁挤伤	1	朱广其	左脚 轻伤	2004-8-16
5	2004-8-7	中班	在 768 石门两道风门间推车时 不填被挤伤	1	赵大宝	右臂 轻伤	2004-8-17
6	2004-8-8	晚班	在 143 工作面被顶板掉落的矸石砸伤	1	李少随	头额 轻伤	2004-8-19
7	2004-8-9	早班	在 722 备用面装面时被弓形梁挤伤	1	王新华	肋部 轻伤	2004-8-22
制表人: 张楠				制表日期:2004-9-1			
前一页 后一页				当前是第 1 页 总共 4 页			

2 关键技术解决方案

在本模型中有几个技术问题需要说明,例如如何画报表中的斜线问题,如何实现分页、打印问题。

2.1 斜线问题

报表中经常会出现带有斜线的表格域,但是 HTML 的标记并不提供在页面上画斜线的功能,因此,在 Servlet 解析描述文件的过程中,当解析到带有斜线域的表格时,Servlet 会产生一个与该表格域大小、所含斜线类型、斜线两侧文字相同的图片,并将该图片嵌入到相应的表格中,从而解决了报表中的斜线问题。

2.2 分页问题

在显示不定长度报表时,由于显示的数据可能比较多,因此需要将这些数据分页显示。在 Servlet 中实现了这一功能,用户将需要查看的页数参数 n 发送给 Servlet,Servlet 根据每页显示的行数 m 来计算显示第 n 页需要读取的数据,然后在客户端将这些数据显示出来。

2.3 打印问题

Web 打印是将数据显示到当前浏览器后通过 IE 的打印功能实现的,因此,浏览器会把报表内容之外的浏览器工具栏等内容一起打印。所以具体实现时,要让打印按钮的 onClick() 事件产生一个新的只包含需打印内容的简单窗口,然后对

该窗口进行打印,这样,就会只打印当前页的报表内容而不会打印出其他的内容。另外,对于分页的报表,通过循环调用打印功能即可实现连续打印。

3 基于 XML 的报表生成模型优点分析

该模型的提出为 B/S 结构下应用程序开发中所遇到的报表生成、打印问题提出了一个通用的解决方案,该方案的优点主要为:

(1) 该方案需要将显示的报表样式用自定义的 XML 标记描述出来并存储到数据表中,因此,在用户需要更改报表样式时,只需要更改数据表中的报表样式文件即可,相比那些在应用程序中采用固定的报表样式,并用 JavaBean 从数据库读取数据的报表显示方案,该方案具有很高的灵活性。

(2) 该方案提出的报表描述规则只是按行依次对每行的表格进行描述,而不是采用先按线条来描述报表样式然后在 Java 画布上进行报表生成的方案^[7]。与后者相比,该方案简便实用,更适合 B/S 结构的应用程序使用。

(3) 该方案中负责解析报表格式描述文件和报表数据描述文件的部分可以在客户端用 JavaScript 等脚本语言实现,在这种结构下,只需将服务器端获得的两个描述文件传到客户端即可实现报表的描述,可以适当减轻服务器端的工作量。

(4) 该方案在生成报表的过程中,实现了报表样式和报表数据的分离,这样不仅增强了数据的共享程度和独立性,而且在生成过程中减少了与数据库服务器的交互次数,在很大程度上减轻了数据库服务器的负担。

4 下一步工作

该报表生成方案解决了一般的 B/S 结构下应用程序的报表生成、打印问题,但该模型还存在一些缺点,例如没有实现报表生成的构件化^[8],下一步工作重点是结合实际的项目,将该模型做成由一些可以重用的构件组成的报表生成系统。这样一来,用户可以在给定的界面中通过拖拉所给的控件来组成需要的报表,并在控件的属性里输入数据来源信息来实现报表中动态数据的填充,使该方案做到真正意义上的构件化、自动化。

参考文献:

[1] 沈兆阳. Java 与 XML 数据库整合应用[M]. 北京: 清华大学出版社, 2002. 10-12.

[2] 李涛涛, 刘连忠, 陈梦东. 基于 XML 技术实现表格的灵活构建[J]. 计算机应用研究, 2004, 21(1): 54-56, 60.

[3] Elliotte Rusty Harold. XML 实用大全[M]. 北京: 中国水利水电出版社, 2000. 215-270.

[4] 刘晓华. J2EE 企业级应用开发[M]. 北京: 电子工业出版社, 2003. 26-29.

[5] 林绍方. 深入掌握 J2EE 编程技术[M]. 北京: 科学出版社, 2002. 93-111.

[6] Ed Roman. 精通 EJB[M]. 北京: 电子工业出版社, 2002. 67-72.

[7] 刘磊, 刘弘, 等. 基于软件重用的报表构件生成系统的研究与实现[J]. 小型微型计算机系统, 2003, 24(4): 739-743.

[8] 刘立志, 孙莹, 陈松乔. 基于构件技术的报表系统的设计与实现[J]. 计算机应用研究, 2003, 20(9): 88-90.

作者简介:

吴雷(1981-), 男, 安徽淮北人, 硕士研究生, 研究方向为软件工程; 袁兆山(1945-), 男, 山东苍山人, 教授, 研究方向为软件工程、计算机网络; 李超(1979-), 男, 安徽合肥人, 硕士研究生, 研究方向为数据库技术。