

SVG – A NEW DIMENSION IN PRODUCING INTERACTIVE NETBOOKS

Vladimir BATAGELJ

University of Ljubljana, FMF, Department of Mathematics
Jadranska 19, SI-1000 Ljubljana, Slovenia
e-mail: vladimir.batagelj@uni-lj.si

Matjaž ZAVERŠNIK

University of Ljubljana, FMF, Department of Mathematics
Jadranska 19, SI-1000 Ljubljana, Slovenia
e-mail: matjaz.zaversnik@fmf.uni-lj.si

ABSTRACT

SVG - Scalable Vector Graphics is a 2D-graphics markup language based on XML. It is compatible with other web standards: HTML, XML Namespace, Xlink, Xpointer, CSS 2, DOM 1, Java, ECMA/JavaScript, Unicode, SMIL 1.0, ... It allows us to include in HTML documents pictures described by their structure – composition of curves and shapes. Since the SVG viewer is not integrated yet into web browsers we need, to view SVG pictures, to install it as a plug-in. An excellent SVG plug-in was produced by Adobe.

The SVG pictures are not static (as standard bitmaps GIF, JPEG, PNG). The SVG viewer provides options to zoom in (to see details) and out (to see global/overall view), to move the picture, to search for text ... Besides this, using built-in animation capabilities or JavaScript program support, the pictures can be made alive and interactive. We can partition a SVG picture to several parts. Changing their attributes we can control their visibility. Using JavaScript this can be done interactively allowing to the user to select the parts to be displayed. We can also dynamically add or delete the elements of the picture and change their properties.

SVG pictures can be produced by drawing tools. But special web applications and programs for visualization of obtained data/results will produce most SVG pictures.

In the paper we present the main features of SVG and discuss their potential educational usage. Some our SVG based solutions are also listed.

Keywords: authoring, interactive, dynamic, visualization, scalable vector graphics, standards, internet.

1. What is SVG ?

In 1985 Adobe presented *Postscript* that, combined with a laser printer, produced a revolution in publishing. Since the Postscript was not completely appropriate for the use on the web they, at Adobe, developed a new format PDF – *Portable Document Format*. PDF established itself as a leading format for publication of closed documents (reports, manuals, papers, tutorials...). Several other companies produced in nineties their own graphics description formats with plug-in viewers for them: QuickTime (Apple), CMX (Corel), Flash (Macromedia) ... Their main drawback is – they are not an open standard.

In 1998 two groups of companies submitted to the *World Wide Web Consortium* - W3C their proposals for web graphics format based on XML. The first group (April 1998 / Adobe, IBM, Netscape and Sun) proposed PGML - Precision Graphics Markup Language; and the second (May 1998 / HP, Macromedia, Microsoft and Visio) proposed VML - Vector Markup Language. Both groups merged into SVG - Scalable Vector Graphics development group that published already in October 1998 the requirements on SVG and in February 1999 the first draft. Several improved versions followed. The last version was published on 4th September 2001 as a W3C Recommendation.

SVG - *Scalable Vector Graphics* is a 2D-graphics markup language based on XML. It is compatible with other web standards: HTML, XML Namespace, Xlink, Xpointer, CSS 2, DOM 1, Java, ECMA/JavaScript, Unicode, SMIL 1.0 ... It allows us to include in HTML documents pictures described by their structure – composition of curves, shapes, text and also bitmaps.

To view a SVG picture we need a special viewer. In the latest versions of the most popular web browsers the viewer is already integrated. If we use an older browser, we need to install SVG viewer as a plug-in. An excellent SVG plug-in for Windows and Macintosh was produced by Adobe.

The SVG pictures are not static (as standard bitmaps GIF, TIFF, JPEG, PNG). The SVG viewer provides options to zoom in (to see details) and out (to see global view), to move the picture, to search for text, ... Besides this, using built-in animation capabilities or JavaScript/Java program support, the pictures can be made alive and interactive.

SVG pictures can be produced using drawing tools. On Windows we can use Adobe Illustrator 10, Corel Draw 10, WebDraw (by Jasc) and Mayura. But special programs for visualization of obtained data/results will produce most SVG pictures.

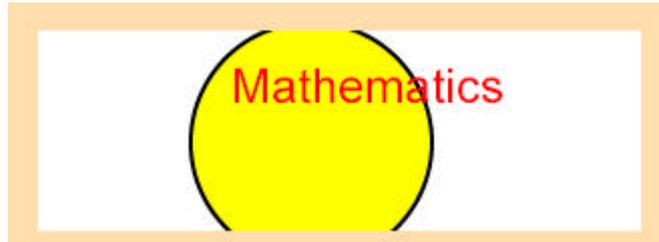
The main applications of SVG are data visualization, presentations (like Power Point), maps (GIS), technical layouts and educational pictures (illustrations).

2. SVG and HTML

Here is a simple example of picture description in SVG.

```
<svg>
  <circle cx="120" cy="65" r="30" style="fill:yellow;stroke:black;"/>
  <text x="100" y="55" style="fill:red;">Mathematics</text>
</svg>
```

It creates yellow circle with black border containing red inscription "Mathematics".



To insert a SVG picture into a HTML document we use the **EMBED** (or **OBJECT**) tag. For the picture from our simple example we have:

```
<EMBED SRC="simple.svg" NAME="simple"  
  WIDTH="300" HEIGHT="100" TYPE="image/svg+xml"  
  PLUGINSOURCE="http://www.adobe.com/svg/viewer/install/"  
>
```

The attribute **SRC** determines the location (URL) of the SVG file; **NAME** becomes important in advanced applications using JavaScript or Java. The attributes **WIDTH** and **HEIGHT** are obligatory and determine the size of rectangle in which the picture is rendered. The value of **TYPE** is the MIME-type of the file - for SVG file it can be **image/svg+xml** or **image/svg+xml**. The attribute **PLUGINSOURCE** directs the user that has not a SVG viewer installed on his computer, to the web site from which he can obtain a viewer.

3. Advantages of SVG format

In this section we present an overview of possibilities offered by SVG found in different applications on the internet. Since one of the strongest features of SVG is interactivity and dynamics, in the paper medium, because of its limitations, we can give only some snapshots. The reader is invited to visit the original pages on the web. The web version of this paper with some additional links is available at <http://www.educa.fmf.uni-lj.si/izodel/dela/SVG/>.

Zooming and moving: in SVG viewer we can **zoom**-in and -out the picture thus obtaining a detailed view of the selected part or an overall view of the picture. We can also **move** the viewing window. In Figure 1 some snapshots from *OECD Europa Atlas* (the page is in German, <http://www.carto.net/papers/svg/eu/oecd.html>) are presented. We first see the overall view of Europe. Then we zoom-in to Aegean region and further into Athens and Crete part. Note that, since in each case the picture is produced from its description, the quality of the picture depends only on the resolution of the output device (screen, printer).

Searching: Since in SVG textual data (for example, labels) are stored as text we can search for selected string. On the left bottom part of Figure 1 the result of searching for string "Grie" is displayed.



The viewer locates the label containing given string, moves the viewing window there and marks the string in the label.

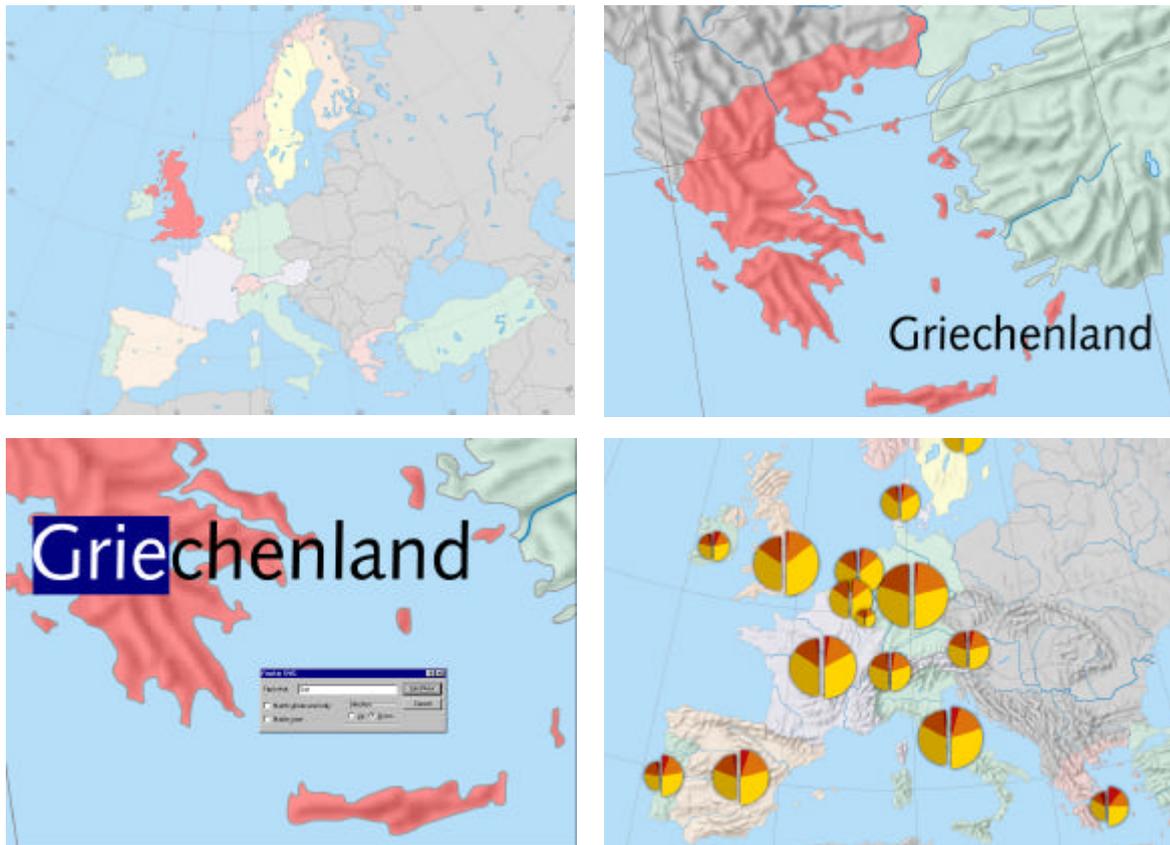


Figure 1: Zooming, searching and control of visibility

Controlling visibility: Using the `g` SVG tag we can partition a SVG picture into several parts. In the `g` tag two attributes `display` and `visibility` are available to control the visibility of the corresponding part. The difference between them is explained in SVG documentation. For example, setting the `display` attribute to `none` we switch the visibility of the part off; setting it to `inline` (or some other value) we switch the visibility of the part on. Using JavaScript this can be done interactively allowing to the user to select the parts to be displayed.

In Figure 1 in the overall view, the display of the relief and rivers is switched off.

In Figure 2 the sequence of states in the construction of the perpendicular from a given point to a given line is presented. To produce a dynamic visualization of the construction in SVG we first draw the final picture with some SVG editor. Then we put each picture increment elements into a separate group with its own visibility control, which is triggered by the user from the picture control HTML page using JavaScript.

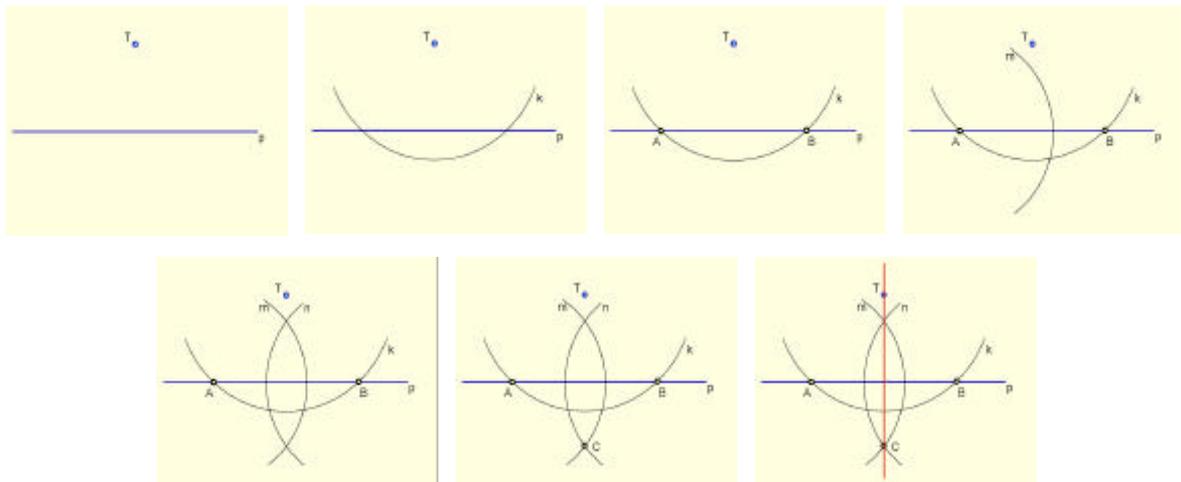


Figure 2: Construction of the perpendicular line

The same technique can be used to display user selected pictures in the same place on the page <http://vlado.fmf.uni-lj.si/pub/CONF/DSI.01/kuk.htm> , or to connect the picture elements with their descriptions/explanations http://www.usbyte.com/index_SVG.htm (select LU (load/unload) type > Interactive Drawing).

Dynamic changes of display elements: Using JavaScript or Java we can dynamically add or delete elements of the picture and change their attributes (size, location, style). These changes can be triggered by the user from the control HTML page.

On the overall view of Europe on Figure 1 we selected UK – the selected country is colored with a darker color. In the right bottom picture on Figure 1 we selected the display of pie charts representing selected economic data. The picture was augmented with pie charts.

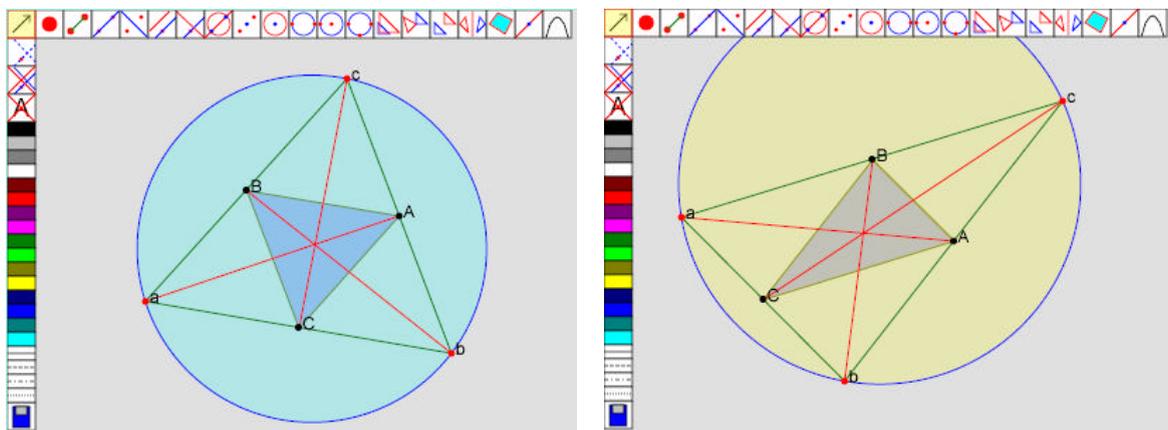


Figure 3: Geometric construction with SVGeom

The most interesting and rich site on the web with educational SVG contents is the *Pilat Informatique Educative* <http://perso.wanadoo.fr/pilat/> that contains several nice mathematical examples and systems SVGeom and SVGFonc.

The system SVGeom is essentially a SVG/JavaScript based dynamic geometry system (similar to Geometry, Cabri or Geometric Sketchpad). In Figure 3 a geometric picture produced interactively in SVGeom is presented on the left. As seen on the right this picture can be interactively transformed preserving the structural relations among elements.

Animation: SVG provides, based on SMIL, basic animation capabilities. To get an impression visit the following examples: *Kaleidoscope*: <http://www.burningpixel.com/svg/Kaleid.htm> , *Lissajou curves*: <http://www.mecxpert.de/svg/lissajou.html> , *PC Technology in motion*: http://www.usbyte.com/index_SVG.htm (select CD optical pick-up system in action).

The SVG format has several additional features: the SVG files are relatively **small**, in addition - **compressed** files are also supported; the picture can be built from ready-to-use components from libraries of **reusable** objects; **extensibility** by combining SVG with problem-specific XML solutions; **internationalization** – SVG supports Unicode and provides elaborate typography options (for example, the direction of writing); the picture description is **independent** of output devices and computer platforms, it can contain **metadata**; SVG provides excellent **color control**.

4. Our support for SVG based visualizations

To ease the preparation of visibility controlled SVG visualizations we prepared the *SVGplayer* - a collection of JavaScript functions for controlling the value of **display** attribute in parts of the SVG picture. At <http://sio.edus.si/list/1/svg/svg04.htm> you will find the page with SVG visualization of the construction of the perpendicular using the *SVGplayer*. A ZIP with the last version of the *SVGplayer* is available at <http://vlado.fmf.uni-lj.si/pub/SVG/SVGplayer/>.

Based on the approach used in *Logo2PS* we prepared also *Logo2SVG* that allows user to save a trace of the Logo turtle as a (visibility controlled) SVG picture. For details see <http://vlado.fmf.uni-lj.si/educa/logo/logo2svg/>.

We prepared also a SVG based *function drawing* page and a SVG/JavaScript version of *EulerGT* (<http://www.educa.fmf.uni-lj.si/izodel/dela/Euler/>).

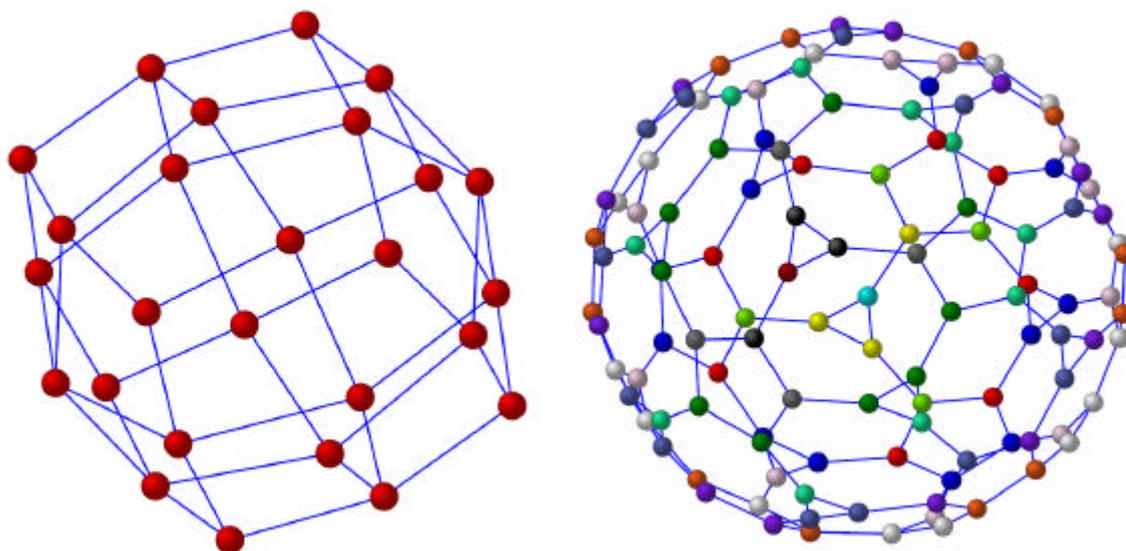


Figure 4: Graph visualizations from Pajek

For an example of the program generating SVG visualizations you can look at our program *Pajek* (<http://vlado.fmf.uni-lj.si/pub/networks/pajek/>). It is aimed for analysis and visualization of large networks. One among several graphics output formats supported by *Pajek* is also SVG. On Figure 4 you can see two SVG pictures of 3D embeddings of graphs produced by *Pajek*.

5. Conclusions

We hope that the presented examples convinced the reader about the broad new space opened by SVG for the development of educational materials. Since visualization is an important component also in mathematical education we expect that SVG will get broader acceptance also in this field in the near future.

REFERENCES

- V. Batagelj: *Logo to Postscript*, Proceedings of Eurologo'97, Budapest, Hungary, 1997, p. 333-341.
<http://vlado.fmf.uni-lj.si/educa/logo/logo2ps/>
- V. Batagelj: *Logo to SVG*, Proceedings of Eurologo'01, Linz, Austria, 2001, p. 229-236.
- V. Batagelj, M. Zaveršnik: *Mathematical notebooks*, International conference on the teaching of mathematics, Samos, Greece, 1998, p. 44-46
- P. A. Mansfield: *Graphical Stylesheets Using XSLT to Generate SVG*:
<http://www.schemasoft.com/gcathools/gca2html/Output/05-05-02.html>
- The World Wide Web Consortium - W³C: <http://www.w3.org>
- XML: <http://www.w3.org/XML/>
- W³C/SVG: <http://www.w3.org/TR/SVG/index.html>
- HTML: <http://www.w3.org/MarkUp/>
- XML Namespace: <http://www.w3.org/TR/REC-xml-names/>
- Xlink: <http://www.w3.org/TR/WD-xlink>
- Xpointer: <http://www.w3.org/TR/WD-xptr>
- CSS 2: <http://www.w3.org/TR/REC-CSS2/>
- DOM 1: <http://www.w3.org/TR/REC-DOM-Level-1/>
- SMIL 1.0: <http://www.w3.org/TR/REC-smil/>
- ECMA/JavaScript: <http://www.ecma.ch/ecma1/stand/ecma-290.htm>
- Java: <http://java.sun.com/>
- Unicode: <http://www.unicode.org/>
- Adobe SVG Viewer: <http://www.adobe.com/svg/viewer/install/>
- Adobe Illustrator: <http://www.adobe.com/products/illustrator/main.html>
- Corel Draw: <http://www.corel.com/svg>
- Jasc Software, WebDraw: <http://www.jasc.com/webdraw>
- Mayura: <http://www.mayura.com/>
- MathML to SVG Converter: <http://www.schemasoft.com/MathML/>
- TESS: <http://www.peda.com/tess/Welcome.html>
- SVG Elves, *SVG developer community*: <http://www.svgelves.com/>
- Vienna, *Social patterns and structures*: <http://www.karto.ethz.ch/~an/cartography/vienna/>
- CR2V, *raster to vector converter*: <http://www.celinea.com/>