



(12) 发明专利

(10) 授权公告号 CN 102436372 B

(45) 授权公告日 2014. 07. 02

(21) 申请号 201110254476. 4

与实现. 《内蒙古工业大学学报》. 2007, 第 26 卷
(第 4 期), 第 291-296 页.

(22) 申请日 2011. 08. 31

审查员 孙莉莉

(73) 专利权人 中国运载火箭技术研究院

地址 100076 北京市 9200 信箱 38 分箱

(72) 发明人 廖馨 张灏龙 冷传航 施国强
李烁 李贵成

(74) 专利代理机构 中国航天科技专利中心
11009

代理人 庞静

(51) Int. Cl.

G06F 9/44 (2006. 01)

G06F 17/30 (2006. 01)

(56) 对比文件

CN 101216817 A, 2008. 07. 09, 全文.

赵中华等. 基于 XML 的 B/S 架构下动态报表的实现方式. 《山东轻工业学院学报》. 2008, 第 22 卷 (第 1 期), 第 29-31 页.

李艳娥等. 基于 AJAX 技术的 Web 报表的研究

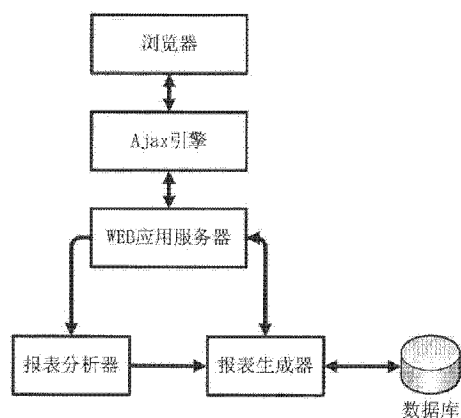
权利要求书1页 说明书8页 附图2页

(54) 发明名称

一种监控表格生成系统

(57) 摘要

一种监控表格生成系统, 包括表格配置文件、表格分析器、表格生成、Ajax 引擎、WEB 应用服务器和浏览器; 该系统具有对表格进行格式设计、取数设计、显示设计以及自动生成等功能, 表格的这些定义过程的实现完全由用户自定义, 不需要编程实现。为了更好地满足动态监控需求, 另外提供了表格的动态刷新频率定义, 或者表格不同字段的刷新频率定义等功能。



1. 一种监控表格生成系统,其特征在于:包括表格配置文件、表格分析器、表格生成器、Ajax 引擎、WEB 应用服务器和浏览器;

表格配置文件,采用 xml 形式定义表格静态信息、取数规则、填充规则和刷新规则,表格配置文件存储在 WEB 应用服务器中;

表格分析器,根据接收到的表格生成请求,从 WEB 应用服务器中获取表格配置文件,对 xml 配置文件进行解析,生成表格静态信息、取数规则、填充规则和刷新规则,将取数规则、填充规则和刷新规则合并为规则集,并将表格静态信息和规则集发送给表格生成器;

表格生成器用来负责处理表格生成工作,接受来自表格分析器的表格静态信息和规则集,根据规则集中的取数规则对数据库进行查询,得到结果数据,根据规则集中的填充规则,将结果数据和表格静态信息组装并生成表格,并将该表格和刷新规则发送至 WEB 应用服务器;

WEB 应用服务器,接收 Ajax 引擎的请求,在接收到表格生成请求时,将其转发给表格分析器;接收表格生成器生成的表格及刷新规则,并将其传递给 Ajax 引擎;在从 Ajax 引擎获取周期查询指令时,向数据库发出数据查询请求,接收数据查询结果,并将其转发给 Ajax 引擎;

Ajax 引擎从 WEB 应用服务器接收表格数据和刷新规则,将表格数据直接传递给浏览器;根据表格的刷新规则,针对需要更新的数据,对 WEB 应用服务器发送周期查询指令,并接收需要更新的数据的查询结果,并将其发送给浏览器,对 html 页面进行动态刷新;

浏览器,浏览器显示表格界面,接受用户对表格生成的请求,向 Ajax 引擎发出相应指令,并获取 Ajax 引擎返回的处理结果,将其展示给用户;

上述 Ajax 引擎、WEB 应用服务器和浏览器之间的数据传递步骤如下:

- (1)WEB 应用服务器将报表数据和刷新规则传递给 Ajax 引擎;
- (2)Ajax 引擎根据刷新规则,生成周期查询命令并发送给 WEB 应用服务器;
- (3)Ajax 引擎将报表数据转化为 HTML 页面数据,并根据刷新规则定义表格中的数据颜色,将页面数据传递给浏览器,显示为页面;
- (4)WEB 应用服务器收到周期查询命令,向数据库发出数据查询请求;
- (5)WEB 应用服务器收到数据查询结果,将其传递给 Ajax 引擎;
- (6)Ajax 引擎收到数据查询结果,将其转化为需要更新的数据发送给浏览器;
- (7)浏览器收到需要更新的数据,更新页面。

2. 根据权利要求 1 所述的一种监控表格生成系统,其特征在于:所述的表格生成器实现步骤如下:

第一步,从表格分析器中获取表格静态信息和规则集;

第二步,对规则集进行规则识别,根据取数规则向数据库发送数据查询请求;

第三步,从数据库获取数据查询结果;

第四步,根据数据查询结果、表格静态信息和填充规则,将三者组装为表格数据;

第五步,将第四步中组装成的表格数据和刷新规则发送至 WEB 应用服务器。

一种监控表格生成系统

技术领域

[0001] 本发明涉及一种监控表格生成的方法,属于计算机技术领域。

背景技术

[0002] 互联网已经深入到人们生活的各个领域,越来越多的企业、组织和机构已经组建或正在组建自己的内部网,许多管理程序的开发也已经从传统的 C/S 架构向基于 Internet/Intranet 的 B/S 架构转变。虽然不同的程序架构带来了开发过程以及开发中所用的各项技术和各类开发工具的转变,但对于大多数的管理程序,监控表格的生成是开发过程中不可避免的一项任务。对于网络计算系统来说,表格系统是网络系统中的数据统计、分析和展示工具,实现历史数据的可视化,让用户对计算资源的使用情况了然于胸,为未来的软硬件规划提供决策依据,帮助预测未来计算设施使用趋势。

[0003] 在传统的 C/S 架构的数据库应用程序的开发过程中,各厂商的应用程序开发工具都对表格的开发提供了强有力的支持,可以方便地完成 C/S 架构下的数据库应用程序开发过程中的表格生成任务。而在 B/S 架构的数据库应用程序开发中,由于其程序运行环境的特殊性,目前还没有简单易用的表格开发工具。此外客户端应用程序运行环境是通用浏览器,只能显示静态的 HTML 文本,这也为动态表格的开发带来了一定的难度。

[0004] 微计算机信息,2005 年 26 期,《基于 XML 的动态数据报表生成技术》只是采用 XML 定义了一种企业数据的抽取集合,不能进行动态展示参数配置;《基于 XML 的层次化项目信息管理系统的设计与实现》采用 XML 定义系统的用户界面;《基于 WEB 的动态报表系统之展示子系统研究》采用 SVG 文件对报表进行展示,与本发明的实现机制不同。

[0005] 《基于 XML 的 B/S 架构下动态报表的实现方式》、《基于 AJAX 技术的 Web 报表的研究与实现》、《基于 Excel 的企业通用动态报表生成工具的设计》和《基于 ActiveX 的 EIP 动态报表系统》论及的动态报表均为静态显示,不能实现动态显示也不能实现定时刷新功能,系统的可继承性不强。《实现动态报表的方法及其数据的组织》没有采用 XML 描述语言。《一种动态报表生成方法及平台》和《ABSTRACTED DYNAMIC REPORT DEFINITION GENERATION FOR USE WITHIN INFORMATION TECHNOLOGY INFRASTRUCTURE》是两篇专利文献,文中论及的动态报表装置采取静态显示,不能实现动态显示也不能实现定时刷新功能,系统的可继承性不强。

发明内容

[0006] 本发明的技术解决问题是:克服现有技术的不足,提供一种基于 XML 语言的监控表格生成系统,该方法能够对 WEB 页面形式的监控表格进行动态的界面配置和生成,并提供数据的自动刷新功能。

[0007] 本发明的技术方案是:一种监控表格生成系统,包括表格配置文件、表格分析器、表格生成、Ajax 引擎、WEB 应用服务器和浏览器;

[0008] 表格配置文件,采用 xml 形式定义表格静态信息、取数规则、填充规则和刷新规

则,表格配置文件存储在 WEB 应用服务器中;

[0009] 表格分析器,根据接收到的表格生成请求,从 WEB 应用服务器中获取表格配置文件,对 xml 配置文件进行解析,生成表格静态信息、取数规则、填充规则和刷新规则,将取数规则、填充规则和刷新规则合并为规则集,并将表格静态信息和规则集发送给表格生成器;

[0010] 表格生成器用来负责处理表格生成工作,接受来自表格分析器的表格静态信息和规则集,根据规则集中的取数规则对数据库进行查询,得到结果数据,根据规则集中的填充规则,将结果数据和表格静态信息组装并生成表格,并将该表格和刷新规则发送至 WEB 应用服务器;

[0011] WEB 应用服务器,接收 Ajax 引擎的请求,在接收到表格生成请求时,将其转发给表格分析器;接收表格生成器生成的表格及刷新规则,并将其传递给 Ajax 引擎;在从 Ajax 引擎获取周期查询指令时,向数据库发出数据查询请求,接收数据查询结果,并将其转发给 Ajax 引擎;

[0012] Ajax 引擎从 WEB 应用服务器接收表格数据和刷新规则,将表格数据直接传递给浏览器;根据表格的刷新规则,针对需要更新的数据,对 WEB 应用服务器发送周期查询指令,并接收需要更新的数据的查询结果,并将其发送给浏览器,对 html 页面进行动态刷新;

[0013] 浏览器,浏览器显示表格界面,接受用户对表格生成的请求,向 Ajax 引擎发出相应指令,并获取 Ajax 引擎返回的处理结果,将其展示给用户。

[0014] 所述的表格生成器实现步骤如下:

[0015] 第一步,从表格分析器中获取表格静态信息和规则集;

[0016] 第二步,对规则集进行规则识别,根据取数规则向数据库发送数据查询请求;

[0017] 第三步,从数据库获取数据查询结果;

[0018] 第四步,根据数据查询结果、表格静态信息和填充规则,将三者组装为表格数据;

[0019] 第五步,将第四步中组装成的表格数据和刷新规则发送至应用服务器。

[0020] 本发明与现有技术相比有益效果为:本发明提出了一种基于 XML 的监控表格生成系统。该系统具有对表格进行格式设计、取数设计、显示设计以及自动生成等功能,表格的这些定义过程的实现完全由用户自定义,不需要编程实现。为了更好地满足动态监控需求,另外提供了表格的动态刷新频率定义,或者表格不同字段的刷新频率定义等功能。具体如下:

[0021] (1) 具备与多个系统集成的能力

[0022] 动态表格生成系统与数据库直接关联,并支持 WEB 浏览器,因此在与 WEB 应用系统集成时,只需要通过页面集成和简单的数据库配置,就可以在 B/S 架构下的各类应用中广泛集成和使用。

[0023] (2) 数据动态刷新

[0024] 本发明提供了表格数据的动态刷新功能,并可以配置数据的刷新参数,适合应用在数据实时更新的系统监控中。

[0025] (3) 报告格式的可继承性

[0026] 本发明将表格的静态信息、取数规则、填充规则和动态刷新规则组合定义为配置文件,将表格的表现形式和内容分开考虑,可以使得表格定制人员继承以前的表格格式,对

数据内容稍作修改就可以新建一份全新的报告,减少了表格定制人员的工作量。

[0027] (4) 简单易用

[0028] 采用基于 WEB 的访问方式,用户在使用时,不需要下载客户端程序,也不需要安装任何其他系统补丁。

附图说明

[0029] 图 1 为本发明系统结构示意图;

[0030] 图 2 为本发明表格分析器示意图;

[0031] 图 3 为本发明表格生成器示意图;

[0032] 图 4 为本发明 Ajax 引擎、WEB 应用服务器和浏览器之间数据传递示意图。

具体实施方式

[0033] 监控表格生成系统以 WEB 形式表现,为用户提供表格生成和显示的功能。如图 1 所示,本发明系统分为表格配置文件、表格分析器、表格生成、Ajax 引擎、WEB 应用服务器和浏览器六部分。在本发明中 Ajax 引擎选择 Dojo、WEB 服务器选择 Tomcat,浏览器选择 IE6.0,均为成熟产品,不详细介绍其实现过程,本发明中。下面详细介绍表格配置文件、表格分析器、表格生成器的实现过程。

[0034] 一、表格配置文件

[0035] 表格配置文件由四个部分构成:表格静态信息、取数规则和填充规则以及动态刷新规则。

[0036] (1) 静态信息定义

[0037] 表格静态信息是对整个表格静态内容的描述。下表是在 html 中表格的描述方式。

[0038] 作业表片段:

[0039]

作业 ID	作业名称	作业节点	作业状态
1	B01X00	1	2
2	C01X00	3	3
3	B01X00	5	10

[0040] 作业表片段的 HTML 表示如下:

[0041] <table>

[0042] <tr>

[0043] <td> 作业 ID</td>

[0044] <td> 作业名称 </td>

[0045] <td> 作业节点 </td>

[0046] <td> 作业状态 </td>

```
[0047]    </tr>
[0048]    <tr>
[0049]    <td>1</td>
[0050]    <td>B01X00</td>
[0051]    <td>1</td>
[0052]    <td>2</td>
[0053]    </tr>
[0054]    <tr>
[0055]    <td>2</td>
[0056]    <td>C01X00</td>
[0057]    <td>3</td>
[0058]    <td>3</td>
[0059]    </tr>
[0060]    <tr>
[0061]    <td>3</td>
[0062]    <td>B01X00</td>
[0063]    <td>5</td>
[0064]    <td>10</td>
[0065]    </tr>
[0066]    </table>
```

[0067] 上面的 html 描述完整反映了作业表格的全貌,并且 table 标签中的内容完全符合 XML 片断的要求,是一个格式规范的 XML 片断,我们在进行表格静态内容的描述时借用了 html 中表格的定义方式。

[0068] 在动态表格系统中不定义单元格描述细节和字体,只定义单元格内容。在作业表中从第二行开始的内容都是从数据库中动态生成的,属于静态的内容只有第一行。因此,只需要用 html 描述表格的第一行信息。

[0069] 表格 xml 配置文档中的静态信息描述片段如下:

```
[0070]    <table>
[0071]    <tr row='1'>
[0072]    <td col='1'>作业 ID</td>
[0073]    <td col='2'>作业名称</td>
[0074]    <td col='3'>作业节点</td>
[0075]    <td col='4'>作业状态</td>
[0076]    </tr>
[0077]    </table>
```

[0078] (2) 取数规则定义

[0079] 取数规则与填充区域的定义是对表格取数动态内容的描述,描述了如何从其他数据库中取数据来填充表格上的相关单元格。一张表格可以包含若干个填充区域,每个填充区域对应一条取数规则,这样可以组成各种复杂表格。在极端情况下每一个单元格对应一

条取数规则。

[0080] 取数规则负责解释如何从数据库中抽取需要的数据。取数规则的核心是一个查询 SQL 语句,定义如何对数据库查询需要的数据。在系统中采用标准 sql 语句,如:

[0081] SELECT id,name,status FROM boinc_job WHERE node=3;

[0082] (3) 填充规则定义

[0083] 填充区域要定义取数规则中的查询返回值与表格每个字段之间的填充映射关系。填充规则可以指定查询返回的首条记录填充到哪些单元格。填充规则由 xml 文档中的 <fields> 元素定义,<fields> 元素的子节点 <field> 描述了数据的填充区域。

[0084] 表格 xml 配置文档中的取数规则和填充方式片段如下:

[0085]

<fillin>

<area>

<connstr>仿真作业<connstr>

<sql> SELECT id,name,status FROM boinc_job WHERE node=3;</sql>

<fields>

<field col='1'/>

<field col='2'/>

<field col='3'/>

<field col='4'/>

</fields>

</area>

</fillin>

[0086] (4) 刷新规则

[0087] 通过刷新定制区域的规则定制,可以选择对不同关键数据具有不同的监控刷新频率,以达到更好的动态监控效果。且可以为监控的数据设定颜色显示区间。例如,当该数据大于某个阈值时,背景颜色设置为绿色,小于某个数时,背景颜色设置为红色。这样能让操作者对数据的当前状态一目了然。

[0088] 在 xml 文档里添加一个元素 <refresh>,<r6fresh> 元素的子节点描述了数据的刷新规则。

[0089] 表格 xml 配置文档中的刷新定制片段如下:

[0090]

```

<refresh>
  <field col='4' frequency='10'>
    <sql SELECT status FROM boinc_job WHERE id in #;/>
    <format comparator='equal'threshold='3'color='green'/>
  <format comparator='lt' threshold='3' color='red'/>
  </field>
  ...
</refresh >

```

[0091] 上述刷新规则中, <sql SELECT status FROM boinc_job WHERE id in#;/> 语句会传给 Ajax 引擎。Ajax 引擎在收到的表格数据中选择所有的 id, 填充该 sql 语句, 如 :sql SELECT status FROM boinc_job WHERE id in(1,2,3), 再将该 sql 语句异步传输给 WEB 应用服务器, 进行查询, 然后根据返回数据, 对表中列号为 4 的单元格进行 10 秒一次的动态刷新。当状态 =3 时, 背景色为绿色, 当状态 >3 时, 背景色为红色。

[0092] 下面给出一个表格定义 XML 文档的完整示例 :

[0093]

```

<report>
  <table>
    <tr row='1'>
      <td col='1'>作业 ID</td>
      <td col='2'>作业名称</td>
      <td col='3'>作业节点</td>
      <td col='4'>作业状态</td>
    </tr>
  </table>
  <fillin>
    <area>
      <connstr>仿真作业</connstr>
    </area>
  </fillin>
</report>

```

[0094]

```

    <sql>    SELECT    id,name,status    FROM    boinc_job    WHERE
node=3;</sql>

    <fields>
        <field col='1'/>
        <field col='2'/>

    <field col='3'/>
    <field col='4'/>
    </fields>

    <refresh>

        <field col='4' frequency='10'>
            <sql SELECT status FROM boinc_job WHERE id in #;/>
        <format comparator='equal'threshold='3' color='green'/>
        <format comparator='lt' threshold='3' color='red'/>
    </field>

        ...

    </refresh >

</area>

</fillin>

</report>

```

[0095] 二、表格分析器

[0096] 表格分析器负责解析 xml 配置文件,根据配置文件发送规则到表格生成器。

[0097] 表格分析器的实现示意图如图 2 所示,实现步骤如下:

[0098] 第一步,从 Web 应用服务器中获取表格配置文件。

[0099] 第二步,使用 DOM 解析器解析表格配置文件,解析 table 标签、fillin 标签、sql 标签、fields 标签和 refresh 标签,生成表格静态信息、取数规则、填充规则和动态刷新规则,并将取数规则、填充规则和动态刷新规则合并为规则集;

[0100] 第三步,将第二步解析出的表格静态信息和规则集发送至表格生成器。

[0101] 三、表格生成器

[0102] 表格生成器用来负责处理表格生成工作,接受来自表格分析器的规则,对数据库进行查询,得到数据,根据规则组装定制数据,生成表格。

[0103] 表格生成器的实现示意图如图 3 所示,实现步骤如下:

[0104] 第一步,从表格分析器中获取表格静态信息和规则集;

[0105] 第二步,对规则集进行规则识别,根据取数规则向数据库发送数据查询请求;

[0106] 第三步,从数据库获取数据查询结果;

[0107] 第四步,根据数据查询结果、表格静态信息和填充规则,将三者组装为表格数据;

[0108] 第五步,将第四步中组装成的表格数据和刷新规则发送至应用服务器。

[0109] 四、Ajax 引擎、WEB 应用服务器和浏览器

[0110] Ajax 引擎是浏览器和应用服务器之间的中间层,使用户操作与服务器响应异步化。页面上的数据验证和数据处理可以由 Ajax 引擎进行,如果需要从服务器读取新的表格数据时再由 Ajax 引擎代为向服务器提交周期查询请求。

[0111] WEB 应用服务器选择 Tomcat。Tomcat 是一个开放源代码、运行 servlet 和 JSP Web 应用程序的基于 Java 的 Web 应用软件容器,实现解析 HTTP 请求并生成响应传回浏览器,并为应用程序提供业务逻辑。

[0112] 浏览器是可以显示 HTML 文件内容,并让用户与这些文件交互的软件。本专利中选择 IE6.0 作为浏览器,接受 Ajax 引擎转交的 HTML 文件,并显示给用户。

[0113] Ajax 引擎、WEB 应用服务器和浏览器之间的数据传递示意图如图 4 所示,实现步骤如下:

[0114] 第一步,应用服务器将报表数据和刷新规则传递给 Ajax 引擎;

[0115] 第二步, Ajax 引擎根据刷新规则,生成周期查询命令并发送给应用服务器;

[0116] 第三步, Ajax 引擎将报表数据转化为 HTML 页面数据,并根据刷新规则定义表格中的数据颜色,将页面数据传递给浏览器,显示为页面;

[0117] 第四步,应用服务器收到周期查询命令,向数据库发出数据查询请求;

[0118] 第五步,应用服务器收到数据查询结果,将其传递给 Ajax 引擎;

[0119] 第六步, Ajax 引擎收到数据查询结果,将其转化为需要更新的数据发送给浏览器;

[0120] 第七步,浏览器收到需要更新的数据,更新页面。

[0121] 本发明未详细说明部分属于本领域技术人员公知常识。

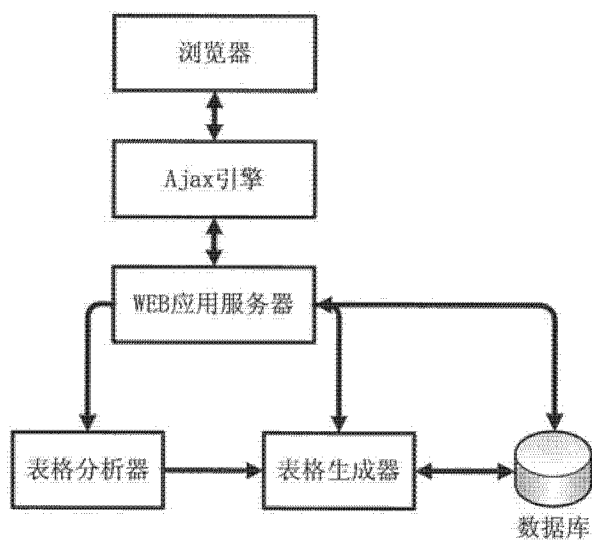


图 1

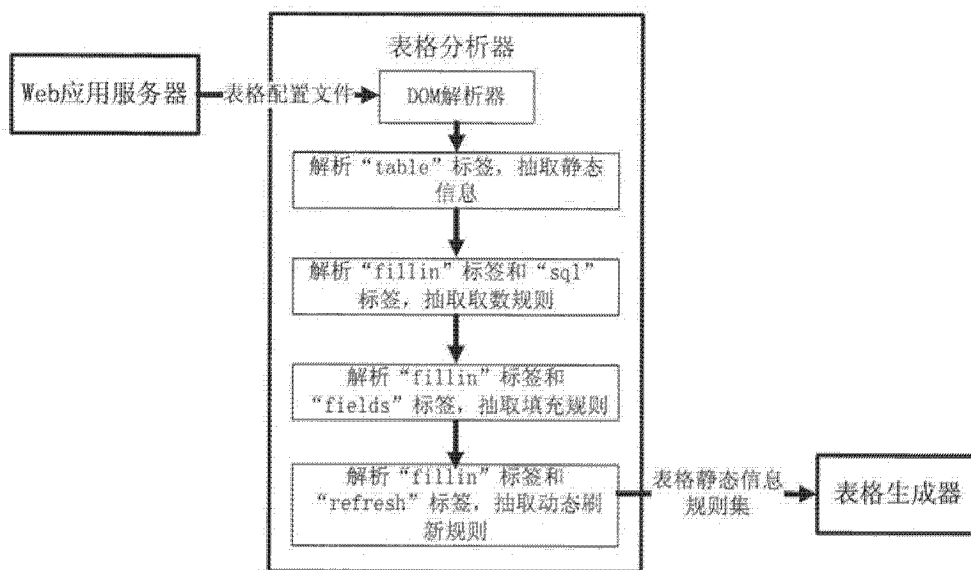


图 2

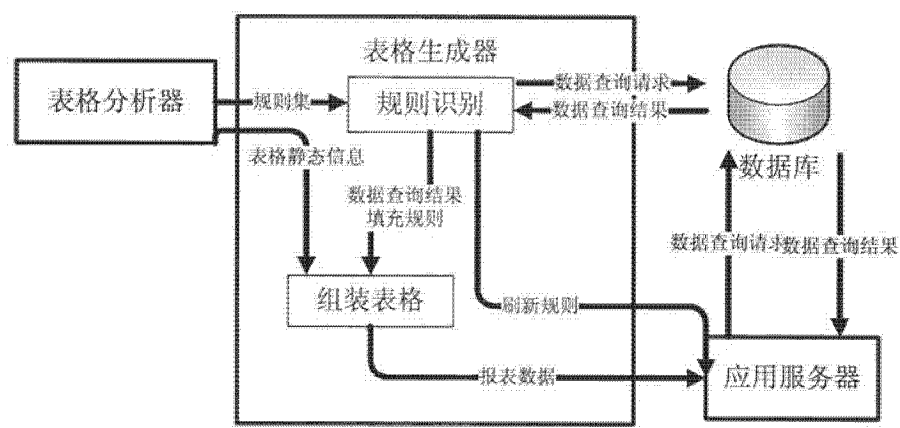


图 3

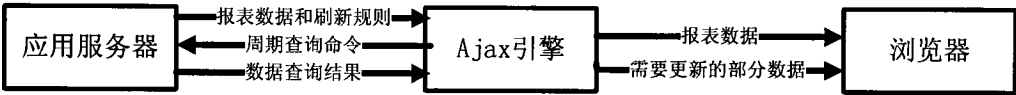


图 4