Graphics-Based Text And Its Applications on Web Presentation Software

Amil Ahmad Ilham[1]

Abstract

This project has produced a system that can be used to create and display slide-shows, like the sort produced by the other presentation software, using *XML* (Extensible Markup Language) as the storage language for the slide-show and *SVG* (Scalable Vector Graphics) as the display language.

XML is a meta-data – a language to create other language – developed by W3C (World Wide Web Consortium) for use across the Word Wide Web. It is a simple, very flexible text format derived from SGML (ISO 8879). Originally designed to meet the challenges of large-scale electronic publishing, XML is also playing an increasingly important role in the exchange of a wide variety of data on the *Web* and elsewhere.

SVG is a language for describing *two-dimensional graphics* and *graphical applications* in XML. The visual appearance of SVG *graphics* is stored as *a text-based description* and its display is recalculated when it is rendered. SVG is used for storage and distribution of images on the Web, and is increasingly well-supported by both commercial and free software. In contrast with raster image formats such as GIF, JPEG, and PNG, which store a matrix of individual pixels that compose an image, an SVG image is a *graphic-based text*. It contains instructions for resolution independent rendering: the same SVG file will be shown in more detail when viewed at a higher resolution.

Introduction

Presentation software is a software product for creating slides and audio-visual aids for public speaking and presentations. Compressed to its essence, a presentation consists of three most basic elements: presenter, audience and presentation tool. Presentation software, which is part of the presentation tool, has an important role in creating effective presentation, especially to produce exciting slides that can grab the attention of audience.

The advances in computer science and software engineering have produced a variety of presentation software that can be used to create and deliver interesting slide presentations. Some of them are Apple Keynote, AppleWorks, Corel Presentations, Curtain Call, Formula Graphics, Harvard Graphics, Kpresenter, Lotus Freelance, MacroMedia (Director, Authorware, Flash), Microsoft PowerPoint, ShowMaker, StarOffice/ OpenOffice Impress, StarSoft Slide-show, WF-Slide-show and Walker.

Of course, these tools have their own advantages and limitations. They are also different in term of the operation systems supported, technologies for creating slide-shows, features for generating web-viewable slide presentations, license, etc.

[1] Department of Electrical Engineering, Faculty of Engineering, Hasanuddin University
Jl. Perintis Kemerdekaan Km. 10, Tamalanrea 90245, Indonesia
E-mail: amil@unhas.ac.id Phone/fax: (0411) 581 642/588 111

Technologies For Web-Based Presentations

There are some representative technologies currently used for Web-based presentations such as HTML (Hyper-Text Markup Language) Text, HTML Images, XML (Extensible Markup Language), SVG (Scalable Vector Graphics), XSLT (Extensible Stylesheet Language Transformations) and CSS (Cascading Style Sheets).

HTML Text

Presentations are displayed as HTML Text. The text in HTML document acts, for the most part, likes the text in a word processor. The advantages of HTML text are that it is easily edited if changes are required and the text can adjust to the width of the screen it is being viewed on. However HTML text also has some severe limitations for design purposes. By and large, most of the textual control is left up to the visitor's browser and things like run text vertically rather than horizontally cannot be done without implementing Cascading Style Sheets (CSS). CSS gives designers greater control over many of the common typographic abilities (such as line and word spacing), but even with CSS, HTML text is severely limited particularly in the special effects department.

HTML Images

Unlike HTML text, HTML images use graphics to present Web contents. Most images on the Web are bitmaps, such as JPEGs and GIFs which treat each graphic as a collection of dots, assigning a specific color to each pixel. When viewed as a whole, the collection of pixels forms an image. Using graphics gives control of the presentation back to the designers of pages and make pages more attractive. Web designers can use images to create graphical elements for presentation, such as fancy borders around the page. They can make images using fancy fonts for rendering text which replace plain text on the page.

However, there are some limitations that go along with using a graphic such larger file sizes (larger graphics mean slower download) and the fact that graphic text is usually difficult to edit. Graphics also take up a set amount of screen space and may be cut off if the screen is not large enough.

XML

An XML document is very similar to an HTML document. It consists of elements with established rules concerning their notation. An element consists of content, a start and end marker. These start and end markers are called tags. A start tag consists of an opening angle bracket, a name, some optional attributes and a closing angle bracket, for example, <slide id=”slideCover”>. An XML document must be well-formed and valid. A well-formed XML document is a document that conforms to the XML syntax rules and it is valid if the well-formed XML document conforms to the rules of a Document Type Definition (DTD).

SVG

SVG is platform independent and open-standard graphics language. It allows for three types of graphic objects: vector graphic shapes (e.g., paths consisting of straight lines and curves), images and text. Vector graphics use mathematical functions to describe the lines, shapes, and solid areas that make up a graphic.

SVG graphics provide numerous immediate benefits beside resolution independence. SVG files are often smaller than an analogous raster image, thus web pages using them may take less time to download. Because SVG is based on XML, SVG files are easy to exchange, process, and analyze. SVG integrates well with CSS (Cascading Style Sheets) specifications, thus enabling some separation of the content of the graphic from the visual appearance of that image. For example, the colors of a graphic can be specified in a style sheet that is independent of the SVG file itself. SVG also preserves image structure at a higher level – for example, a web browser can directly read the text included in an SVG figure. This ability, along with the separation of style from content, dramatically improves the accessibility of images for users with color-blindness or other visual impairments.

XSLT

Extensible Stylesheet Language Transformations (XSLT) is a part of XSL that is concerned with the transformation process. The XSLT elements therefore describe how the transformation must be carried out. A transformation expressed in XSLT is called a stylesheet. This is because, in the case when XSLT is transforming into the XSL formatting vocabulary, the transformation functions as a stylesheet. Figure 2.1 shows how XSLT processor generates an output XML document e.g. SVG from an XML input document against XSLT Stylesheet.

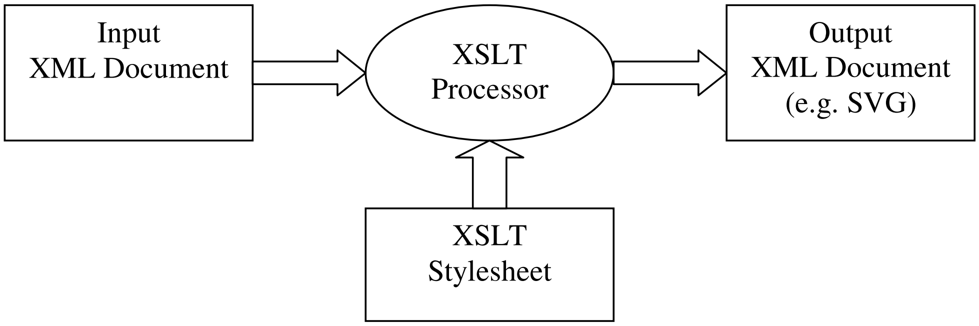

Figure 1. XSL Transformation process

CSS

Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g. fonts, colors, spacing) to Web documents. CSS can be applied to HTML and any XML-based document, such as SVG. The styling information defines the visual image of the content. The main goal is to separate the content of the document from the layout defined by CSS.

System Prototype

We have designed a system to achieve the principal requirements of presentation software.

Requirements

The system must be able to provide the following features to the user:

1. A user can easily create slide presentation for different purposes in different events.
2. A user can control as much as possible the look and the style of the slide presentation.
3. A user can create attractive slides by embedding images, using slide animation, etc.
4. A user can create slide-shows for on-line presentation and the slide-shows can be viewed by any computer (or similar device) with a modern Web browser.
5. A user can use slide-show features in the presentation such as loading slide-shows, moving forward to the next slide, moving backward to the previous slide and jumping to a certain slide.

Design

Figure 2 shows Presentation software use case diagram and figure 3 shows Presentation software class diagram.

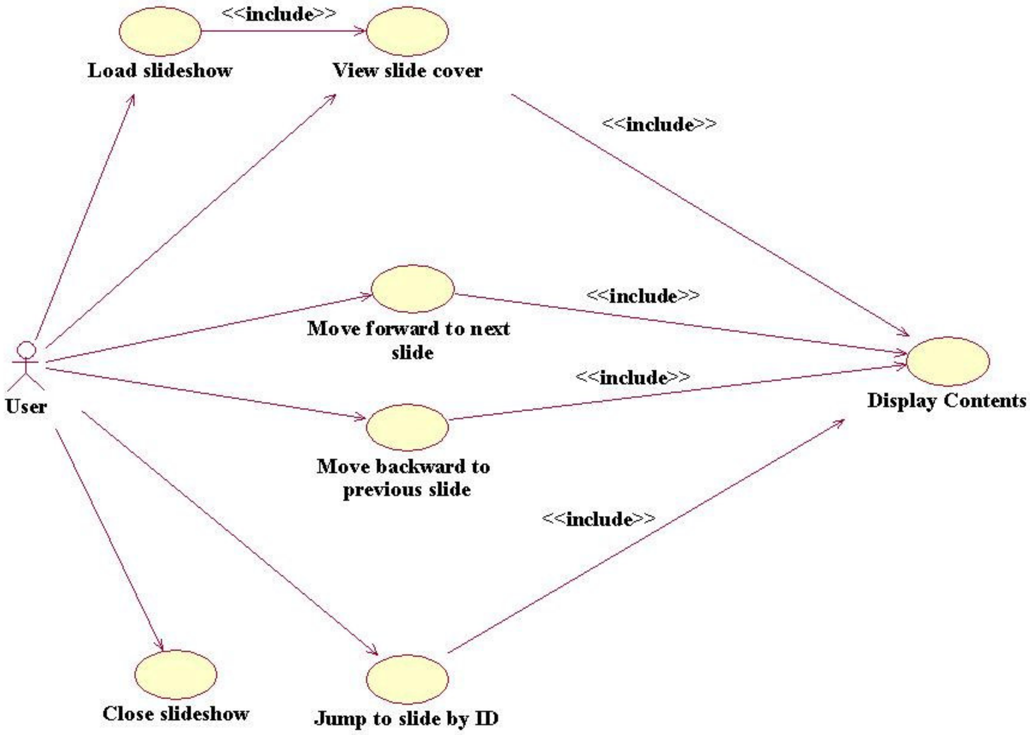

Figure 2. Presentation software use case diagram

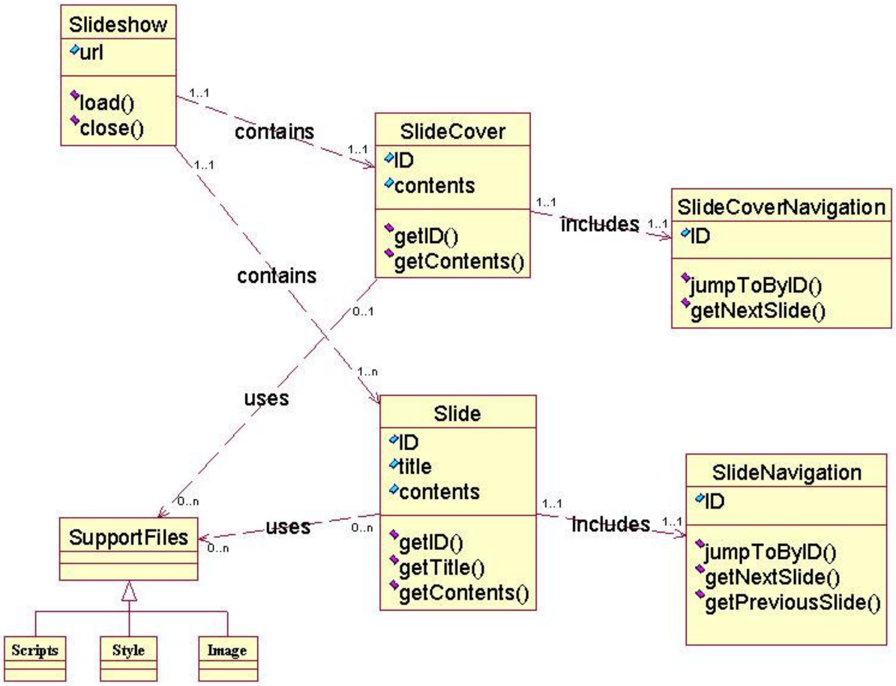

Figure 3. Presentation software class diagram

Implementation

The current prototype provides some features such as rendering text (e.g. title, paragraph, list, hyperlinks, footnotes), rendering images (e.g. image, hyperlinks, background, bullets), animation (e.g. transition effects) and navigation controls.

The overall structure of generated slide-show contains one page for slide presentation cover and at least one page for slide. The cover page will contain a slide presentation title, author's or presenter's name and slide background preference. Author's or presenter's job title and logo can be added to the cover page but they are optional. The slide will contain at least slide title and slide background preference. A slide might have no contents since this element is optional to the slide. A transition effect is optional for both slide cover and slide but a unique identification name (ID) for each slide including slide cover is compulsory.

This structure was defined in DTD as follows:

```
<!ELEMENT slideset    (cover, slide+)>
<!ELEMENT cover    (title, author, logo?, (background|nobackground), slidetransition?) >
<!ATTLIST cover    id            ID           #REQUIRED >
<!ELEMENT author  (name, jobTitle?)>
<!ELEMENT slide       (title, contents?, (background|nobackground), slidetransition?)>
<!ATTLIST slide      id            ID    #REQUIRED>
```

An example of well-formed and valid XML input to generate SVG slide-shows is

```
<?xml version='1.0' encoding="utf-8"?>
<!DOCTYPE slideset SYSTEM "myslides.dtd">
<slideset>
    <cover id="slideCover">
            <title> Graphics-Based Text And Its Applications on Web Presentation Software</title>
            <author><name>Amil A. Ilham</name> </author>
            <nobackground />
   </cover>
   <slide id="slide1">
            <title>Introduction</title>
            <nobackground />
   </slide>
</slideset>
```

An example of well-formed and valid XML input for the <title> element is

```
<cover id="slideCover">
            <title> Graphics-Based Text And Its Applications on Web Presentation Software</title>
</cover>
```

The XSLT instruction for returning the content of this title is

```
<xsl:template match="slideset/cover">
            ..........
                        <xsl:with-param name="fullText" select="./title"/>
            ..........
</xsl:template>
```

However to render text in SVG, other parameters are required. The following XSLT codes show all parameters that are required to render the title.

```
<xsl:template match="slideset/cover">
    <xsl:variable name="ss_title_fontsize" select="65" />
    <xsl:variable name="ss_title_maxLength" select="35" />
    <xsl:call-template name="renderText_1">
        <xsl:with-param name="fullText" select="./title"/>
        ...........
    </xsl:call-template>
</xsl:template>
```

The following XSLT template will transform text from XML into SVG structure.

```
<xsl:template name="renderText_1">
    <xsl:param name="fullText" />
    .............
        <xsl:choose>
            <xsl:when test="string-length($fullText) &gt; $maxLength">
                <xsl:call-template name="splitPointPosition">
                    <xsl:with-param name="str" select="$fullText"/>
                    <xsl:with-param name="maxLength"
                                           select="$maxLength"/>
                </xsl:call-template>
            </xsl:when>
            <xsl:otherwise>
                <xsl:value-of select="string-length($fullText)"/>
            </xsl:otherwise>
        </xsl:choose>
    </xsl:variable>
    <xsl:variable name="yIncrement">
        <xsl:value-of select="round(number($fontsize) + (number($fontsize) * 0.1))"/>
    </xsl:variable>
    <xsl:variable name="textHead" select="substring( $fullText, 1, $splitPoint )"/>
    <xsl:variable name="textTail" select="substring( $fullText, ($splitPoint))"/>
    <text x="{$x}" y="{$y}" class="{$class}" font-size="{$fontsize}" >
        <xsl:value-of select="$textHead" />
    </text>
    <xsl:if test="string-length($fullText) &gt; $maxLength">
        <xsl:call-template name="renderText_1">
            <xsl:with-param name="fullText" select="$textTail"/>
            ...........
        </xsl:call-template>
    </xsl:if>
</xsl:template>
```

The generated SVG codes from the example input are:

```
<text font-size="65" class="slideshowTitle" y="400" x="700"> Graphics-Based Text And Its Applications </text>
<text font-size="65" class="slideshowTitle" y="472" x="700"> on Web Presentation Software</text>
```

The following styling information is put in the class slideshowTitle in CSS document.

```
.slideshowTitle {
    text-anchor: middle;
    font-family: "Times New Roman", "Symbol", serif;
    fill: brown; }
```

Conclusion

This system prototype can be developed to allow the creation of slide-show presentations. The input to this system is a simple plain-text description of that presentation. The output is a presentation that can contain complex hypermedia elements. The system is implementing such that the resulting presentations in SVG can be displayed using freely available open-source technologies

References

[1]Knobloch, M., Kopp, M. 2003. *Web Design with XML*, Wiley.
[2]Livingstone, D. 2003. *Advanced JavaScript: insights and innovative techniques*, Prentice-Hall.
[3]Watt, A. H. 2002. *Designing SVG Web Graphics*, 1st printing, New Riders.

[4]Cascading Style Sheets Home page, W3C, http://www.w3.org/Style/CSS/ [18 November 2003]
[5]Document Object Model (DOM), W3C, http://www.w3.org/DOM/ [7 November 2003]
[6]Extensible Markup Language (XML) 1.0 (Second Edition), W3C Recommendation, http://www.w3.org/TR/REC-xml [6 October 2000]
[7]Flash Kit, a Flash Developer Resource, Jupitermedia Corp., http://www.flashkit.com/index.shtml [24 November 2003]
[8]Glossary of Terms: Definition of presentation software, Scala, Inc., General Web page, http://www.scala.com/definition/presentation-software.html [10 July 2004]
[9]GIF (Graphics Interchange Format), fCoder Group, General Web page; http://www.image-formats.com/gif.htm [24 April 2005]
[10]HyperText Markup Language (HTML) Home page, W3C, http://www.w3.org/MarkUp/ [04 November 2003]
[11]JPEG, Elysium Ltd., General Web page; http://www.jpeg.org/ [24 April 2005]
[12]PDFZone.com, InternetBiz.net, LLC., General Web page; http://www.pdfzone.com/ [24 April 2005]
[13]Scalable Vector Graphics (SVG) 1.1 Specification, W3C Recommendation; http://www.w3c.org/TR/SVG11/ [14 January 2003]
[14]Synchronized Multimedia, W3C, http://www.w3.org/AudioVideo/ [02 November 2003]
[15]The Extensible Stylesheet Language Family (XSL), W3C, http://www.w3.org/Style/XSL/ [18 November 2003]
[16]XSL Transformations (XSLT) Version 1.0, W3C Recommendation; http://www.w3.org/TR/xslt [16 November 1999]
[17]XForms 1.0, W3C Recommendation, http://www.w3.org/TR/2003/REC-xforms-20031014/ [14 October 2003]