SVG Language (Scalable Vector Graphics)
For 2D Graphics in XML and Applications

Marin Vlada

University of Bucharest, 14 Academiei Street, RO-010014, Romania
E-mail: vlada@fmi.unibuc.ro, Web: www.ad-astra.ro/marinvlada

Abstract
The SVG technology is an open source copyrighted material of the W3C consortium and it is a language for 2D graphics within the XML (eXtensible Markup Language). The combination between SVG and JavaScript offers a powerful platform usable for interactive 2D graphics, comparable to the Flash and Java technologies. SVG offers XML graphics for the Web using three types of graphical objects: vector graphic shapes (lines and curves), images and text. Objects may be grouped, transformed and represented, dynamically and interactively. The SVG uses XML text standards, JPEG and PNG image formats, DOM (Document Object Model) for scripting and interactivity, SMIL for animation and CSS for styling. The present paper constitutes a presentation of the SVG and it also describes a few applications written in SVG.

Keywords: SVG Technology, 2D Graphics, XML, Document Object Model.

1. Introduction – Web 1.0 and Web 2.0

MOTTO: "Things to watch: SVG – Scalable Vector Graphics – at last, graphics which can be rendered optimally on all sizes of device" (TIM BERNERS-LEE, inventor of the World Wide Web www.w3.org/People/Berners-Lee)

The complexity of computer applications in various fields (including education), has powered the improvement of both operation systems and programming languages, as well as the improvement of technologies and platforms. New operation systems, new programming languages and new technologies have been conceived and developed. If the invention and putting into use of the microprocessor in the ’70’s meant a true revolution in the field of computer hardware architecture, the 90’s marked a true revolution in the field of both computer networks as well as the field of programming languages (through the advent of Java and JavaScript) and operating systems (Linux, Windows). Thus, Web technologies appeared. One should also mention the development and evolution of the C++ language which during the 80’s implemented and developed the object oriented model of programming (the objectual programming model is rooted in the SmallTalk or Lisp as well as other programming languages) and also object oriented programming (OOP – Object Oriented Programming) [8].
At the beginning of the 90’s, the HTML (Hypertext Markup Language) appeared a thing that determined the dissemination of static Web pages as well as an explosive development of the WWW (World Wide Web) system. The need to develop dynamic Web pages has determined the advent of various other technologies such as: JavaScript, JavaServer Pages (JSP), VBScript, PHP, ASP, Macromedia Dreamweaver, a.s.o., which were mainly meant for server side applications, while others were meant for client side applications. In the field of dynamic and interactive graphic applications the last 10 years were dominated by the Java and Flash technologies: Web 1.0 generation.

As a consequence of the developments offered by the XML (eXtensible Markup Language), JavaScript language, DOM (Document Object Model) for scripting and interactivity, SMIL (Synchronized Multimedia Integration Language) for animation and CSS for styling, in 2003, the W3C (World Wide Web Consortium – W3C) elaborates the 1.1 SVG (Scalable Vector Graphics) specification – http://www.w3.org/Graphics/SVG/.

This is an open source platform and is a consistent alternative to the Java and Flash technologies. „SVG is used in many business areas including Web graphics, animation, user interfaces, graphics interchange, print and hardcopy output, mobile applications and high-quality design” [7]. The authors of this platform are: Adobe, Agfa, Apple, Canon, Corel, Ericsson, HP, IBM, Kodak, Macromedia, Microsoft, Nokia, Sharp and Sun Microsystems.

The examples and graphical applications shall be viewed (using a compatible browser) with Firefox 1.5+, Opera 9 or Internet Explorer with Adobe SVG plug-in (Adobe SVG Viewer [1]). Since 2003, appears the generation Web 2.0: new technologies and Web services.

Using the SVG platform, in the field of education, a number of graphical applications for various sciences (mathematics, physics, information technology, chemistry, a.s.o) can be imagined and developed, so as to aid in the presentation of various phenomena, terms and concepts. The presentations can be made using a computer and a video projector installed in the classroom/room where pupils/students assist the lesson. The dynamic aspect and interactivity of the presentation make these presentations attractive. The teachers and IT specialists must cooperate, with the final purpose of developing such graphical applications. The classic textbook for a certain subject is unable to offer the level of interactivity and dynamics that are offered by the SVG, therefore it is desirable that the textbook is completed through the use of adequate software products (educational software), a software that will meet the requirements of pupils, students and educators. While a classical textbook constitutes the work of one or several specialized authors, a software product which is meant for educational use can only be completed through collaborative work of several specialists from various fields: education, IT, psychology, learning sciences and even pupils or students. Thus, lessons and classes shall be at the same time attractive and useful, to the purpose of obtaining the competencies that are taken into consideration through the adequate curricula.
2. Scalable Vector Graphics (SVG)

SVG is a platform for describing the 2D graphical applications within the XML language. The combination between the SVG and JavaScript offers a powerful platform for interactive 2D graphics, comparable to the Flash and Java technologies. The SVG platform offers XML graphics for the Web through three types of graphic objects: vector graphic shapes (lines and curves), images and text. The objects can be grouped, transformed and represented dynamically and interactively. SVG uses XML standards for text, the JPEG and PNG formats for images, DOM (Document Object Model) for scripting and interactivity, SMIL (Synchronized Multimedia Integration Language) for animation and CSS for styling. The SVG platform consists of two parts: a basic XML type file and API programming for the 2D graphic applications. „Key features include shapes, text and embedded raster graphics, with many different painting styles. It supports scripting through languages such as ECMAScript and has comprehensive support for animation” [12]. An in-depth familiarization with the HTML, XML as well as object oriented programming (OOP) will assist in a clearer understanding of the usage of the SVG specs [14]. The XML language (eXtensible Markup Language) is the language that offers a format for storing and transmitting data through a declarative description. The XML „grammar” includes XHTML (the XML version of HTML), SVG, MathML (the Mathematics Markup Language), ChemML (the Chemistry Markup Language) and GML (the Geography Markup Language).

As a scripting language, through the use of JavaScript and SVG Document Object Model (SVG-DOM), SVG is an extension of HTML DOM level 2, which is highly familiar to all Web developers. SVG elements can be characterized by animation through the use of Synchronized Multimedia Integration Language (SMIL).

Just like XHTNK and MathML, SVG is of XML type; all SVG files and documents share a .svg extension and can be edited using a plain text editor, (such as for instance Notepad). These files are not compiled; instead they are merely interpreted by the browser (Firefox 1.5+, Opera 9 or Internet Explorer with Adobe SVG Viewer).

Document Type Definition (DTD) is the description of elements and attributes (svg11.dtd) corresponding to the SVG type statements, which are subsequently used in graphic applications ("http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd"). The „root” elements of the XML documents are those XML tags that are interpreted by the browser. For instance, <html> for XML and <svg> for SVG. „Namespace” XML or "xmlns" attributes will perform a unique identification description of the SVG attributes using XML data. SVG applications need the following statements:

- xmlns = http://www.w3.org/2000/svg
- xmlns:xlink = http://www.w3.org/1999/xlink
For SVG version 1.0 DTD reference (svg10.dtd) is:

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
 "http://www.w3.org/TR/2001/REC-SVG20010904/DTD/svg10.dtd">
```

Development of graphic applications using SVG implies a knowledge of SVG specs and use of definitions and attributes in accordance with the XML and SVG definitions. All SVG content shall be enclosed between the <svg> tags. It can be taught through examples.

### 2.1. Programming in SVG

The essential elements (tags) for graphical apps are: <circle>, <ellipse>, <rect>, <line>, <polyline>, <polygon> and <path>. An all important tag is also <g> for grouping elements (shapes) and <use> employed in reusing of the elements predefined in the <defs> section. A complete list regarding definition, syntax and use of SVG elements can be found at the W3C SVG 1.1 Recommendation (www.w3.org/TR/SVG11/) web address [13] and also at the W3Schools SVG site [15]. As an example, we are herewith presenting the SVG code (ex1.svg) that performs generation of geometric shapes: polygon, circle, rectangle, line (straight) and path, while indicating for each of them the corresponding attributes and elements of identification. <text> element is also to be employed for creating text.
Drawing of geometrical shapes and text

**Ex1.svg** *(based on an example by David Lane [4])*

```xml
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg xmlns="http://www.w3.org/2000/svg"
     xmlns:xlink="http://www.w3.org/1999/xlink"
     width="570" height="470">
    <polygon style="stroke:#24a;stroke-width:1.5;fill:#eefefe"
            points="10,10,400,10,430,230,10,280,10,10" />
    <circle style="stroke:#d33;stroke-width:2;fill:#7ce"
            cx="100" cy="80" r="50" />
    <rect style="stroke:#2aa;stroke-width:7;fill:#ded;opacity:.8"
          x="170" y="80" height="120" width="220" />
    <line style="stroke:#eea;stroke-width:6"
         x1="30" y1="250" x2="340" y2="60" />
    <path style="fill:#daa;fill-rule:evenodd;stroke:none"
          d="M 230,250 C 360,30 10,255 110,140 z" />
    <circle style="stroke:#d33;stroke-width:2;fill:#7ce"
            cx="370" cy="140" r="30" />
    <text x="270" y="50">Example 1</text>
</svg>
```

If it is desired that the text generated shall include a certain typeface of a specified
size, a class shall be defined through a CSS style sheet *(class="title")* (which
shall be introduced by the `<polygon>` tag) in the `<defs>` section:

```xml
<defs>
    <style type="text/css"><![CDATA[
 .title { font-size:30px; font-weight: bold; font-family: batang;
        stroke: none; fill: black; text-anchor: middle}
 ]]></style>
</defs>
```

In this case, the `<text>` element shall be:

```xml
<text class="title" x="270" y="50">Example 1</text>
```
Changing coordinates (changing to user defined coordinates)

In the preceding example, coordinates are expressed in pixels and their value is relative to the origin (0,0) which in SVG (as well as in other languages) is implicitly the upper left corner, while positive X coordinates are to the right as related to origin and the Y coordinates are downwards [9, 13]. Certain calculations must be performed within the applications, so that all coordinates that are used should be relative to this origin. In order to avoid a large amount of mathematical calculations, homogenous coordinates \((x,y,w)\), are used, whereas \((x,y)\) are Cartesian coordinates, and \(w\) is the multiplication factor. 2D transformations are widely used in computer graphics \((\text{scaling, translation, rotation})\), by representing them as a multitude of 3 – dimensional linear transformations.

Such representation is defined by a simple 3 x 3 matrix:

\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix}
\]

This way, each transformation has its own transformation matrix which is used in calculations. The result is that all transformation is represented by a matrix calculation: the \((x, y, w)\) line vector is matriceal multiplied with the transformation matrix.

As an example, should we have to move the default origin (0, 0) to the point described by coordinates (250, 250) and change the Y positive axis to the upwards direction (as it is generally represented in mathematics), the following are needed: \(a = 1, b = c = 0, d = -1, e = f = 250\), thus the transformation matrix is \(\text{matrix}(1, 0, 0, -1, 250, 250)\). For further details: "Coordinate System Transformations" (Sec 7.4) - http://www.w3.org/TR/SVG/coords.html [4,13]. In SVG this is achieved through the use of the <g> element and the „transform” attribute:

```
<g transform="matrix(1, 0, 0, -1, 250, 250)">
... svg elements centered in (250,250)and the positive Y upwards ...
</g>
```

Drag and Drop using SVG

SVG uses Document Object Model (http://www.w3.org/TR/SVG/svgedit.html) through which it includes OOP JavaScript techniques with the purpose of generating objects, properties and methods. More details regarding the description of these properties and methods can be found in chapter 5 of Document Structure (http://www.w3.org/TR/SVG/struct.html). JavaScript code is introduced using <script> tag, subsequent to the opening of the <svg> tag, but preceding the <defs> section:
3. Graphic applications developed using SVG

Dynamic and interactive aspects and facilities offered by SVG offer a distinct possibility of developing graphic dynamic and interactive applications for various items of study. These applications can be used by the educators and trainers (teachers) within a class of pupils/students in order to explain various phenomena, concepts, terms through direct involvement of the user in understanding theoretical and practical aspects of the subject/theme approached. This approach creates an environment which is similar to an experiment which actually constitutes the very basis of the learning process. Learning sciences and psychology studies demonstrate the fact that the pupil/student is actively involved in the activities that request performing experiments, analysis and interpretations of phenomena, concepts, terms. Within the learning process, this is deemed as an important step to learning through discovery.

Object Oriented Programming (OOP), including events generated through the use and programming of mouse movements, animation programming using SMIL (Synchronized Multimedia Integration Language), DOM (Document Object Model), use of Java technology and „Drag and Drop” technique are some of the essential parameters that define the advantages of the SVG platform. We recommend the study of the source code of applications developed by David Lane [4]: Thales.svg (triangle inscribed in a half-circle; modification of the triangle apex position on the circumference using the mouse); UnitCerc.svg (the unit circle represented in either ortho or polar coordinates; indicating a point on the circumference using the mouse, calculating the radius angle in degrees, rads and also values of the cos and sin functions); ParamPlot.svg (drawing a curve using parametric equations – generating trajectory is indicated as well as the manner of drawing the curve). Below we present various applications by the parametric equations of curves.
**Pascal's snail** (it is proposed that the experiment / demo):

```javascript
function xFunc(t)
    return (a+b)*Math.cos((b/a)*t) - b*Math.cos(t+(t*b)/a);

function yFunc(t)
    return (a+b)*Math.sin((b/a)*t) - b*Math.sin(t+(t*b)/a);
```

```
function makePlot(){
    // function of drawing a curve by meshing [4]
    var dstring="M"+xFunc(0)+","+yFunc(0)+" ";
    for(t=0;t<nplotpoints;t++){
        dstring=dstring+"L"+xFunc(period*(t+1)/nplotpoints)+","+yFunc(period*(t+1)/nplotpoints)+" ";
    }
    fPlot.setAttributeNS(null, "d", dstring);
```
NOTE. Drawing a curve (Whether $C$ is a curve that borders the field $D$. The curve $C$ is modeled using the polygon line $P = P_1 \ldots P_n$, $P_i(x_i, y_i)$, $i = 1, n$), $C = FrD = Im \gamma$, $\gamma$ class route $C^1$ upon portions, $\gamma : [a, b] \rightarrow R^2$, $\gamma_i(t) = (x(t), y(t))$, $a t b$.

Using the bijective application between the real segments $[0,1]$ and $[a,b]$, given by $\varphi(t) = a + t(b - a)$, the polygon line is modeled using the reunion of the $\gamma_i$ routes parametrically represented as follows:

$$\gamma_i : [a, b] \rightarrow R^2, \gamma_i(t) = (x(t), y(t)), i = 1, n,$$

whereas

$$x(t) = x_i + t (x_{i+1} - x_i), y(t) = y_i + t (y_{i+1} - y_i), i = 1, n - 1$$

noting that for the last route, $\gamma_n$ parametric equations are

$$x(t) = x_n + t (x_1 - x_n), y(t) = y_n + t (y_1 - y_n).$$

4. Conclusions

a) "SVG is a language for describing two-dimensional graphics in XML. SVG allows for three types of graphic objects: vector graphic shapes (e.g., paths consisting of straight lines and curves), images and text. Graphical objects can be grouped, styled,
transformed and composited into previously rendered objects. Text can be in any XML
namespace suitable to the application, which enhances searchability and accessibility of
the SVG graphics. The feature set includes nested transformations, clipping paths, alpha
masks, filter effects, template objects and extensibility." [7].

b) „Scalable Vector Graphics (SVG) is the open source Worldwide Web Consortium
(W3C) recommendation for two dimensional vector graphics. The combination of SVG
and JavaScript is a powerful platform for creating interactive graphics, comparable to
Flash and Java.” (David Lane) [4].

REFERENCES

&nodeId=1381.
[8] MARIN VLADA (2006), From Green’s Theorem to Computational Geometry, CNIV-2006,