使用 Flex 呈现可缩放矢量图形和位图

级别： 初级

Sandeep Malik, 技术主管, IBM

2009 年 6 月 01 日

可缩放矢量图形（Scalable Vector Graphics，SVG）是图像领域内一种非常重要的技术。Adobe Flash Player 及 Flex 提供了对 SVG 的完全支持。不过，使用 SVG 创建复杂图形一直都十分困难。要让 SVG 与位图数据能够协同工作，您需要充分理解一些复杂的概念，比如矩阵旋转、透明性等。那么就请进入 Flex 吧。本文为您介绍了 Flex 内奇妙的 SVG 世界。通过矢量绘图就能创建定制图形并构建美观的组件。

简介

在图像领域，有两种主要的呈现技术占据着主导地位。位图呈现是第一个也是最为重要的一种技术，该技术已经存在数十年了，现在很多围绕它的技术和工具也都已经十分成熟。图像可以以多种格式显示，比如 *jpeg*、 *png*、*bmp* 等。设计人员借助先进的工具能够生成赏心悦目、细致入微的图像，并将其用在 Web 站点，让人耳目一新。像 Adobe® Photoshop 和 Image Editor 这样的工具都具有一系列复杂的算法，可应用于图像的位图，并能够更改图像的整体外观及给人的感觉。位图呈现已经得到了很好的应用，目前位图图像已经成为了所有 Web 站点设计的一个非常重要的部分。不过，就目前的情况而言，位图呈现还存在着两个很重要的缺陷：首先，它是一个静态图像，没有任何动画（注意，我们虽然可以通过使用 *gif* 格式来支持动画，但是这些动画大多数都是预先定义的，而且通常并不能对应于用户交互）。第二，它们只适合于特定的分辨率，或者最多只能适合于一组分辨率，在这些分辨率下，图像的美观不会受到影响。一旦分辨率改变，就会出现一些丑陋的矩形（像素块），这无疑破坏了图像的质量，削弱了它的光彩。因此，位图虽然能够生成不错的图像，但它最多只能满足在静态且屏幕分辨率固定这一特定条件下的某些需求。

随着对 Web 上动态内容需求的不断增长，让图像能够响应用户交互已经成为了一种必然。换言之，就是要求图像具有行为特性。可缩放矢量图形（SVG）恰巧能实现这一点。理解 “可缩放矢量图形” 一词的含义非常重要。矢量图形的意思是指所画的图像并非由一组彩色像素组成。矢量作图更像是用铅笔画画，先拿铅笔确定一些点，然后用直线、曲线、方框和椭圆框连接这些点。最后再用不同的颜色填充这些封闭的区域。因此，路径的概念对矢量作图非常重要。其中的一个好处是路径独立于屏幕分辨率。一般地，路径基于一种单位比例进行开发，而整个图形则可以按用户想要的任何分辨率呈现，所以即便分辨率改变，图像质量仍能保持不变。我们之所以称矢量图是 “可缩放的”，是因为它们可以缩放到平台（一般是 Web 浏览器）所支持的任意的分辨率。

但 SVG 的确具有一些局限性。它是一个全新的技术，并且这个工具还不太成熟，因此，所生成的图片的精细程度还达不到位图的水平。此外，为一个精细的图像定义复杂的路径并不容易。其结果就是，SVG 图形只能局限于圆圈、矩形等这类简单的图形。但这种情况*正在*改观。旨在提供对 SVG 的全面支持的新技术不断出现，而且工具也更为先进。

在本文中，我将对 SVG 进行详细的介绍，并会将其与位图相对比来展示它的优点。然后，我将着重讨论支持 SVG 的现有工具和技术，特别是那些开源技术和工具。这之后，我将介绍如何综合利用位图和 SVG 来创建既细致又 “生动” 的图形。最后，我将提供一个这样的例子并会带您亲历它的源代码。

SVG 特性简介

最近，SVG 势头强劲。Web 站点设计者开始考虑将 SVG 作为位图的一个主要替代品。几乎所有支持 AIR（Rich Internet Application）的新技术都会以这样或那样的形式为矢量图提供支持，并且 SVG 相对于位图的确有一些优点。首先，SVG 是一个基于文本的表示。一个 SVG 图像的完整描述都是通过一个符合 W3C 标准的 XML 格式表示的。这让 SVG 图像具有了 “可读性”，并且通过一个文本编辑器很容易就能对之进行更改。SVG 文件中的任何一条路径都可以通过解析 XML 或使用像 XQuery 或 XPath 这样的先进技术被定位。这就让 SVG 足以满足用户的需求。

举个例子，假设您正在制作一个老虎的 SVG 图像，并且希望在用户单击某个按钮时这个老虎会眨眼。在位图

中，我们一般会通过嵌入两个或三个相互叠加的图像的方法来实现这种效果。然而在 SVG 中，可以定位画出老虎眼睛外形的那个路径，然后用不同的颜色重新填充这个路径，这样就实现了老虎闭眼睛的外观效果！更妙的是，在一个 SVG 编辑器中，可以同时画两个区域（一个区域就是一个封闭路径）：睁眼和闭眼，并让后者透明。要让老虎眨眼，可以通过在相应的 XML 中定位这两个区域及其路径来在两个区域间交替透明性。最终生成了一个具有交互性的 SVG 图像，它会根据用户的交互改变行为。

通过操作路径，可以实现更多类似的交互性。实际上，SVG 允许嵌入 sciplet（ECMAScript），它可以包含很多事件，比如鼠标悬停、鼠标移出、单击等。SVG 的特性让它成为了图形技术的一个重要的选择，相对于位图，它有两个优点：

1. 用 SVG 可以得到一个独立于显示器分辨率的可缩放图像。
2. 要达到同样的效果，SVG 图像的大小要小于位图。

SVG 图像，以纯文本形式表示，可以使用一些常用的文本压缩算法，比如 gzip，对之进行压缩。这些压缩格式以 *.svgz* 文件表示，而未压缩的格式则以 *.svg* 文件表示。

支持 SVG 的开源工具和技术

很多开源工具和技术都为 SVG 图像设计提供了很好的支持。Inkscape（参见 参考资料）就是其中一个值得一提的工具。它可以免费使用，并且是创造优秀艺术作品的良好平台。Inkscape 不仅允许用户以 SVG 格式保存图片，而且还允许用户将图片保存为其他格式，比如 XAML（Microsoft® XAML）、PS（post script）、GPL（GIMP Palette）、ODG（OpenDocument Drawing）等。此外，还可以以位图文件导出图片。Inkscape 提供了一个功能丰富的编辑器，可以用梯度、贝塞尔曲线、3D 填充物制作精细图像。它绝对值得 SVG 的热衷者去看一看。

如今，几乎所有的最新技术都支持 SVG，比如 Microsoft 的 Silverlight、Sun® 的 JavaFX、OpenLazlo 和 Adobe Flex（或 Flash）。在本文中，我们会把重点放在 Flex 提供的支持上。Flex 是 Adobe Flash Player 的一个面向企业的版本，是面向启用了 RIA 的 Web 站点的一个理想平台。因为 Flex 的大多数特性都继承自 Flash，所以它能很好的支持矢量作图。Flex 中每个从基础类 `Sprite` 扩展而来的类都具有一个嵌入的图片对象。这个图片对象可被用来在 sprite 上进行矢量作图。例如，要画一个半径为 5 的圆，可以使用如下代码：

清单 1. 展示矢量作图基础知识的一个示例

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Canvas xmlns:mx="http://www.adobe.com/2006/mxml" width="400" height="300"
creationComplete="onCC()">
  <mx:Script>
  <![CDATA[
  private function onCC() : void {
  graphics.clear();
  graphics.lineStyle(2,0xffff00);
  graphics.drawCircle(100,100,5);
  }
  ]]>
 </mx:Script>
</mx:Canvas>
```

在这个例子中，我们已经从 Canvas（扩展自 Sprite）做了扩展，并且一旦 `creationComplete` 事件被 Canvas 触发，我们就可以绘制这个圆。请注意其中的 `graphics.clear()` 调用。在开始画新图前，要先清除之前的图片，这一点很重要。否则，新图就会呈现在旧图的上面，如此重叠画图最终会降低处理的速度。`graphics.lineStyle()` 设定了画图采用的线的类型。最后，`graphics.drawCircle()` 以 (100,100) 为中心、以 5 为半径画了一个圆。就这么简单。Flex 支持各种形状的并集或交集。假设有两个圆，若只想填充这两个圆非交叉的部分，可使用清单 2 所示的方法。

清单 2. 两个矢量图交集或并集的示例

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Canvas xmlns:mx="http://www.adobe.com/2006/mxml" width="400" height="300"
creationComplete="onCC()">
 <mx:Script>
 <![CDATA[ private function onCC() : void {
 graphics.clear();
 graphics.lineStyle(1,0,0);
 graphics.beginFill(0xcccccc);
 graphics.drawCircle(100,100,50);
 graphics.drawCircle(110,100,50);
 graphics.endFill();
 }
 ]]>
 </mx:Script>
</mx:Canvas>
```

图 1 显示了图像在浏览器内呈现后的样子：

图 1. 在浏览器内呈现的矢量图

此外，graphics 类还给出了可用来绘制诸如矩形、曲线、圆角矩形、线、三角等这类基础图形的其他便捷方法。这些方法应该可以满足一个想将它们用在 Web 站点设计中的应用程序开发人员的基本需求。然而，当需要画一些复杂的矢量图时，这些基础图形就不够用了。此时需要利用一些更先进的工具或框架。Degrafa（参见 参考资料）就是这样一个可用来创建具有交互性的复杂图像的开源框架。Flex 的未来版本 Flex 4.0 也将提供对复杂 SVG 图形的框架级支持。不利的一面就是为了使用它们，您必须要掌握这些新 API。不过，还有另外一种折衷的方法可以帮助您很好地使用位图和矢量作图。我将在下一个小节中做详细解释。

在 Flex 中同时使用位图和 SVG

假设我们想创建一个包含有三个区的测量组件：正常、告警和危险。这些区的整个范围在 210 度到 -30 度之间。这个范围被分成三个部分。“危险” 区在 210 度到 130 度间。“告警” 区在 130 度到 50 度间。“正常” 区在 50 度到 -30度间。下图显示的就是这个测量组件。

图 2. 测量组件

这是一个全梯度和全填充的静态图像。不过，我们需要给它一些 “活力”，我们要给它添加一个指针来显示基于一些事件的当前状态。这个指针需要有来自于用户的输入，此测量组件将基于此输入将指针放在恰当的区。仅使用位图很难实现这种类型的交互性，甚至不可能实现。但是，用矢量图就可以实现。首先，为使用矢量图的指针编写代码。然后从基类 UIComponent 扩展，并覆盖 updateDisplayList() 方法来绘制这个定制矢量图。

清单 3. 指针组件的基本设计

```
<?xml version="1.0" encoding="utf-8"?>
<mx:UIComponent xmlns:mx="http://www.adobe.com/2006/mxml">
 <mx:Script>
 <![CDATA[
 private var _angle : Number;
 // this angle must range from 210 deg to -30 deg:
 public function set angle(value : Number) : void {
 if(value > 210 || value < -30) throw new Error("Unsupported Angle Value "+value);
 _angle = value;
 inited = true;
 invalidateDisplayList();
 }
 private var inited : Boolean;
 public var center : Point;
 public var radius : Number;
 public function get angle() : Number {
 return _angle;
 }
 ]]>
 </mx:Script>
 </mx:UIComponent>
```

在显示指针前，这个指针组件需要声明三个所需变量。它们是 “angle”、“center” 和 “radius”。在设置值时，需要检查有效的角度范围。最后调用 invalidateDisplayList() 来重新画图。现在，我们来了解一下 updateDisplayList() 方法及其作用：

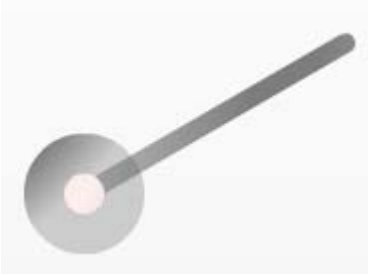
清单 4. 指针组件的主要画图代码

```
override protected function
updateDisplayList(unscaledWidth:Number, unscaledHeight:Number):void {
 // we need to first convert the angle to a convenient value,
```

```
since flex follows a different angle convention:
 var ang : Number = -angle*Math.PI/180;
 graphics.clear();
 graphics.beginGradientFill(GradientType.RADIAL,[0,0xcccccc],[1,1],
[20,200],null,SpreadMethod.REFLECT);
 graphics.drawCircle(center.x,center.y,30);
 graphics.endFill();
 if(inited) {
 graphics.lineStyle(10,0,1,false,"normal");
 graphics.lineGradientStyle(GradientType.LINEAR,[0,0xcccccc],[.5,.5],
[20,200],null,SpreadMethod.REFLECT,"rgb",.5);
 graphics.moveTo(center.x,center.y);
 var p : Point = getCoordinates(ang);
 graphics.lineTo(center.x + p.x,center.y + p.y);
 graphics.lineStyle(1,0,0);
 graphics.moveTo(center.x,center.y);
 }
```

对于一个 30 度的角，指针组件应该类似于图 3。

图 3. 30 度的指针组件



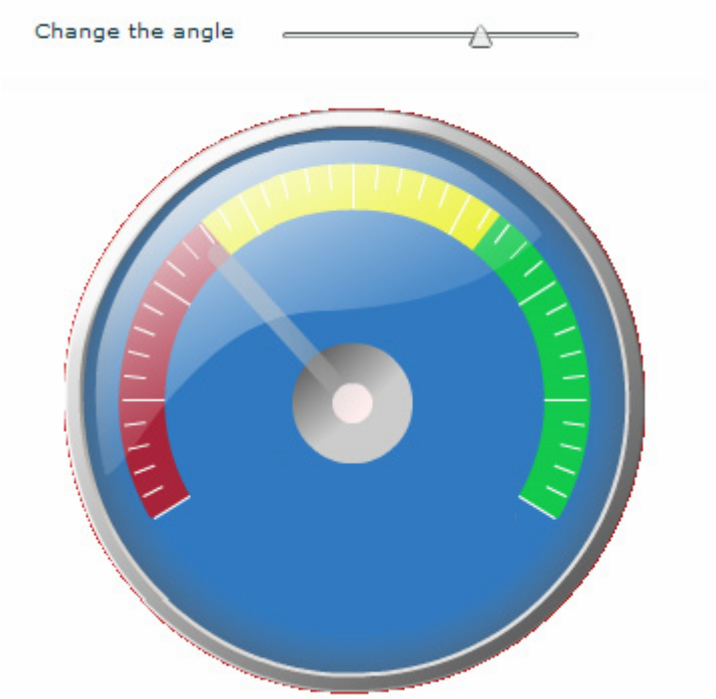
现在所要做的是重叠位图和这个矢量图，这样，这个测量组件就基本建好了。参见下面的代码片段：

清单 5. 测量组件

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
layout="absolute" xmlns:components="article2.components.*" backgroundColor="0xffffff">
 <mx:FormItem label="Change the angle" x="20" y="10">
 <mx:HSlider id="slider" minimum="-30" maximum="210" value="0"
liveDragging="true" change="needle.angle = slider.value"/>
 </mx:FormItem>
 <mx:Canvas y="50" backgroundImage="article2/images/gauge_skin.PNG"
width="362" height="310" id="gauge"/>
 <components:Needle id="needle" center="
{new Point(gauge.x+gauge.width/2,gauge.y+gauge.height/2)}" radius="100"/>
 </mx:Application>
```

图 4 是上面这个代码在执行时的一个屏幕快照。

图 4. 运行中的测量组件



至此，这个测量组件就完全建好了！可以下载下面的 有效 swf 文件。

结束语

本文对 SVG 和位图做了一个简短的介绍。您了解了 SVG 是如何为静态位图带来用户交互性的。在 Flex 中嵌入一个丰富的位图是很容易的（通过提供 backgroundImage property），添加一个带矢量图的组件就更容易了。本文中的示例为应用程序的设计或开发人员开启了一扇新的大门，在保持传统的位图制图的同时，又可以使用 Flex 添加 SVG 支持。

下载

描述	名字	大小	下载方法
本文中创建的 .swf 文件	Download.zip	279KB	**HTTP**

关于下载方法的信息

参考资料

学习

- 查阅有关 Scalable Vector Graphics 的 W3C 规范。
- 随时关注 developerWorks 技术活动和网络广播。

获得产品和技术

- 了解并下载面向不同平台的 Inkscape。

- 获得有关 Degrafa 框架的更多信息。
- 利用可直接从 developerWorks 下载的 IBM 试用软件 构建您的下一个开发项目。

讨论

- developerWorks blogs：加入 developerWorks 社区。

关于作者

Sandeep Malik 在印度 Pune 的 IBM India Software Labs 工作，是 IBM Cognos NOW! 的一名技术主管。他曾参与了 Cognos NOW! 新一代 UI 的设计和架构阶段以及记忆和实时流线化 OBI（Operation Business Intelligence）引擎。Sandeep 在重载图形、图表库、客户端流、非阻塞 I/O 以及异步系统方面具有大量经验。加入 IBM 之前， Sandeep 曾工作于网络安全领域，分析能够改变网络流量分布（比如瓶颈、蠕虫扫描等）的模型。在之前供职的一家公司，他还曾从事过 Servlets 2.4 规范的实现。在业余时间，他喜欢观看板球比赛，而且幻想着成为一名板球队员！