

# 基于 SVG 的 Web 远程实时监测客户端研究 \*

杨 斌, 张利欣, 章立军, 时 鹏

(北京科技大学 国家材料服役安全科学中心, 北京 100083)

**摘 要:** 针对现有基于 Web 的监测系统客户端技术在实时性方面的不足, 利用可缩放矢量图形 (SVG) 和 Ajax 技术设计了一种新的基于 Web 的远程实时监测客户端系统。该系统利用 SVG 技术实现图表绘制, 利用 Ajax 进行数据通信和传输, 完成了一套完整的基于 Web 的图表组件, 并在振动监测系统中得到了实现, 具有良好的实时性。

**关键词:** 可缩放矢量图形; 图表组件; Web; 远程实时监测客户端

中图分类号: TP393 文献标志码: A 文章编号: 1001-3695(2010)06-2144-03

doi:10.3969/j.issn.1001-3695.2010.06.042

## Web remote real-time monitoring client based on scalable vector graphics

YANG Bin, ZHANG Li-xin, ZHANG Li-jun, SHI Peng

(National Center for Materials Service Safety, University of Science & Technology Beijing, Beijing 100083, China)

**Abstract:** Based on the analysis on real-time problem of existing Web remote real-time monitoring client, proposed a new client system which utilize the SVG and Ajax. Used the SVG to draw the real-time chart, and used the Ajax technology to exchange data between client and server. Furthermore, implemented a set of chart component including SVG and Ajax, and used to vibration monitoring system. The system performs real-time well.

**Key words:** SVG (scalable vector graphics); chart component; Web; remote real-time monitoring client

## 0 引言

随着计算机技术和网络技术的不断发展, 大型装备的状态监测技术也发生了深远的变化, 逐步向实时、远程、分布式和网络化的方向发展。现有的远程实时在线监测系统普遍采用 C/S (客户端/服务器) 和 B/S (浏览器/服务器) 模式。其中 B/S 模式结构提供了系统更大的灵活性与开放性, 便于系统的扩展、维护以及管理。大大提高客户端性能是关系到整个监测系统运行效果的重要因素, 因此受到了极大的关注<sup>[1-3]</sup>。

但 Web 通信采用的是 HTTP 协议, 该协议的工作模式是请求/响应模式, 且无连接, 所以 Web 服务器无法实时地将服务器中的数据主动传送给 Web 浏览器。针对该问题, 现有的解决方案主要采用以下两种主要技术: a) 通过开发 ActiveX 控件扩展浏览器前端功能。该方式利用 ActiveX 与应用服务器通信, 实时获取最新的数据, 然后在客户端显示出来。b) 通过开发 Applet 来扩展浏览器前端功能, 从而实现基于 Web 的远程实时数据监测。该方式与 ActiveX 类似, 只是在实现技术上采用的是 Java 技术<sup>[4,5]</sup>。Web 远程实时监测除了考虑数据传送的实时性外, 还需要考虑实时数据在浏览器的显示问题, 特别是各种图表如线图、饼图、直方图等显示, 因为这些图表在生成和显示时会消耗浏览器大量的资源, 很容易造成显示上的不连贯现象及延滞现象。

本课题在一些相关研究成果的基础上, 针对监测系统的功

能需求, 设计和实现了一套基于 SVG<sup>[6]</sup> 和 Ajax (asynchronous JavaScript and XML)<sup>[7]</sup> (用于数据的获取) 技术的 Web 动态图表组件, 用于构建基于 Web 的远程实时监测客户端。

## 1 SVG 技术

SVG 是一种基于 XML 文档的开放的矢量图形描述语言, 用于描述二维矢量图形和矢量、点阵混合图形的标志语言。与位图不同, SVG 的矢量图形不是用大量的单个像素建立图像, 而主要用数学公式对物体 (如圆、线等) 行为描述, 从而记录图形信息, 并根据图形信息进行绘制。SVG 的关键是如何通过计算机指令和数学公式来描述物体。W3C 组织定义了 SVG 的相关规范, 其中包括 SVG 的特征、语法和显示效果, 包括模块化的 XML 命名空间 (nameSpace) 和 SVG 文档对象模型 (document object model, DOM)。

SVG 元素是指示如何绘制图像的一些指令, 阅读器 (viewer) 解释这些指令, 把 SVG 图像在指定设备上显示出来, 同时 SVG 可以使用样式表 CSS、Script 脚本和 DOM 进行控制。SVG 作为新一代的图形处理技术主要有以下几方面的优势: a) 文件尺寸小, 方便 Web 页面下载; b) SVG 图形格式可以用来动态生成图形, 可以 SVG 动态生成具有交互功能的数据图形, 嵌入网页中, 并显示给终端用户; c) 可以用脚本控制与客户的交互事件; d) SVG 图形格式可以方便地建立文字索引, 从而实现基于内容的图像搜索; e) 基于文本的格式, 可以轻松地与其他

收稿日期: 2009-11-15; 修回日期: 2009-12-23 基金项目: 国家高技术研究发展计划资助项目 (2007AA04Z169); 国家自然科学基金资助项目 (50674010)

作者简介: 杨斌 (1979-), 男, 湖南长沙人, 博士后, 主要研究方向为设备远程状态监测、无线通信 (jpuyang007@gmail.com); 张利欣 (1977-), 女, 讲师, 硕士, 主要研究方向为图像处理、远程通信; 章立军 (1978-), 男, 副研究员, 博士, 主要研究方向为信号处理、状态监测、故障诊断; 时鹏 (1977-), 男, 讲师, 博士, 主要研究方向为数据挖掘、数据库技术、计算机通信。

Web 技术集成。

SVG 是图形、图像和文字的整合体。根据功能不同,SVG 的主要对象可以分为基本要素和页面描述功能两大类。SVG 除了支持 HTML 中常用的标记,如文本、图像、链接、交互性、CSS 的使用、脚本外,还提供了大量针对图形、图像、动画的特定标记。如图 1 所示,SVG 定义了六种基本形状,同时 SVG 定义了直线、斜线、圆弧和贝塞曲线等绘图路径指令,通过这六种基本形状和不同的路径进行组合,则可以形成任何图像。

## 2 基于 SVG 的 Web 实时监测客户端关键技术

### 2.1 SVG 数据的嵌入与控制

虽然 SVG 文档本身就可以被所支持的 Web 浏览器(需要安装 SVG viewer 插件)直接显示,但通常应用系统需要将其嵌入在 Web 页面中使用。同时将 SVG 嵌入在 Web 页面中也使得 SVG 的显示更加容易,有些浏览器由于用户的设置可能会拒绝打开一个 SVG 文件,但显示作为 Web 页面一部分的 SVG 图像将不会有问题。通常使用 HTML 的<EMBED>标签将 SVG 图形数据文件嵌入在 Web 页面中。

基于 SVG 的动态图表效果(类似动画效果)是通过 JavaScript 脚本的控制来实现的。JavaScript 通过 SVG 的 DOM 接口与 SVG 文档联系在一起,控制 SVG 的元素及属性,可以动态修改 SVG 元素的属性,从而生成特定的动态效果。SVG 标准允许将 JavaScript 脚本代码以两种方式来实现:a)使用 Script 元素将 JavaScript 脚本内嵌在 SVG 文件中;b)使用 Script 的 xlink:href 属性从 SVG 文件之外连接 JavaScript 脚本文件。

在 JavaScript 下可以通过 DOM 定义的接口遍历访问在 SVG 文档中的 XML 树中的各个节点。通过 JavaScript 脚本可以对找到的节点属性重新赋值,甚至还可以在当前文档中删除节点或添加新创建的节点。JavaScript 在接收到服务器端的实时数据后,使用 DOM 进行解析返回的 XML 文本,获取元素的 ID、属性名称、属性值。根据这些信息来动态地改变 SVG 中对应元素的显示属性,最后使 SVG 中的元素反应出数据的变化情况,实现实时监测的效果。

### 2.2 SVG 文档的数据结构设计

从本质上来说,SVG 文档是 XML 文档,SVG 文档具有 XML 文档的所有属性。根据 SVG 的文档结构规范,普通的 SVG 文件由文件版本、文件类型定义部分、元素定义部分和绘图部分组成<sup>[8]</sup>。在一个复杂的 SVG 图形中可以使用图层元素对 SVG 进行结构和层次的划分。在 SVG 文档中,可以使用分组元素<g>来描述一个图层。图层作为一个容器对象,将各种相关的图元对象组合和管理起来,更好地表达了需要构建的图表的数据结构层次。针对监测系统中图表的构造特征,如图 2 所示,可以把 SVG 文档划分为以下几个层次:a)背景层,主要负责构建图表的背景和一些特殊效果,如曲线图的背景网格;b)坐标层,主要用于图表中一些坐标轴的显示,如 X 轴、Y 轴及其对应坐标等;c)动态图元对象层,这里的基本图元是指用于动态显示信号数据的文字、棒图、曲线或指针等,同时这些图元对象将设计为可以通过 JavaScript 脚本进行动态控制的组件;d)文字描述层,用于显示图表中当前的一些特定属性(如当前动态曲线对应的名称)或相关的图元注释等;e)用户交互层,用于向使用者提供交互功能的 SVG 控制部分,这些交互功能包括图表的放大与缩小、特定点的选取和显示等。通过层的 Id 属性可以访问和控制改变,从而可以在层中动态创建各种

图形和图元对象。

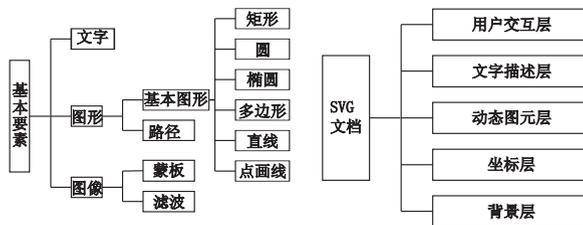


图1 SVG基本要素的结构图

图2 SVG文档的层次结构图

### 2.3 服务器端的 SVG 控件

SVG 的图形数据文档可以由服务器端的服务程序生成(如 C#或 Java 等)。当客户端向服务器请求 SVG 文件时(将在 embedded 标签中 SVG 源指向服务器端的执行文件),可以由服务器端的程序根据相关参数自动生成 SVG 文档再传给 Web 客户端。利用服务器端的服务程序自动生成 SVG 图形数据文件最大好处在于可以利用服务器端编程语言的强大功能实现非常复杂的 SVG 控件。

基于 SVG 的图表组件可以分为两个逻辑层面:a)描述图表数据结构和动态行为的抽象数学模型;b)用于显示各个图元对象的 SVG 的数据文件,即图表的动态实现。为了能够更加有效地控制 SVG 文档数据结构和图元显示,可以将图表的数学抽象模型与 SVG 的实现部分进行分离,使它们都可以独立地变化,从而可以快速地实现多样的图表形式。

这里参照上述的分离思想将整个 SVG 的控制模块设计为两个抽象类(C\_Chart 和 C\_Series)和两个实现类(C\_SVGShapes 和 C\_SVGElements)。其中:C\_Chart 和 C\_Series 系列类用于控制各个图表的数学抽象模型;而 C\_SVGShapes 和 C\_SVGElements 类则用于实际的 SVG 数据文档部分。在实际图表的创建过程中只需要使用 C\_Chart 和 C\_Series 类向外部提供抽象接口,其内部则调用了 C\_SVGShapes 和 C\_SVGElements 的功能输出实际的 SVG 数据文档。

C\_Chart 类提供了创建图表组件的基本容器。C\_Chart 类主要有两个功能:a)通过 C\_Chart 可以指定整个图表的大小和图表背景的基本属性;b)管理图表中的各个动态元素对象,即维护动态图元的列表和显示的相对属性等。C\_Chart 的 UML 类图如图 3 所示。其中函数 GetSVGDoc 返回当前生成的 SVG 图形的数据文档,在 Chart 配置完成后通过此函数获取完成的 SVG 文档数据。函数 CreateBaseSVGDoc 用于创建 SVG 的基本文档(SVG 的 Head 部分);setBackground 用于设置图表的背景属性;SetGrid 用于设置图表的网格(不使用则不设置);AddASeries、DeleteASeries、ClearSeries 和 IndexOf 函数用于管理 Chart 中的动态图元对象。

C\_Series 系列类包括 C\_SeriesBase 类(C\_SeriesBase 是图元对象的抽象类)和其继承类 C\_SeriesLine 及 C\_SeriesBar(将来还可以扩展其他新的动态图元对象)。这些类和接口提供了创建动态图元对象,即曲线图中的曲线或者棒图中的矩形。C\_SVGShape 则提供了基本的形状绘图接口,通过调用这些接口,可以根据输入的参数生成绘制图形的 SVG 文档数据,如直线、折线、矩形和圆等。C\_SVGElements 用于其他 SVG 元素的创建,如层对象、文字元素等。

## 3 实时监测客户端的实现

本课题针对监测系统的功能需求,设计和实现了一套基于

SVG 和 Ajax(用于数据的获取)技术的 Web 动态图表组件。利用 SVG 和 Ajax 技术实现 Web 下的动态图表基本原理如图 4 所示。首先利用 SVG 来绘制基本的图表形状,然后通过使用 Ajax 数据层向 Web 服务器请求最新的实时数据,当接收到实时数据时,再通过 JavaScript 脚本调用 DOM 接口动态修改 SVG 图形,这样就实现了 Web 下的动态图表。

进行动态操作,而是对曲线图中所有的数据点进行动态操作。

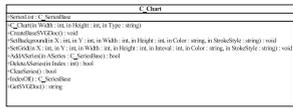


图3 C\_Chart的UML类图设计

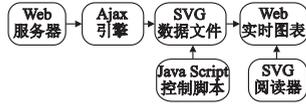


图4 SVG和Ajax技术实现动态图表的基本原理

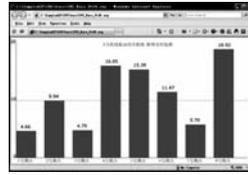


图5 特征值棒图组件运行效果图

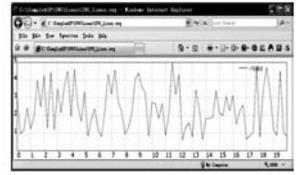


图6 特征值曲线运行效果图

在监测系统中经常需要使用的图表是振动特征值的棒图显示、振动的波形数据的曲线显示以及信号数据的趋势曲线图显示。下面重点介绍这几种类型的图表的设计与实现。在设计这些图表组件之前最重要的任务是需要计算图形的比例因子,就是说在绘图时每单位的数据值在对应图形中的多少像素。比例因子可以用图表的中坐标轴的尺寸除以数据的最大值来获得。

#### 4 结束语

针对现有 B/S 监测系统客户端存在的实时性问题,本文研究了一种基于 SVG 的 Web 远程实时监测系统客户端系统。该系统中设计了一套基于 SVG 和 Ajax 技术基于 Web 的动态图表组件,并在此基础上实现了基于 Web 的远程实时监测客户端。系统利用 SVG 来绘制基本的图表形状,通过使用 Ajax 数据层向 Web 服务器请求最新的实时数据,当接收到实时数据后再通过 JavaScript 脚本调用 DOM 接口动态修改 SVG 图形,SVG 与 Ajax 的结合带来了一种全新的程序开发模式,提供了一种与桌面应用程序相同的丰富的互联网体验。新的远程实时监测客户端将很大程度上改变现有客户端系统的多种弊端,具有较好的应用前景。

##### 1) 特征值棒图组件

动态棒图的动态图元对象是矩形(SVG 的标签是 RECT)。动态生成棒图的基本模型是:通过给定的图表的宽度和矩形对象的个数(参数)可以计算出每个矩形和间隔所占的宽度,从而确定每个矩形的起始点和宽度。矩形的高度由实际数据值(乘以比例因子)控制,而矩形的颜色可以从颜色数组中依次选取。在创建每个矩形的同时,在矩形的下方创建其文字描述对象并插入文字层。棒图组件的实际运行效果如图 5 所示。

#### 参考文献:

##### 2) 信号趋势曲线图的设计与实现

曲线图的动态图元对象是折线(SVG 的标签是 POLYLINE)。动态生成曲线图的基本模型是相对简单,即根据相关参数设定 POLYLINE 的属性,然后根据实际数据值(乘以比例因子)修改其 POINTS 属性。在曲线图的图元对象中还需要有一个数据点队列,一方面这个数据队列保存了曲线图中所有数据点的坐标值;另一方面通过这个数据点队列可以动态地添加、修改和删除数据点,从而实现动态曲线的绘制。趋势曲线图的设计运行效果如图 6 所示(注释表示测点 1 的均方根值)。

振动波形曲线图的设计和实现与上述的趋势曲线图是完全一样的,唯一的不同在于振动波形曲线图不是对单个数据点

[1] 周丹晨,殷国富,龙红能,等. 基于 Web 的制造信息集成管理系统[J]. 计算机集成制造系统,2003, 9(2):962-100.  
 [2] 饶志敏,余阳,李长森. 基于 Web Portal 的工作流客户端的研究与设计[J]. 计算机应用研究,2007,24(4):212-214.  
 [3] 陈忠,郑时雄. 基于 Web 技术的多级智能监测及诊断体系结构[J]. 华南理工大学学报,2005,33(8):11-15.  
 [4] 罗培,黄辉先,马茜. JNI 和 ActiveX 技术在组态监控系统网络化迁移中的应用[J]. 计算机应用研究,2007,24(12):266-268.  
 [5] 童雪娟,李霖,尹作为. 基于 SVG/Java Applet 的地图 Web 发布技术[J]. 计算机工程,2005,31(11):62-64.  
 [6] 刘啸,毕永年. 基于 XML 的 SVG 应用指南[M]. 北京:科海集团公司,2001.  
 [7] 柯自聪. Ajax 开发精要—概念、案例与框架[M]. 北京:电子工业出版社,2006.  
 [8] 张晓峰. SVG 在网络化测试曲线实时监控系统中的应用研究[D]. 青岛:中国海洋大学,2006.

(上接第 2133 页)

#### 参考文献:

[1] ERNST M D. Dynamically discovering likely program invariants[D]. Seattle: University of Washington, 2000.  
 [2] 刘磊. 程序分析技术[M]. 北京:机械工业出版社,2005.  
 [3] KHURSHID S, MARINOV D. TestEra: a novel framework for automated testing of Java programs[C]//Proc of Automated Software Engineering. Washington DC: IEEE Computer Society, 2001.  
 [4] SULLIVAN K, YANG Jin-lin, COPPIT D, et al. Software assurance by bounded exhaustive testing[C]//Proc of ACM SIGSOFT International Symposium on Software Testing and Analysis. New York: ACM Press, 2004:133-142.  
 [5] SEN K. Concolic testing[C]//Proc of the 21th IEEE/ACM International Conference on Automated Software Engineering. New York: ACM Press, 2007:571-572.  
 [6] HOARE C A R. An axiomatic basis for computer programming[J]. Communications of the ACM,1969,12(10): 576-585.  
 [7] ERNST M D, COCKRELL J, GRISWOLD W G, et al. Dynamically discovering likely program invariants to support program evolution[J]. IEEE Trans on Software Engineering, 2001,27(2):99-123.  
 [8] BOSHERNITSAN M, DOONG R, SAVOIA A. From Daikon to agitator: lessons and challenges in building a commercial tool for developer testing[C]//Proc of International Symposium on Software Testing and Analysis. New York: ACM Press, 2006:169-180.  
 [9] BOYAPATI C, KHURSHID S, MARINOV D. Korat: automated testing based on Java predicates[C]//Proc of ACM International Symposium on Software Testing and Analysis. New York: ACM Press,2002: 123-133.  
 [10] SEN K, AGHA G. CUTE and jCUTE: concolic unit testing and explicit path model-checking tools[C]//Proc of International Conference on Computer Aided Verification. Germany: Springer-Verlag, 2006: 419-423.  
 [11] DAVEY B A, PRIESTLEY H A. Introduction to lattices and order[M]. 2nd ed. Cambridge: Cambridge University Press, 2002.