

1 基于XML的BOM清单

物料清单(BOM)是一张列表,包含着生产每单位产品所需要的所有部件、组件、零件和原材料等信息,呈现出层次结构的特征。BOM表包含了产品中所有零部件信息和零部件的层次结构信息。作为ERP系统的基础和企业组织生产的依据,BOM必须完整准确地反映产品的结构及其组成元素。它应满足以下条件:

- (1) 完整性: 应该包括该产品的所有零部件;
- (2) 唯一性: 统一零部件应该只有一种表述;
- (3) 层次性: 零部件应该有明确的层次关系;
- (4) 扩展性: 当产品结构发生变化时,BOM应该比较容易调整;
- (5) 可读性: 产品机构应该具有良好的可读性。

传统的BOM表通常采用关系数据库来存储,这种存储有一个弊端,那就是必须把具有层次结构的BOM转化为父子关系形式进而存储在二维表中。在本质上,关系数据库并不支持层次型的数据结构,从而使得对产品结构树的各种操作都必须由用户自行开发实现,这就使其在应用上受到了很大的限制。本文将采用XML来实现BOM表的存储。

XML是W3C积极制定和推行的数据标准,是一种具备自我描述能力的结构化数据。和关系数据库相比,XML具有更好的可扩展性、开放性、灵活性、自我描述性和可操作性以及规范的层次结构。利用XML来构造BOM,有如下优势:

- ① 层次结构是XML对数据的组织形式,因此用XML来描述产品结构极为简单方便;
- ② XML属于结构化数据,它的自我描述能力和灵活性使得产品结构的各个组成部分的独立存储成为可能,从而可以实现产品结构的组件化设计;

用SVG实现基于XML的BOM清单的图形显示

The Graphic Display of XML-Based BOM in ERP Using SVG

刘海川 陈培久 (大连理工大学管理学院 116023)

摘要: 本文首先介绍了基于XML的BOM清单格式,随后介绍了SVG的特点,然后具体探讨了如何采用SVG技术来图形显示基于XML的ERP系统的BOM清单结构。

关键词: SVG XML BOM 图形显示

③ XML所具有的通用编程接口DOM标准使得对产品结构的管理和操作变得很方便。

随着计算机技术的应用和普及,BOM应该包含更多的信息,比如声音、图形、动画等对象,从而使得BOM能够以更直观、更友好的方式来传递信息。本文研究了一种利用SVG技术实现BOM的图形显示的方法。

2 SVG技术

SVG(Scalable Vector Graphics,可伸缩矢量图形)是W3C组织为适应Internet应用飞速发展的需要而制定的一套基于XML语言的用来描述二维矢量图形和矢量/点阵混合图形的标识语言。是图形、图像和文字的有机统一。SVG提供了六种类型的对象。

矢量图形、图像、渐变填充、滤镜操作、可复用单元和文本,它可以对图形对象进行组合、样式化、几何变换和重新组合等操作。SVG通过使用元素和属性等标记来生成任意复杂的图像。SVG标准中既有专门用于矢量图形描述的标记,如矩形、圆、椭圆、直线、折线、多边形等,还可通过<PATH>标记进行Bezier曲线的定义和操作,同时可对相应路

径进行勾勒、填充、裁剪、蒙板和合成等一系列操作。从而大大丰富了图形图像的显示效果。

和目前因特网上常用的GIF、JPG等点阵图像文件相比,SVG与具有以下优点:

(1) 高质量图形。由于SVG是矢量图像,分辨率独立。因此文件的大小与图形的具体尺寸无关,只与图形的复杂程度有关,同时图形的显示尺寸可以无级缩放,变化后不影响图形的质量。

(2) 便于网络传输。SVG是基于XML的,是一种文本格式的图像,因此文件更小,受带宽的限制小,下载浏览的速度更快,特别适合网络传输。

(3) 交互性强,SVG图像基于XML,因此承袭了XML强大的动态交互功能,能够对用户操作做出亮度变化、动画、声效等响应。这一点是一般图形所不能达到的。

(4) 文字独立,便于检索。由于SVG图形内的文字以纯文本方式独立于图像而存在,因此便于实现对图形中的文字检索,同时还可以通过XML灵活的标记特性相结合,从而可以方便地进行文字的编辑与查询。

(5) 超链接性。SVG文档中的图形对象可直接与其他媒体(如图象,视频,音频等)进行链接,使图形元素成为一扇通向其他信息的大门。

(6) 中立性。由于SVG是一个正式定义的国际开放标准,因此,它不属于任何团体和个人。

(7) 跨平台功能。SVG与XML规范的无缝连接以及标记语言的跨平台性使得SVG可以很好地跨平台工作。

(8) 色彩适应性强。SVG图像具有一个1600万色彩的调色板,支持ICC(国际色彩协会色彩标准,支持RGB色彩模式和蒙板,从而使得SVG可以在不同设备上能够不失真地、准确地显示。

3 SVG 的图象显示

SVG文件的显示需要在浏览器中安装相应的插件,可以从Adobe公司的网站(<http://www.adobe.com/SVG/>)下载支持SVG的插件SVG VIEWER3.0,在本地机上安装,然后就可以在浏览器中查看相应的SVG图像了。

4 SVG 在基于 XML 的 BOM 清单图形显示中的应用

4.1 基于 XML 的 BOM 清单

一般而言, BOM设计主要考虑有两个因素:装配层次关系和产品属性描述,而层次结构是XML对数据的基本组织形式,产品属性描述可以用XML文档的节点属性来加以描述,因此根据XML规范,我们选择复杂的泵业产品为例,设计了如下基于XML形式的产品结构模型:

```
<?xml version="1.0" encoding="utf-8" ?>
<产品列表>
.....
<<产品 名称="EAP25K1-200" 编码
="EAP25K1-200">
    <部件 名称="过流部件" 编码
="EAP25K1-1">
```

```
<子件 名称="泵体" 编码="EAP25K1-1-1"></子件>
<子件 名称="泵盖" 编码="EAP25K1-1-2"></子件>
<子件 名称="叶轮" 编码="EAP25K1-1-3"></子件>
<子件 名称="泵体口环" 编码="EAP25K1-1-4"></子件>
<子件 名称="配对法兰" 编码="EAP25K1-1-5"></子件>
<子件 名称="O形圈" 编码="EAP25K1-1-6"></子件>
<子件 名称="标准件" 编码="EAP25K1-1-7"></子件>
</部件>
.....
</产品>
.....
</产品列表>
```

与一般的关系数据库的存储不同,该模型充分利用了XML的层次结构特点,以更加自然的方式来描述产品结构,从而使产品结构的可读性大大增加。

4.2 BOM 的图形显示

系统的体系结构如图1所示:

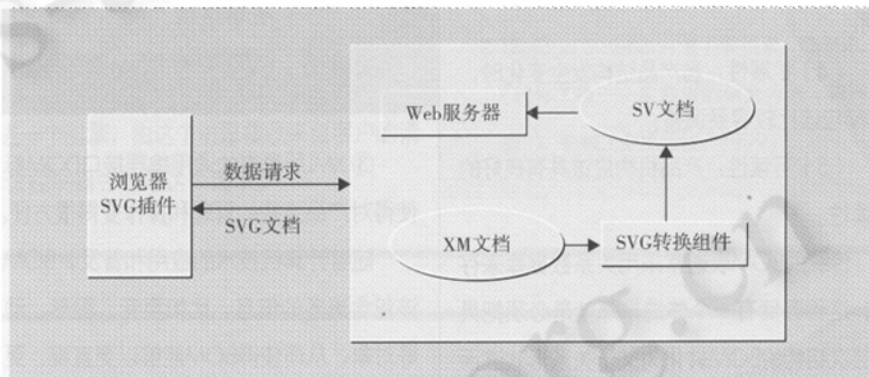


图1 系统体系结构

基本操作是根据用户要求在XML中查询相应的产品节点,利用自行开发的SVG转换组件,将相应的XML产品节点转化为SVG图形文件,然后保存该临时文件,接下来页面调用该临时文件,从而实现图形的显示。部分代码如下:

调用转换组件的部分代码如下:

```
Dim filename As String '定义字符串保存临时文件名
filename = Server.MapPath("../SVG/product.svg") '保存临时文件名
File.Delete(filename) '如果临时已存在,则删除
Dim mystream As New FileStream(filename, FileMode.OpenOrCreate) '创建新的文件流
Dim strwriterobj As New StreamWriter(mystream) '创建实例对象向流中写入字符
Dim xmlsvg As New xmltosvg() '定义新的转换类
Dim mysvg As String '定义字符串,保存生成的SVG图形文件
xmlsvg.Path = Server.MapPath("../XML/") '指定XML文件的位置
mysvg = xmlsvg.getsvg("00183-1") '指定要显示的产品的编码,从而得到相应的SVG图形文件
strwriterobj.Write(mysvg) '将图形文件保存
```

```
strwriterobj.Close()
```

关闭实例

要在页面中显示SVG图形,必须在页面中用插入对象的方法插入SVG文件,代码如下:

```
<embed src="product.svg" id="mysvg"
runat="server" type="image/svg+xml"></
embed>
```

生成的svg文件的代码如下:

```
<?xml version='1.0' encoding='utf-8'?> <!
DOCTYPE svg PUBLIC "-//W3C//DTD SVG
20000802//EN" 'http://www.w3.org/TR/
2000/CR-SVG-20000802/DTD/svg-
20000802.dtd'>
```

```
<svg width='1120' height='621'>
<rect x='10' y='10' width='60'
height='10' style='fill: yellow'/>
```

```
<text x='10' y='20' style='fill: red'>
EAP25K1-200</text>
```

```
<line x1='40' y1='30' x2='85' y2='30'
style='stroke: RED;stroke-width:2'/>
```

```
<rect x='88' y='25' width='60'
height='10' style='fill: yellow'/>
```

```
<text x='88' y='35' style='fill: red'>过流
部件</text>
```

```
<line x1='118' y1='45' x2='163'
y2='45' style='stroke: RED;stroke-width:2'/>
```

```
<rect x='166' y='40' width='60'
height='10' style='fill: yellow'/>
```

```
<text x='166' y='50' style='fill: red'>泵体
</text>
```

```
<line x1='118' y1='58' x2='163'
y2='58' style='stroke: RED;stroke-width:2'/>
```

```
<rect x='166' y='53' width='60'
height='10' style='fill: yellow'/>
```

```
<text x='166' y='63' style='fill: red'>泵盖
</text>
```

.....

```
<line x1='40' y1='20' x2='40' y2='588'
style='stroke: RED;stroke-width:2'/>
```

```
</svg>
```

显示效果如右图2

所示。

用图形方式显示的BOM清单,可以使用户直观查看BOM的组成和结构,提高了BOM的可读性,也使其构造和维护更加容易。

5 结论

本文采用SVG来实现BOM清单的图形显示,提供了一种可视化BOM清单的表达方式,从而使BOM表能够以更加友好的方式显示给用户。SVG作为W3C组织正式推荐的图像格式,综合了矢量图形、位图图像和文字的优点,具有统一的标准和开放性,适于多种文字图像综合应用的领域。由于它具有体积小、可链接、可缩放等特点,必将得到越来越广泛的应用。



图2

参考文献

- 1 周强中、谈俊忠, SVG在WEBGIS中的应用[J], 计算机应用研究, 2003.1(108)。
- 2 原野、沈均毅、何妹, SVG技术在统计图表中的应用研究[J], 计算机工程与应用, 2002.17(33)。
- 3 侯宇、李素育, 基于XML的SVG技术及其应用[J], 计算机应用研究, 2002.5(136)。
- 4 吴敏, 使用XSL技术实现XML到SVG文件的转换[J], 计算机应用, 2002.5(102)。
- 5 吴冬梅、王树国、蔡鹤皋, 基于XML的制造业产品结构的有效实现[J], 组合机床与自动化加工技术, 2002.10(15)。
- 6 蓝骑士译, 矢量文本图形-SVG专题教程, <http://www.yesky.com/20010315/164991.shtml>.2003.5。