

文章编号: 1005-8451 (2012) 02-0058-04

# 使用 JSP+VML 动态绘制列车运行图

赵旭辉, 冯俊杰

(辽宁铁道职业技术学院 运输系, 锦州 121000)

**摘要:** 列车运行图是重要的行车文件, 在铁路运营管理中发挥着重要作用。使用 VML+JSP 技术, 通过读取后台的列车时刻表数据, 动态生成运行图, 可以及时地反应出列车数据的动态变化, 与图片格式的运行图相比, 数据传输量小, 能够充分利用客户端浏览器的计算能力, 减轻服务器的负担。

**关键字:** 列车运行图; VML; JSP

**中图分类号:** U292.42 : TP39      **文献标识码:** A

## Method of drawing dynamic train diagram by JSP+VML

ZHAO Xu-hui, FENG Jun-jie

(Department of Transportation, Liaoning Railway Vocational and Technical College, Jinzhou 121000, China)

**Abstract:** Train diagram was an important document in the management of railway operations and played an important role. Using VML + JSP technology, by reading the train schedule data of the back to generate a dynamically diagram, it could reflect the dynamic changes of data in time. Compared with the diagram of image format, the amount of data transfer was smaller, it could take full advantage of computing power for the client browser, reduce the burden on the server.

**Key words:** train diagram; VML; JSP

铁路列车运行图是列车运行的图解, 它既是行车组织工作的基础, 又是联系各部门工作的纽带。为配合教学, 我们使用 VML+JSP 技术, 开发了“铁路现在车管理系统教育模拟软件”, 较好地实现了列车运行图的动态绘制。本文主要介绍通过读取后台的列车时刻表数据, 动态绘制列车运行图的方法。

### 1 铁路列车运行图概述

铁路列车运行图是运用坐标原理表示列车运行时间、空间关系的一种图解形式。是列车在铁路线路运行及在车站到发或通过时的技术文件。列车运行图以垂直线等分横轴表示时间, 按每一等分的时间不同, 运行图分为二分格运行图, 十分格运行图和小时格运行图, 将横轴按一定比例用横线加以划分, 每一横线表示一个车站的中心线; 列车运行线, 由于列车速度的不断变化本来是一条不规则的曲线, 为简化起见而将其画为斜直线。

收稿日期: 2011-11-30

作者简介: 赵旭辉, 讲师; 冯俊杰, 副教授。

### 2 使用 VML 绘图

VML (Vector Markup Language) 是一种基于 XML 的矢量标记语言, 它的核心也是基于 HTML。IE 5.0 以上版本都对 VML 提供支持。VML 相当于 IE 中的画笔, 可以使用代码标签在网页内绘制各种矢量图形。VML 标签可以由 IE 浏览器直接解析、输出。使用它可以方便地在 WEB 上建立高效、灵活、简洁的各种矢量图形。

在 Web 页面中引用 VML, 首先需要在 <HTML> 标签中加上关键语句, 指明该网页创建一个名为“V”的空间。

```
<html xmlns:v="urn:schemas-microsoft-com:
vml">
```

然后, 在 CSS 样式表的定义语句添加关键语句, 指明 xml 名域“v”引用的数据是 VML 标记语言。

```
<STYLE>
```

```
v:* { BEHAVIOR: url(#default#VML) }
```

```
</STYLE>
```

通过以上设置, 就可以在 HTML 语句中嵌套

VML语句,与正常的HTML一样使用。

在VML中单个的元素被定义为形状,大多数形状是由矢量路径描述的,它提供了一些预定义的形状如直线、曲线等。VML还提供了shapeType模版功能,利用它可以事先将复杂的图形绘制为模板,使用时直接声明type属性即可。这样即减少了代码的书写量,又使可读性大大提高。

在绘制运行图过程中,我们设计了一个信封作为列车的确报标志,通过右键点击会弹出收发确报的功能菜单。在这个标志的绘制上预先绘制了模板。代码如下:

确报标志绘制:

```
<v:shapetype id="quebao" strokeweight=0.5pt strokecolor=red coordsize="600 600">
  <v:path v="m 0,0 c 200,300 300,400 400,300 1600,0 0,0 0,500 600,500 600,0 e"/>
</v:shapetype>
```

对于列车的始发标志和终到标志也同样绘制了模板。

终到标志绘制:

```
<v:shapetype id="zhongdao" strokeweight=1pt strokecolor=green coordsize="600 600">
  <v:path v="m 200,0 l 200,200 300,200 200,500 100,200 200,200 e"/>
</v:shapetype>
```

列车始发标志:

```
<v:shapetype id="shifa" strokeweight=1pt strokecolor=green coordsize="600 600">
  <v:path v="m 0,0 l 200,0 100,0 100,300 e"/>
</v:shapetype>
```

列车运行线的绘制使用polyLine进行,通过获取列车在2个相邻站始发和本站到达时刻的坐标进行绘制。

### 3 数据的动态获取与绘制

JSP (Java Server Page) 是比较常用的 Web 开发技术,通过使用JSTL标签和EL表达式可以极其方便地完成与后台服务器的信息交互。列车的到发时刻可以从列车时刻表获取,但是列车时刻表只提供时间数据,需要进行相应的转换,变成到发时刻的坐标数据。VML中shape的CoordSize

属性就是用来定义坐标的。

将列车的到发时刻转换为坐标的方法如下:

```
public void caculatedot(Shikebao skb)
{
    double begindot;
    double enddot;
    double devdot;
    int beginhour=Integer.valueOf(skb.getDaodate().substring(0,2));
    int endmin=Integer.valueOf(skb.getDaodate().substring(3));
    if(skb.getLeixing()==1)
    {
        enddot=100+(((beginhour<18?24+beginhour:beginhour)-18)*60+endmin)*8.5;
        begindot=(enddot-15*8.5>100?enddot-15*8.5:100);
        devdot=begindot+(enddot-begindot)/2;
    }
    else
    {
        begindot=100+(((beginhour<18?24+beginhour:beginhour)-18)*60+endmin)*8.5;
        enddot=(begindot+15*8.5);
        devdot=begindot+(enddot-begindot)/2;
    }
    skb.setBegindot(String.valueOf(begindot));
    skb.setEnddot(String.valueOf(enddot));
    skb.setDevlope(String.valueOf(devdot));
    skdao.saveskb(skb);
}
```

以上实现了由列车时刻到坐标的转换,再通过VML的polyLine画线就可以完成一条列车运行线的绘制。

在JSP页面中使用forEach标签、If标签可以将时刻表中的所有车次都绘制出来。并将已经接入的车次显示为绿色信封帮助用户清楚的辨别出列车是否已经接车。对于车次和始发、终到等标志可以与列车的运行线相连接,完成运行图的绘制。其主要代码如下:

```
<c:forEach items="${requestScope['com.zxh.xzc.shikebiao.list']}" var="skb" varStatus="s">
```

```

// 判断列车类型是终到
<c:if test="${skb.leixing==1}">
    // 绘制列车运行线
    <v:polyLine id="${skb.checi}"
        style="position:relative;z-index:9"
        filled=f strokecolor=green
strokeweight=0.5pt
        points="${skb.beginDot},500 $
{skb.endDot},1900"/>
    // 列车运行线的上方显示当前的
    车次
    <P style="LEFT:${skb.beginDot}px;
POSITION:absolute;TOP:55px;HEIGHT:5.6pt;
        font-size:10pt;TEXT-ALIGN:
center">${skb.checi}</P>
    // 绘制列车确报的标志, 此处进行判断如
    已接车, 则显示为绿色信封, 否则为红色信封, 并
    装配右键点击的弹出菜单参数
    <v:shape id="${skb.checi}dev"
type="#quebao"
        <c:if test="${skb.jieche==1}">
strokecolor=green </c:if>
        style="cursor:hand;position:
relative;z-index:10;
            left:${skb.devlope};top:1300;
width:40;height:250"
            title="${skb.leixing}${skb.checi}$
{skb.jieche}"
            onmousedown="showmenu($
{skb.leixing}${skb.checi}${skb.jieche})"
            onmouseout="hidemenu()"/>
    // 显示列车到发时刻的 10 min 以下的
    数字
    <P style="LEFT:${41/70*skb.
endDot}px;POSITION:absolute;TOP:120px;
        HEIGHT:5.6pt;font-size:10pt;
TEXT-ALIGN:center">
        ${fn:substring(skb.
daodateime,4,5)}</P>
    // 在列车运行线下方增加终到标志
    <v:shape id="${skb.checi}zd"
type="#zhongdao" style="position:relative;

```

```

z-index:11;left:${skb.enddot-
35};top:1900;width:100;height:150" />
    </c:if>
    // 以下为始发列车的画法, 与上同。
    <c:if test="${skb.leixing==2}">
    <v:polyLine id="${skb.checi}"
style="position:relative;z-index:9"
        filled=f strokecolor=green
strokeweight=1pt
        points="${skb.beginDot},
1900 ${skb.endDot},3000"/>
    <P style="LEFT:${41/70*skb.
beginDot}px;POSITION:absolute;TOP:145px;
        HEIGHT:5.6pt;font-size:10pt;
TEXT-ALIGN:center">${skb.checi}</P>
    <v:shape id="${skb.checi}dev"
type="#quebao"
    // 已发出的列车显示为绿色
    <c:if test="${skb.jieche==1}
"> strokecolor=green </c:if>
        style="cursor:hand;position:
relative;z-index:10;
            left:${skb.devlope};top:2450;
width:40;height:250"
            title="${skb.leixing}${skb.checi}
${skb.jieche}"
            onmousedown="showmenu($
{skb.checi}${skb.jieche})"
            onmouseout="hidemenu()"/>
    <p style="LEFT:${41/70*skb.
beginDot}px;POSITION:absolute;TOP:115px;
        HEIGHT:5.6pt;font-size:10pt;
TEXT-ALIGN:center">
        ${fn:substring(skb.daodateime,
4,5)}</P>
    <v:shape id="${skb.checi}zd"
type="#shifa" style="position:relative;
        z-index:11;left:${skb.beginDot-
17};top:1810;width:100;height:150" />
    </c:if>
</c:forEach>
系统动态绘制的运行图如图 1。

```

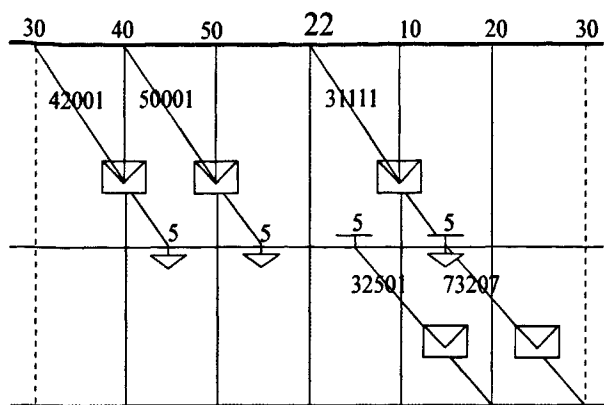


图 1 系统动态绘制的运行图

## 4 结束语

传统的使用图片方式绘图运行图，需要事先

绘制，不能随动态数据变化，图片需要从服务器端下载到客户端才能显示，有一定的传输时延，用户的体验性不好，服务器的负担也较大。而采用 VML 绘制运行图，充分利用了客户浏览器本身的计算功能，绘图工作在客户端完成，服务器压力较小，而且动态绘图能够适应信息的即时变化，方便快捷，实用性较高。

### 参考文献：

- [1] 张雅净, 王鹤鸣. 铁路行车调度[M]. 北京: 中国铁道出版社, 2010.
- [2] 欧国明, 黄云峰, 黄键, 等. 用 ASP+VML 实现气象图表的快速绘制[J]. 广东气象, 2010, 32 (5).

责任编辑 方 圆

(上接 P57)

LEFT JOIN machine m WITH (nolock) ON  
m.id=d.machine\_id

在上述视图建立好的基础上，然后在 Microsoft SQL Server Management Studio 下面用 SQL 语言编写测试代码<sup>[6]</sup>，具体如下：

```
DECLARE @num_before BIGINT,@num_old  
BIGINT
```

```
SELECT @num_before=count(p.id) FROM  
pathDS
```

```
SELECT @num_old=count(p.id) FROM  
pathDS p JOIN pathDSView pv ON
```

```
pv.id=p.id WHERE p.[path] = pv.[path]
```

```
IF @num_before==@num_old PRINT '数据  
转移成功！'
```

```
ELSE PRINT '数据转移失败！'
```

上述 SQL 代码执行后，如果输出‘数据转移成功！’，证明结果正确；如果输出‘数据转移失败！’，说明结果有误。本文所研究的海量数据转移在实际项目当中结果完全正确。

## 4 结束语

论文针对数据库优化过程中，数据表结构发生变化的时候，海量数据转移过程中遇到的数据

完整性和安全性问题，提出了一种基于 Python 语言，结合数据库事务特性和存储过程，快速稳定地实现数据库的合理优化，最后建立数据库视图，测试最终结果。因此，这样一种处理数据转移的方法，不仅满足本文项目本身的需要，也对普遍存在的两个不同数据库之间的数据交互处理问题起到一定作用，有助于用户更高效的使用数据库数据资源。

### 参考文献：

- [1] 龙 帅. 深入浅出 SQL Server 数据库开发[M]. 北京: 中国青年电子出版社, 2006.
- [2] (美) Dusan Petkovic. Microsoft SQL Server 2005 初学者指南[M]. 冯 飞, 薛 莹, 译. 北京: 清华大学出版社, 2007, 66-242.
- [3] Martin C. Brown. python 技术参考大全[M]. 康 博, 译. 北京: 清华大学出版社, 2002.
- [4] Martelli, A.; Alex Martelli. Python in a nutshell PYTHON 技术手册[M]. 南京: 东南大学出版, 2006.
- [5] 吴涛峰, 张玉清. 数据库安全综述[J]. 计算机工程, 2006 (32): 85-88.
- [6] Stefan Brass, Christian Goldberg. Proving the Safety of SQL Queries, Proceedings of the Fifth International Conference on Quality Software (QSIC' 05), 2005.

责任编辑 徐侃春