

文章编号: 1005—0523(2005)05—0067—04

网络化测控中 SVG 技术的应用

蒋先刚¹, 刘觉夫², 张红斌²

(华东交通大学 1. 基础科学学院; 2. 信息工程学院, 江西 南昌 330013)

摘要: 探讨在网络化测控软件中如何将测试结果曲线进行 WEB 发布, 并结合 SVG 和 XSLT 技术实现测试过程的实时图形监测。

关 键 词: SVG; 网络化测控软件; JScript; 数据池。

中图分类号: TP368.1

文献标识码: A

1 概述

1.1 测控软件中数据发布的现状

众所周知, 网络的发展极大地带动了相关软硬件技术的提升, 人们对信息的认知和需求, 已经不仅仅停留在数值和字符的表现形式上, 更多的信息需求者, 希望获得内容丰富、表现形式多种多样且具备较强互动性的图形信息, 所以, 作为信息的发布者, 如何将数字化的信息更加直观地发布出去, 已经成为一个迫切的应用需求。在工业测试这样一个特定的领域里, 如果能够通过 WEB 发布实时测试曲线, 那么位于网络中的任何一个结点都可以在第一时间获取最新的图形化测试结果。

目前来看, 测试结果曲线的发布有几种常用方法, 例如将测试结果曲线以位图的方式进行存储并压缩成合适的格式, 再通过 TCP/IP 协议传输给网络中发出查询请求的结点; 同时, 还可将测试结果曲线位图上传给 WEB 服务器, 供网络中的其他结点浏览; 上述方法在技术上是可行的, 但图像传输的可靠性差, 并且设计出来的系统不具备很强的交互功能。

SVG 技术的出现使得这些问题迎刃而解。SVG (Scalable Vector Graphic) 即可缩放矢量图形, 它提供给用户基于 XML 的图像, 这种图像既可以是静态的, 也可以是具有动画效果和交互性内容的。

1.2 SVG 的主要特点

SVG 采用简单高效的矢量指令, 即利用点和线的描述来绘制图形, 这使得 SVG 文档的下载速度大大提高; SVG 从较低分辨率的便携式计算机到较高分辨率的台式机, 再到高分辨率的打印机, 都能提供良好的视觉效果, 而不会出现锯齿和模糊; SVG 文档可以进行 DOM 操作, 脚本和编程语言通过 DOM 对 SVG 实施编程工作, 从而很方便地实现对 SVG 图形的交互。在此基础上再结合 ASP、JSP 等动态服务器技术即可完成与测试结点或远程数据库的数据访问, 实现 SVG 图形属性的动态修改, 达到 SVG 图形动态显示的目的。

2 基于 SVG 技术的网络化测控软件的系统构成

2.1 网络化测试系统的结构及工作原理

网络化测控系统主要由两部分组成, 第一部分是分布在网络中的测试结点, 图 1 中只绘制了一个测试结点, 其他测试结点的工作原理相同, 所有的测试结点利用 PC 中的测试软件完成对继电器(本测控软件最终应用到对小型电磁继电器的测试中)的测试, 获取测试数据, 并将测试数据传送到网络中的数据库服务器上。软件的第二部分是网络发布, 它包括两个功能, 其一完成测试结果曲线的 WEB 发布, 其二完成测试过程的实时图形监测。

收稿日期: 2005—12—18

作者简介: 蒋先刚(1958—), 湖南永州人, 华东交通大学基础科学学院教授。

中国知网 <https://www.cnki.net>

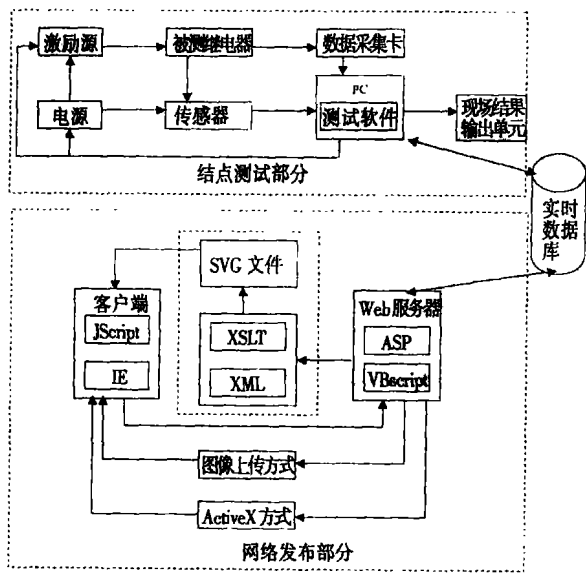


图 1 网络化测控软件的系统结构图

2.2 测试结果曲线 WEB 发布子系统的结构及工作原理

测试结果曲线 WEB 发布子系统的结构图如图 2 所示：

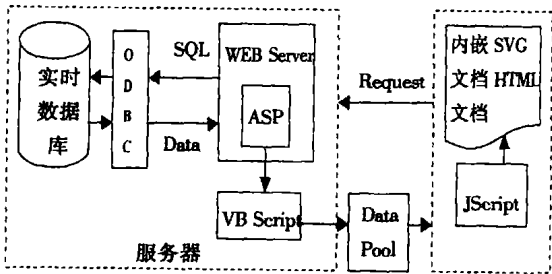


图 2 测试结果曲线 WEB 发布子系统结构图

该子系统被设计为一个三层的体系结构，包括服务器、客户机和中间层——“数据池”，运行在服务器端的是 ASP 和 VBScript 脚本，服务器上还存放了经过结点测试得到的所有数据，即图中的实时数据库。

而运行在客户机上的是内嵌 JScript 脚本的 SVG 源文件，JScript 脚本解析图中“数据池”——Data Pool 中的内容，得到用于发布曲线的所有数据。

介于客户机和服务器之间的 Data Pool 称为“数据池”，它的主要作用是从服务器端获取用于绘制动态曲线的数据，同时向客户端 SVG 文档提供一个可访问的接口，这个接口与 JScript 是相互兼容的，也就是说，SVG 文档中内嵌的 JScript 通过这个接口能够很容易地访问“数据池”中的数据。

客户机上的 SVG 文档获取了“数据池”中的数据之后，使用 JScript 对自身进行 DOM 编程，重新设置 SVG 文档的相关属性，从而实现测试结果曲线在客户机上的 SVG 显示。

客户机和服务器发生联系后，都是通过“数据池”来彼此通信的。如果有多个客户机向服务器发出测试结果曲线的 SVG 显示时，在服务器端只需维系一个很小的“数据池”开销就可以了，所以，这种体系结构不会因为客户机数量的猛

增而降低服务器端的运行性能。

2.3 测试过程实时图形监测子系统的结构及工作原理

测试系统实时网上监测子系统的结构图如下所示：

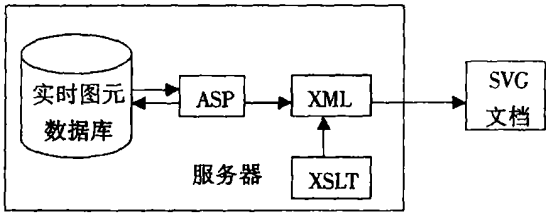


图 3 实时网上监测子系统结构图

实时网上监测就是将测试结构和数据以 SVG 的方式在 WEB 上发布出去，例如测试过程中一个具体的测试电路图，SVG 图形可以分解为许多最基本的图元，如线段、矩形、椭圆、圆等。利用实时图元数据库存放组成测试过程实时图形的所有基本图元，即分别建立线段数据表、矩形数据表、椭圆数据表等来存放相关实时图元。例如在线段基本表中下面几个数据字段必不可少：线段的起始点坐标、线段的终结点坐标、线段的宽度、线段的颜色以及对线段的简要文字说明。

软件的主要运行过程是：首先通过服务器端的 ASP 脚本取出图元数据库中存放的所有实时图元，然后将这些图元依次转存为 XML 文档中的一个结点，每个结点又都分别建立相应的子结点来存放从数据库中取出的描述该图元的属性值。例如，某线段在 XML 文档中的存储格式为：

```
<Line>
  <Color>Red</Color>
  <Width>2</Width>
  <Start_Point_X>30</Start_Point_X>
  <Start_Point_Y>24</Start_Point_Y>
  <End_Point_X>50</End_Point_X>
  <End_Point_Y>27</End_Point_Y>
</Line>
```

存放实时图元的 XML 文档生成之后，就可借助已经编写好的 XSLT 文档将这个 XML 文档转换为 SVG 文档。这个 SVG 文档就是在客户机端可以查看的实时测试过程的图形表示了，在 XSLT 文档中设置了几个转换 XML 中图元结点的模板，分别是转换线段的模板、转换矩形的模板、转换圆的模板和转换椭圆的模板。利用这几个模板就可完成数据到图形的转换工作。

3 SVG 技术的实际应用

3.1 测试结果曲线 WEB 发布子系统的设计

根据 2.1 中对该子系统结构的分析可知关键问题有 3。1. 使用 ASP 正确地访问实时数据库，获取测试数据；2. 将数据存入“数据池”；3. 在客户端动态地显示 SVG 文档。

3.1.1 访问实时数据库

在服务器端运行的 ASP 中写入对数据库操作的 VB Script 脚本即可. 可以选择多种方式连接数据库, 如指定 ODBC 中的数据源实现连接, 或利用 ADO 建立与数据库的动态连接.

3.1.2 操作“数据池”

“数据池”是数据从服务器端传送到客户端 SVG 文档中的唯一通道, 所以, “数据池”的操作非常重要, 它是 SVG 能正确显示的关键. “数据池”的操作是双向的, 对于服务器端, 必须在 ASP 中编写脚本, 完成将查询出的测试数据存入“数据池”, 而对于客户端则必须想办法正确地读取“数据池”中的内容.

“数据池”作为一个重要的媒介, 它的选取变得非常重要, 有几种方法可供选择, 例如在服务器端设置一个缓冲区, 由它充当“数据池”, 但考虑到数据量不是很大, 使用缓冲区是一种浪费, 所以, 此方法不适合中小测控系统的设计. 在服务器端使用短小精悍的小文件可适应于快捷的中小型测试系统.

我们在服务器端选择文本文件作为“数据池”, 因为服务器端的 ASP 脚本使用 FileSystemObject 组件可以非常容易地操作文本文件, 而 SVG 文档也可使用脚本对文本文件实现读取. 所以, 第一步需将数据库中查询得到的有关测试结果数据实时地存放在服务器端的文本文件中. 脚本代码如下:

```
<%  
'利用 Request 对象获取窗体间传递的动态数据, 本例为吸  
合时间数据.  
  
DrawTime = Request.querystring("DrawTime");  
DrawTime78 = Request.querystring("DrawTime78");  
ReleaseTime = Request.querystring("ReleaseTime");  
ReleaseTime78 = Request.querystring("ReleaseTime78");  
'建立一个 FileSystemObject 对象  
Set fs = Server.CreateObject("Scripting.FileSystemObject");  
'获取文件对象  
Time_file = Server.MapPath("/ActionTime.txt");  
'以对文件写的方式启动 actiontime.txt 文件  
Set txt = fs.OpenTextFile(Time_file, 2, true);  
'将吸合时间和释放时间分别写入 actiontime.txt 文件之中  
txt.WriteLine(DrawTime); txt.WriteLine(ReleaseTime);  
txt.WriteLine(DrawTime78); txt.WriteLine(ReleaseTime78);  
txt.Close;  
%>
```

对于客户端的每一个数据访问, 都由一个文本文件充当活动的“数据池”, 当访问结束后, 该文本文件将被自动删除, 这样, 在保证测试数据正确到位的同时, 又不会使服务器端因为文本文件的增多而出现数据混乱.

3.1.3 动态显示 SVG

如何实现 SVG 的动态显示呢? 我们还必须借助 JScript 脚本的功能和 SVG 文档开放的 DOM 接口. 在 SVG 文档中嵌入 JScript 脚本就可以很容易地对 3.1.2 中公共“数据池”一

文本文件进行操作. 而 SVG 的 DOM 接口又使得脚本可以方便地对 SVG 文档进行任意修改, 即用“数据池”中获取的数据来重新设置 SVG 文档的相关属性, 完成 SVG 的动态显示.

1) 将数据池中的内容读出, 代码如下:

```
<Script language = "Javascript"><  
'建立一个 FileSystemObject 对象  
var fso = new ActiveXObject("Scripting.FileSystemObject");  
'以只读的方式打开存放动态数据的文本文件  
var a = fso.OpenTextFile(Server.MapPath("/ActionTime.txt"));  
  
'逐行读出所有的动态数据, 总共四个测试数据  
var Draw_Time = a.ReadLine(); Release_Time = a.ReadLine();  
var Draw_Time78 = a.ReadLine(); Release_Time78 = a.ReadLine();  
a.Close();  
>  
</Script>
```

2) 对 SVG 文档进行动态修改. 使用 SVG 的 onload 事件触发 Main 函数实现. 代码如下:

```
Function Main(evt){  
var svg, root;  
'创建当前活动 SVG 文档的实例  
Svg = evt.getTarget();  
'获取 SVG 文档的根结点  
Root = svg.ownerDocument();  
'变换逻辑坐标为物理坐标  
X_Point1=Draw_Time * 96; X_Point1=Math.floor(X_Point1);  
X_Point2=Draw_Time78 * 96; X_Point2=Math.floor(X_Point2);  
X_Point3=Release_Time * 96; X_Point3=Math.floor(X_Point3);  
X_Point4=Release_Time78 * 96; X_Point4=Math.floor(X_Point4);  
  
'修改第一图形区的动态曲线  
rect1_1 = root.getElementById('rect1_1');  
rect1_1.setAttribute("width", X_Point1);  
rect1_2 = root.getElementById('rect1_2');  
rect1_2.setAttribute("x", X_Point1+30);  
rect1_3 = root.getElementById('rect1_3');  
rect1_3.setAttribute("x", X_Point1+30);  
  
'其他修改第二、三和四图形区的动态曲线, 代码省略  
}
```

通过这两步就可完成 SVG 的动态显示, 继电器动作时间测试曲线的 SVG 图形显示如图 3.

3.2 测试过程实时图形监测子系统的设计

根据 2.3 中对该子系统的分析可知,问题的关键是如何编写将 XML 文档转换为 SVG 文档的 XSLT 文档,其中,各种图元转换的模板尤其重要,下面是 XSLT 文档中转换图元线段的模板,其他图元的转换模板类似.

```
<xsl: for-each Select="Line"/>
<xsl: variable name="Start_X" Select="Start_Point
_X"/>
<xsl: variable name="Start_Y" Select="Start_Point
_Y"/>
```

```
<xsl: variable name="End_X" Select="End_Point
_X"/>
```

```
<xsl: variable name="End_Y" Select="End_Point
_Y"/>
```

```
<xsl: variable name="Color" Select="Color"/>
<Line x1="{ $Start_X}" y1="{ $Start_Y}" x2=
"{ $End_X}" y2="{ $End_Y}" Style=" stroke: { $Col-
or};" />
```

在客户端每隔 2 秒刷新显示一次,就可以获得测试过程的实时图形显示,以方便地监测测控的整个过程.

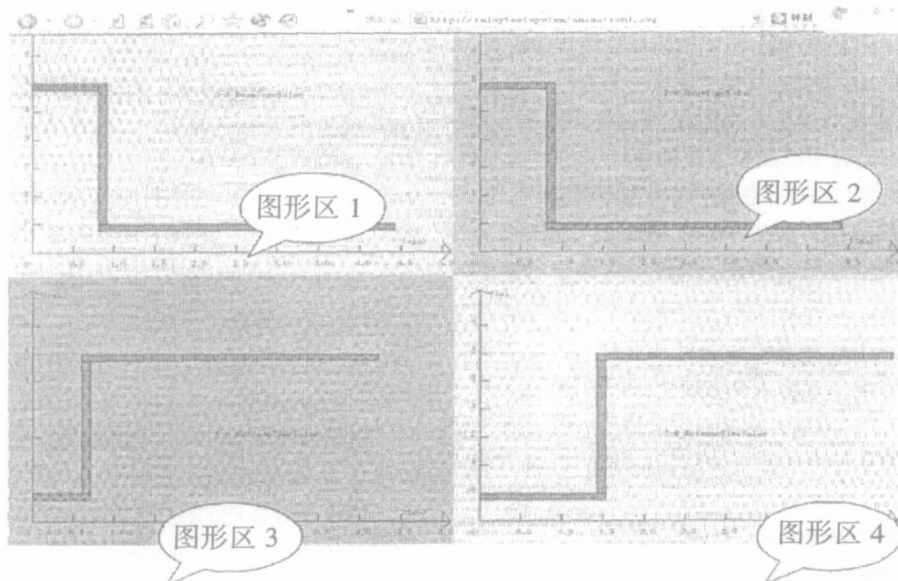


图 4 继电器动作时间测试曲线的 SVG 动态显示

4 结束语

本文介绍的 SVG 技术的应用方案已经在网络化测试系统的 WEB 数据发布模块中得到了很好的使用,系统运行中的测试结果曲线可以实现 SVG 的动态显示,系统可提供直观、可靠、及时的 SVG 图形显示.SVG 技术的简单方便使得越来越多的 WEB 图形系统开始使用它,SVG 技术为推动网络化的测控系统的实现提供了一个良好的图形开发工具.

参考文献:

- [1] 陈传波,吴方文.基于 SVG 的图元对象描述模型的研究[J].华中科技大学学报,2002,30(10):50—52.
- [2] 吴敏.使用 XSL 实现 XML 到 SVG 文件的转换[J].计算机应用,2002,22(5):102—103.
- [3] 陈传波,赵婷.SVG 与 XML 的集成技术在动态 WEB 图象上的应用[J].计算机工程与科学,2002,24(2):11—14.
- [4] 陈玉萍,蒋先刚.工业控制图形的不同格式文件转换的程序设计[J].华东交通大学学报,2001,18(4):51—55.

The Application of SVG in Networked Measurement System

JIANG Xian-gang¹, LIU Jue-fu², ZHANG Hong-bin²

(East China Jiaotong Univ., 1. School of Natural Science; 2. School of Information Engineering, Nanchang 330013, China)

Abstract: This paper introduces how to release test graphs on WEB in a networked measurement system, it also tells how to monitor real-time test procedure by the integration of SVG and XSLT.

Key words: SVG; networked measurement system; JScript; data pool