Dušan Tošić,
Vladimir Filipović,
(Matematički fakultet, Beograd)
Jozef Kratica
(Matematički institut SANU, Beograd)

USING SVG–XML FOR REPRESENTATION OF HISTORICAL GRAPHICAL DATA

Abstract: Modern data representation requires XML-based approach. One of the ways to represent any kind of graphical data in electronic form is to use Scalable Vector Graphics (SVG). So, XML and SVG are ideal means for the digital representation of national heritage. Moreover, for the powerful using of SVG one should learn a very complex syntax and related XML applications. In this paper the advantages and drawbacks of SVG, in processing of national heritage, are specified. Some examples about processing frescos and manuscripts are presented.

1. Introduction

SVG is language for describing two-dimensional graphics in XML. In fact, it is a new technology aimed to support a workable cross-platform solution to Web imaging. A quite good description of SVG is stated in [1]: "Scalable Vector Graphics (SVG) is a new graphics file format and Web development language based on XML. SVG enables Web developers and designers to create dynamically generated, high-quality graphics from real time data with precise structural and visual control". SVG allows describing three types of graphic objects: vector graphic shapes (e.g., paths consisting of straight lines and curves), images and text. SVG-format is not proprietary. It is open standard built upon other open standards from the W3C and ECMA organizations. XML, CSS (Cascading Style Sheets), ECMAScript and SMIL (Synchronized Multimedia Integration Language) all contribute to the standard. SVG documents are described in XML so they can be easily generated and read by programs. SVG arose as a collaborative effort of some of the biggest experts in the computer world. As the result of their work the different recommendation documents appeared (see: [6], [7]). Nowadays we can find a lot of books related to SVG (see: [3], [4], [10]), but much more different articles can be found on the Web (see: [2], [5], [9]). There are many applications that work with SVG data. According to these facts, we can conclude that SVG could be very useful for the digitalization of graphical, historical documents.

2. Some advantages and drawbacks of SVG

SVG is a XML-application. The grow of XML-technology predicts that it will be used much more in future. SVG is convenient for creation a new generation of Web applications based on:

- data-driven,
- interactive, and
- personalized graphics.

By this short characterizing we can see some good features of SVG. Moreover, there are much more reasons to accept SVG in processing of graphical data.

- SVG is based on XML – a general accepted language for description of data (emerging through the cooperative efforts of the World Wide Web Consortium (W3C) and its members) (see [8]).
- With SVG one can use existing skills (JavaScript language, Document Object Model (DOM), JSP, PHP, and ASP).
- SVG format is not proprietary. It is open standard built upon other open standards from the W3C and ECMA standards organizations.
- SVG is compatible text based and works seamlessly with current Web technologies like HTML, GIF, JPEG, PNG, SMIL, …
- Graphics created in SVG can be scaled without loss of quality across various platforms and devices.
- SVG is compatible with the languages used in wireless devices.
- …

As technical advantages of SVG (more details can be found in [5]), we can mention:

- Small file sizes
- Zoomable graphics
- Choice of large number of colors and fonts
- Scripting control for the interactive events and animation
- Bitmap-style filter effects for high-impact graphics
- …

Of course there are some drawbacks about using SVG (see [2]). We are mentioning some of them:

- contemporary browsers do not fully support SVG
- complex syntax and grammar of SVG
- lack of online SVG material in developing Web sites
- clumsy documentations
- close connection to the other XML applications
- …

The lack of fully SVG support (from currently browsers) requires the use of a plug-in (see [1]). To plug in a SVG viewer one should download and install it. If all possibilities of SVG are not supported by the installed viewer, additional problems may appear. SVG is very rich language. It contains many built-in elements and large number of predefined constants. Because of that the syntax of SVG is not simple and related documentation is complex.

3. Structure of a SVG document

The structure of a SVG document is according to XML syntax. The following elements have to be included into SVG file:

1. <?xml version="1.0" encoding="UTF-8"?>
2. <!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN"
   "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
3. <svg>
4. <!-- SVG content goes here -->
5. </svg>

XML declaration <?xml version="1.0" encoding="UTF-8"> must be included in the beginning of SVG file (this declaration could be a little different depending on encoding). XML declaration on the beginning tells that the following content is a XML application. The second line is not obligated, but it gives version information and the URL where the Document Type Definition can be found. This line is used to identify SVG as a standard set of rules. The <svg> tag indicates to the browser that this is a SVG document. The description of all SVG elements must be placed between the <svg> </svg> tags. Since SVG is a XML application, all tags must be closed. So </svg> is used to close the document.

In the next section we will used some of SVG elements, but without detailed description of these elements. The description could be found in [6] and [7] or in the books: [3] and [10].

4. How to process frescos and manuscripts by using SVG

In this part we put on some ideas about the processing of frescos and manuscripts by using the elements of SVG. In fact, we present some examples where the possibilities of SVG are used in showing and modifying of digital images.

The frescos and manuscripts can be represented as the raster graphics, e.g. images based on pixel data (format: PNG, JPEG, GIF,…). SVG is vector-oriented and has limited possibility in dealing out with the raster images. However for some kinds of tasks related to digitization of national heritage, the raster images are very important.

We can use <image> tag to present an image. In the following SVG file (named BeliAndjeo.svg) the image file BeliAndjeo.jpg is presented.

```
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg">
  <title>Beli Andjeo </title>
    <image x="10" y="10" width="400px" height="300px"
      xlink:href="BeliAndjeo.jpg">
    </image>
</svg>
```
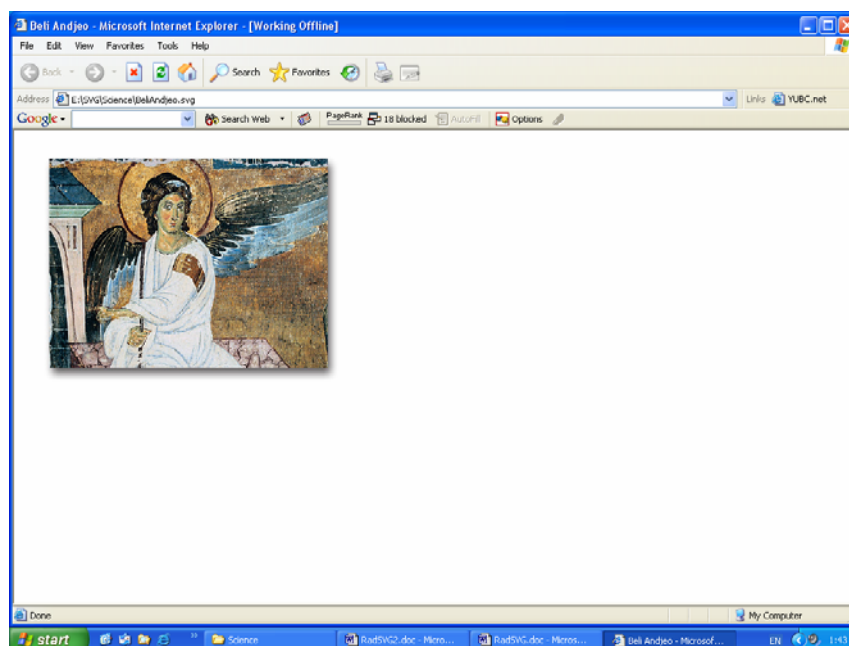

Fig. 1

If SVG viewer is plug-in and the file BeliAndjeo.jpg is in the same folder as BeliAndjeo.svg, Internet Explorer (IE) shows previous file as a picture above.

A SVG file may be embedded into a HTML (XHTML) document by using <object> or <embed> tag. In the following example, <object> tag is used.

```
<html>
<head><title> White Angel</title></head>
<body>
<h2> WHITE ANGEL </h2>
<object data="BeliAndjeo.svg" type="image/svg+xml" width="400"
            height="300"></object>
<p>WHITE ANGEL - the fresco from Mileševa monastery</p>
<p>Mileševa monastery is foundation of king Vladislav (1234-1243).</br>
    Painting is from third decade of 13th century, before 1228.</p>
</body>
</html>
```
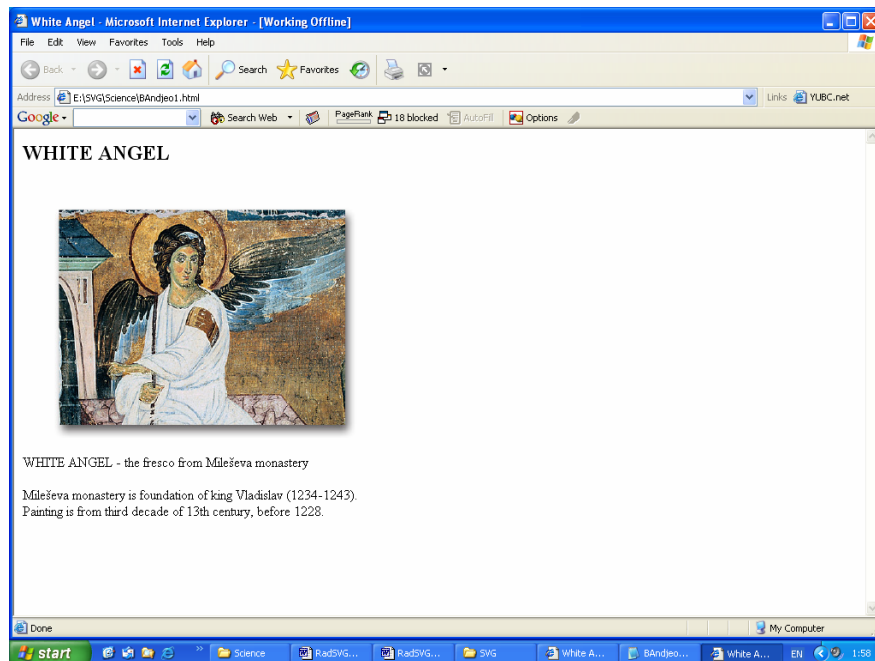
IE shows this file in the following manner:

Fig. 2

The images can be presented on different positions and in different sizes. The following SVG file contains the images from the manuscript "Miroslav's Gospel":

```
<?xml version="1.0" encoding="UTF-8"?>
<svg xmlns="http://www.w3.org/2000/svg">
  <title> Miroslavovo jevandjelje </title>
    <image x="0" y="0" width="300px" height="400px"
       xlink:href="ME0.jpg">  </image>
    <image x="400" y="0" width="300px" height="400px"
       xlink:href="ME1.gif">  </image>
    <image x="0" y="410" width="100px" height="100px"
       xlink:href="ME0.jpg">  </image>
```

```
    <image x="150" y="410" width="100px" height="200px"
      xlink:href="ME1.gif">  </image>
    </image>
</svg>
```

If the files ME0.jpg and ME1.gif are in the same folder as previous file, IE will give:
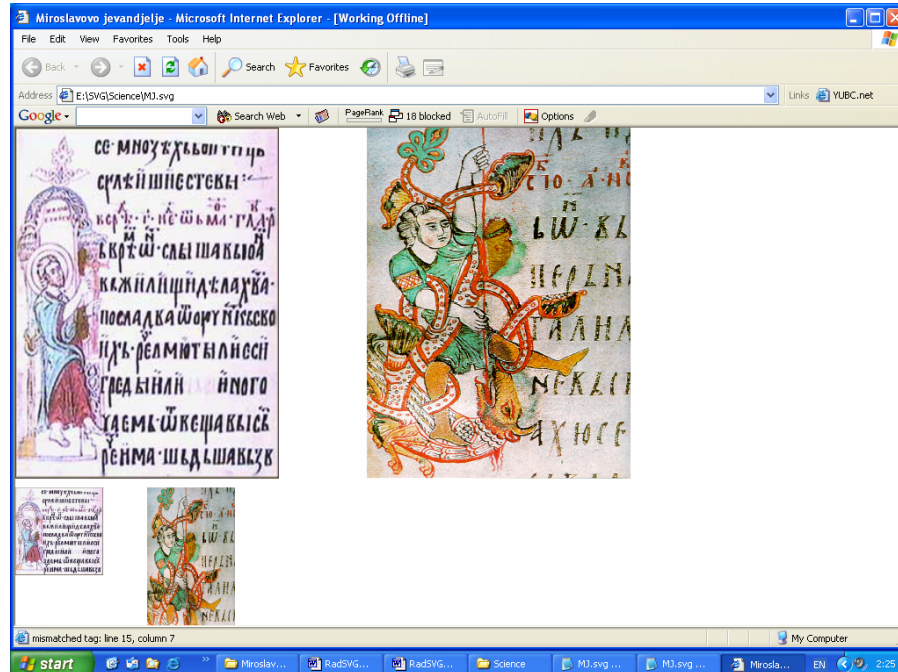

Fig. 3

The original pictures could be changed by adding different figures and shapes. In the following example the red rectangle and blue circle are filled over the origin picture.

```
    <?xml version="1.0" encoding="UTF-8"?>
    <svg xmlns="http://www.w3.org/2000/svg">
      <title>Modifikovano Miroslavljevo jevandjelje</title>
        <image x="0" y="0" width="400px" height="500px"
          xlink:href="ME0.jpg"></image>
        <rect x="0" y="0" width="200" height="150" fill="red" />
        <circle cx="300" cy="300" r="50" fill="blue" />
    </svg>
```

In this way we damage the original picture, because we get something like Fig. 4. Moreover, the tags related to drawing and filling different shapes are very useful for the processing of original picture. We can delete, cover, color etc. the parts of an original picture. Using these techniques we can repair damages in the original picture or adapt the picture for some needs.

The text can be added to picture, too. SVG provides a lot of possibilities in the presentation of text. In the next example, the rotated text is written over the manuscript:

```
    <?xml version="1.0" encoding="UTF-8"?>
    <svg xmlns="http://www.w3.org/2000/svg">
      <title>Modifikovano Miroslavljevo jevandjelje</title>
        <image x="0" y="0" width="400px" height="500px"
          xlink:href="ME0.jpg"></image>
        <g transform="rotate(20)">
          <text x="100" y="100" font-size="36"
```

```
        font-family="sans-serif" fill="red" >
        Text overe the manuscript!
      </text>
    </g>
</svg>
```
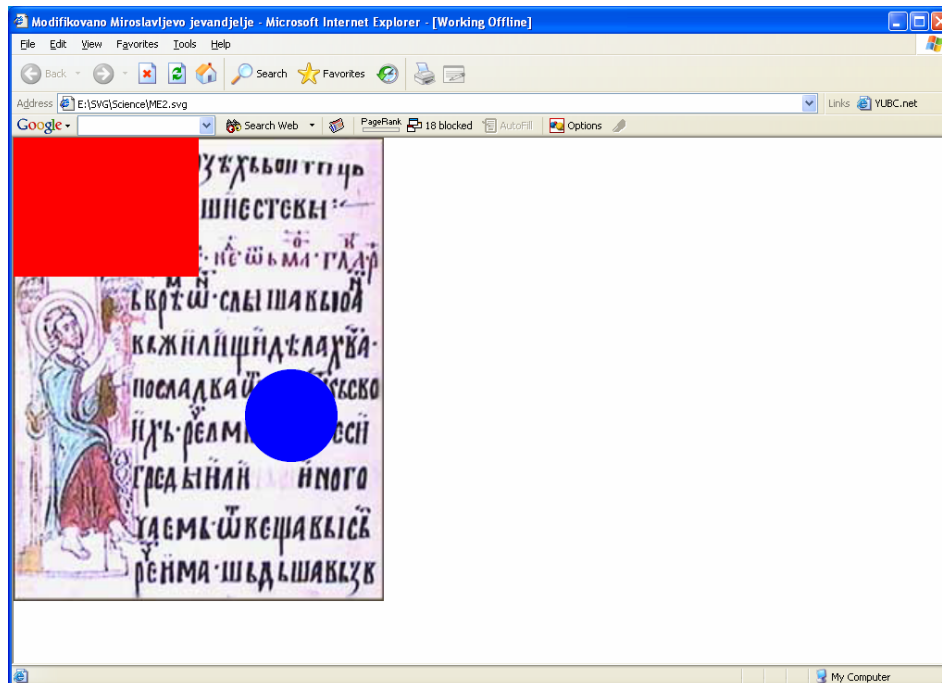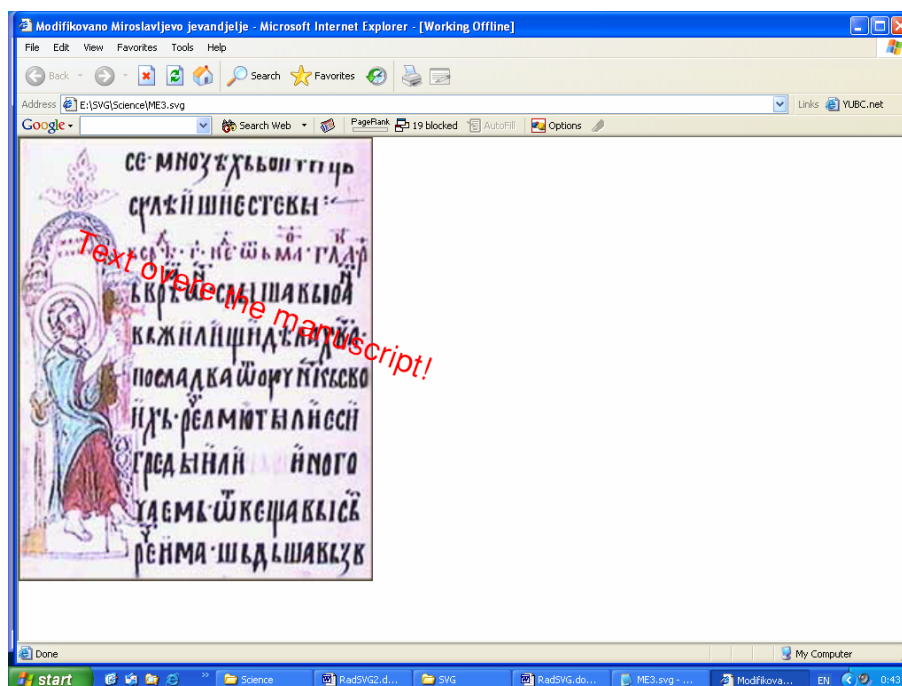

Fig. 4

The result of this processing is:

Fig. 5

These examples show only some of SVG possibilities in processing national heritage represented by images. Of course, we can do much more, but through these examples we demonstrate some simple opportunities of SVG.

5. Conclusion

XML technology is a good choice for the digitization of national heritage. The presence of large number of images in our cultural heritage needs adequate processing of 2Dgraphics. So, the natural choice is the XML graphical application, i.e. SVG. By using XML markup, we can create and implement graphic images, animations, and interactive opportunities. We can expect that SVG will become a standard language (technology) for the processing of graphical elements in our national heritage.

References and links

[1] Adobe on SVG http://www.adobe.com/svg
[2] Duce D: Scalable Vector Graphics (SVG): Vector Graphics for the Web, http://www.ariadne.ac.uk/issue28/graphics, 2001.
[3] Eisenberg D. J: SVG Essentials, Oreilly, 2002.
[4] Holzner S: Sams Teach Yourself XML in 21 Days, Third Edition, Sams Publishing, 2003.
[5] Holzschlag M.E: Scalable Vector Graphics, http://www.webtechniques.com/archives/2001/04/desi/, 2001.
[6] Scalable Vector Graphics (SVG), Specification 1.1, http://www.w3.org/TR/SVG11/
[7] Scalable Vector Graphics (SVG) 1.2, http://www.w3.org/TR/SVG12/
[8] Tošić D: XML-tehnologije i digitalizacija, Pregled nacionalnog centra za digitalizaciju, 3, pp. 1-12, 2003.
[9] Traversa E. Scalable Vector Graphics: The Art is in the Code, http://wwwwebrefrence.com/authoring/languages/svg , 2001.
[10]Watt A., Chris Lilley C. et. al: SVG Unleashed, Sams Publishing, 2002.