This document is downloaded from DR-NTU, Nanyang Technological University Library, Singapore.

Title	Designing a virtual exhibition using scalable vector graphics
Author(s)	Goh, Dion Hoe-Lian; Wang, Julie Chih Ee
Citation	Goh, D., & Wang, J. (2004). Designing a virtual exhibition using Scalable Vector Graphics. Aslib Proceedings, 56(3), 144-155.
Date	2004
URL	http://hdl.handle.net/10220/6169
Rights	

Goh, D.H., and Wang, J.C.E. (2004). Designing a virtual exhibition using Scalable Vector Graphics. *Aslib Proceedings*, 56(3), 144-155.

Designing a virtual exhibition using scalable vector graphics

Dion Hoe-Lian Goh and **Julie Chih-Ee Wang**
Division of Information Studies, School of Communication and Information, Nanyang Technological University, 31 Nanyang Link, Singapore 637718, Singapore
ashlgoh@ntu.edu.sg

Keywords Scalable vector graphics, Flash, Archives, Virtual exhibitions, World Wide Web

Abstract

Scalable Vector Graphics (SVG) is a XML-based vector graphics standard developed by the World Wide Web Consortium. The National Archives of Singapore recently launched an online virtual exhibition, *Colours in the Wind*, which was developed using Macromedia Flash, the most widely used format for animated and interactive vector-based Web content. The current work reports on the experiences in developing an SVG-based version of this exhibition as well as the exploration of SVG's potential for developing future virtual exhibitions as an alternative to Flash. Observations from the SVG effort are discussed together with descriptions of techniques used and the limitations and problems that SVG poses to developers. A comparative study involving the usability features of the Flash and SVG versions of the virtual exhibition is also described. Results indicate that users are positive towards SVG and suggest the viability of the technology as a development platform.

Introduction

The National Archives of Singapore (NAS) is a Singapore government organization that is tasked with the permanent preservation of the country's records that are of historical importance and significance (Siam, 2001). Established in 1968, it is responsible for acquiring, collecting, preserving, managing, and disseminating information about the country's history, culture, arts, and heritage.

The NAS identifies, selects, and acquires records of historical interest from local and overseas individuals and organizations. Intrinsically valuable records are preserved and conserved in their original form in environmentally controlled repositories. Today, the heritage scene in Singapore has become vibrant and much effort has been expended to promote awareness of the country's history and heritage. Publications, CD-ROMs and exhibitions are some of the media adopted to disseminate information.

In particular, the NAS has been, over the years, organizing exhibitions in museums, schools and public places. These exhibitions are assembled from various archived original materials and artefacts culled from the NAS' own repository. The displays are meant to be educational and target both educational institutions and the general public. To extend its reach into the Singapore and global community, the NAS, through its collaboration with Nanyang Technological University (Singapore),

initiated its first Web-based virtual exhibition opened in December 2001. The exhibition, entitled *Colours in the Wind: Old Hill Street Police Station in Retrospect* (http://aavis.nhb.gov.sg/exhibit/help/colours.html), documents the history and preservation of the Old Hill Street Police Station and its transformation to the Ministry of Information and the Arts Building (MITA). The exhibition also serves as a testing ground for NAS' effectiveness in arousing the younger generation's interest in Singapore's history and heritage via an online, interactive medium.

The virtual exhibition was developed in Macromedia Flash, as it is the most widely accepted vector-based multimedia Web format. However, there were several major issues encountered in utilizing Flash in this exhibition. Firstly, Flash is a proprietary product of Macromedia. Developing Flash applications would thus require the NAS to purchase copies of the Flash development environment, without which editing cannot take place. Although buying several licenses of the software may not be an issue for a government agency, the NAS has to take into consideration the problem of upgrading and having to maintain multiple versions of Flash to cater for users who do not upgrade their browser's Flash plug-ins.

A second related problem is that Flash is not readily accessible for editing, unlike HTML files where the source code is accessible at the point of download. This made it difficult for the NAS to make changes to the exhibition as much of it was outsourced. Considerable effort thus had to be made to obtain and understand the source code before changes could be made. Another problem is that Flash files are in a binary format and their contents cannot be extracted by Web crawlers for indexing, thus eliminating the possibility for the exhibition to be searchable over the Web. In addition, the "Find" feature available on Web browsers to search for text does not work.

Upon completion of *Colours in the Wind*, a separate project was undertaken to evaluate alternative technologies for developing future NAS virtual exhibitions. The remainder of this paper discusses the experiences in using one such technology, Scalable Vector Graphics (SVG), a World Wide Web Consortium standard. It will first introduce SVG and then describe the tools and techniques used to develop an SVG version of the virtual exhibition. Findings, benefits and problems encountered will be reported. A comparative study involving the SVG and Flash versions of the virtual exhibition will also be described.

An overview of SVG

Basic concepts

The World Wide Web Consortium (W3C) first initiated the creation of an XML-based vector graphics format at the beginning of 1998 that eventually emerged as the Scalable Vector Graphics (SVG) specification in 1999. SVG 1.0 was released on 4th September, 2001. Now a W3C standard, SVG is "a language for describing two-dimensional vector and mixed vector/raster graphics in XML" (Bowler *et al*., 2001). Being XML-based, SVG source code is readily accessible for viewing, similar to HTML files. Watt (2001) also observed that there is a need for an XML-based graphics format to complement the large family of XML technologies. SVG can be

regarded as the visual family component of XML and is easily integrated into other XML-based applications.

SVG may be regarded as Macromedia Flash's biggest potential competitor. A key feature SVG has over Flash is that one can develop vector graphics or an entire Web page without depending on a third-party application save for a text editor. Before SVG, no one could draw such graphics without the use of at least a graphics application and a mouse. SVG graphics are resolution-independent and they can be magnified or shrunk without any loss in quality. SVG graphics can also be zoomed and panned and this is especially useful in viewing SVG-based cartographic maps. While SVG is vector-based, raster images can be embedded in SVG files as well.

SVG also has a facility called SVG fonts that provides "a common font format for SVG that is guaranteed to be supported by all conforming SVG viewers" (W3C, 2001a). This means that end users viewing a Web page that incorporates SVG will receive the necessary data to display the correct fonts even if the end user has not installed the fonts on his/her browser.

SVG is "scalable" in two aspects. The W3C Graphics Activity Statement (W3C, 2002a) states that:

> The term *Scalable Vector Graphics* has been chosen to emphasize that the graphics generated by a vector graphics format for the Web will render properly even if they are re-sized when displayed on different screens, or for the purpose of printing on paper. This benefit to Web users as a community cannot be understated.

SVG is also scalable as it was "designed to be stylable, to work well across platforms, output resolutions, color spaces, and a range of available bandwidths" (W3C, 2002a). Since SVG is stylable, as one would style text using Cascading Style Sheets, SVG graphics are also reusable. There is thus no need to create two identical graphics using different colours for example. In addition, internationalisation of Web pages is more easily achieved as SVG text is easily adapted to different languages without having to create additional graphics to cater for other languages.

Interactivity in SVG

One of the attractions of Flash is the ability to provide new levels of interactivity beyond the selection of links afforded by traditional HTML documents. Like Flash, SVG graphics can be interactive and be manipulated to perform an action under the control of time or triggered events, either by the user or by another event. Some examples of user-side interactivity include clicking of the mouse buttons, moving the cursor over areas that may have links or embedded functions, zooming and panning over a graphic.

Interactivity is achieved through the SVG DOM (W3C, 2001b), which is compatible with the DOM Level 2 Specification and maintains consistency with the HTML Document Object Model. The SVG DOM allows SVG elements and attributes to be manipulated by scripts such as ECMAScript (JavaScript). The syntax involved in accessing the SVG is similar to the DOM used in other non-SVG contexts.

For example, *-Roids (http://www.kevlindev.com/games/roids/index.htm) is a SVG game modelled on the old Asteroids computer game. Interactivity plays the main role here and the SVG-based game is embedded in an HTML file. Keyboard control is enabled, as it is more familiar to users and easier to control the game than using a mouse. The way the individual SVG elements are animated is determined by user input and controlled by ECMAScript code which accesses the SVG DOM, making references to various SVG elements.

Developing SVG graphics

Speed and efficiency are important factors in creating, editing or updating SVG. SVG is not confined to any proprietary program, and even Microsoft's Notepad will suffice for creating or editing an SVG file since it is a text-based format. An advantage to being text-based is reusability (Sommerville, 2001; Zand and Samadzadeh, 1994). Web sites have been set up for SVG developers to offer examples, share code snippets and provide debugging advice.

For novice developers, software vendors in the market provide applications especially for drawing SVG graphics or to integrate SVG functionality into existing drawing applications. Some examples include Adobe Illustrator and Corel Draw. Nevertheless, in a developer's most desperate moment, at a client's computer for example, he/she can rely on editing SVG files using any simple text editor while a Flash developer still requires a proprietary program.

Archiving SVG files

Digital documents that require long retention periods face accessibility problems because of the obsolescence of hardware and software (Heminger and Robertson, 2000). With regards to images, the report *Electronic Digital Imaging Standards for Archiving Records* by the United States Department of Defense noted that there is not one digital image format that can offer long-term preservation. Consequently, the goal is to identify formats that are likely to "live longer than others" (MacTavish and Pickard, 1999). In addition, the report also recognizes that an electronic record's long-term preservation requires that as the technology changes, the record should be "migrated from one format to another and then verified to ensure no loss of data" (MacTavish and Pickard, 1999). By observing this, the cost and frequency of translating an image standard into another can be greatly reduced.

The last issue is the most directly related to SVG's versatility. Many software vendors over the years have tried to come up with different file formats for various uses. A small percentage survived in the industry and the rest became obsolete. As they dwindle into extinction, support for them has consequently diminished too. If editability were a major consideration in archiving images, then Flash graphics would need to be saved in two formats – FLA and SWF. The FLA format is essential for future editing, while the SWF format is used for playback. On the other hand, since SVG is XML-based and hence in the form of text, archiving such files is not likely to be an issue. In addition, support for editing is also likely to be available. Playback will be an issue only if the SVG standard is no longer supported.

Searching SVG files

Searching is perhaps one of the most popular activities on the Web and is also one of the more important ways for a Web site to be discovered. Being text-based, SVG

graphics can be easily indexed by search engines. However, search engines do not yet support the indexing of SVG graphics, perhaps because the standard is still new and not yet popular on the Web.

To counter this, the W3C has made available a suite of tools on its Web site that can extract text and metadata from SVG and convert graphics into various formats. One of its goals is to demonstrate how easy it is for search engines to handle SVG files (W3C, 2002b). Despite the current lack of support, it should be noted that SVG is nevertheless readily indexable and searchable. On the other hand, Flash content cannot be searched and indexed because its file format "uses a binary rather than a text-based format which provides few 'hooks' for text searchability or image extraction" (Probets *et al*., 2001). Thus, when a collection is huge, it becomes a difficult task to search for the right Flash file unless appropriate metadata is associated with each file.

Introducing *Colours in the Wind*

This section introduces the SVG version of *Colours in the Wind*. It highlights the major features of the virtual exhibition, setting the stage for the discussion of the lessons learnt in the next section. The "front page" of the virtual exhibition consists of six sections, each represented by a window as shown in Figure 1.

Take in Figure 1 Front page of *Colours in the Wind*

The first window, "A Time and a Place: History of the Site", gives an overview of the geographical and historical development of the site on which the MITA Building is now standing. There are five black-and-white maps provided with historical facts in chronological order, and a user can zoom and pan with the aid of a "magnifying glass" (see Figure 2).

Take in Figure 2 A Time and a Place

The second window, "Here, There and Everywhere: The Surrounding Areas", gives a tour of the vicinity where the Old Hill Street Police Station used to be a central landmark. An interactive map shows the locations of both present and long demolished historical buildings, bridges and monuments. Hot spots that provide further information will animate and be identified by name when a user positions the cursor over the map (see Figure 3). On clicking a structure, a user can peruse its history and photographs if available.

Take in Figure 3 Here, There and Everywhere

"Memories in the Shell of Time: Old Hill Street Police Station" is the next window and it explores the building's history and origin (see Figure 4). Three sub-sections are available. The first gives a summary of the building's facts, explains the architectural concept and features, and summarizes the construction process related to the MITA structure. The next section narrates the history between 1819 to 1980, while the third describes the events and activities that the previous occupants were involved in. Photographs in the form of slide shows are available and each may be zoomed in for greater detail.

Take in Figure 4 Memories in the Shell of Time

The fourth window, "A Present from the Past: The Transformation to MITA Building", allows users to navigate to relevant information by clicking on interactive floor and building plans of the Old Hill Street Police Station. Users can view the changes that took place over the years in its preservation, restoration and conversion to the present MITA Building (see Figure 5). Changes are described by text and thumbnail images that can be enlarged.

Take in Figure 5 A Present from the Past

"The Wind Beneath a Nation's Wings: Ministry of Information, Communications and the Arts" is the fifth window and portrays the history of the various occupants of the MITA Building. A menu is provided as a timeline at the bottom of the main page. For each time period, users can view images and textual descriptions by selecting each time period. Images may be enlarged for greater detail.

"Contact Us" is the last section in the exhibition and provides contact information for users. It also summarizes the entire project and provides an online gallery containing all the images used in the virtual exhibition.

Tools and resources used

The virtual exhibition's content and images were culled from existing resources provided by the NAS and are the same as those used in the Flash version. A plain text editor was used to code the exhibition, while Adobe Illustrator 10 was used to create additional graphics (for example, the NAS logo) and export them as SVG files. Testing was done with Internet Explorer 5 and 6 as well as Netscape 4.7 and 6.2, with the Adobe SVG Viewer 3.0 (http://www.adobe.com/svg/viewer/install/main.html) used as the SVG Web browser plug-in.

In addition to SVG books, the Web proved to be a valuable source of SVG material, and useful for developers both novices and experts alike. For example, the forums at W3C (http://lists.w3.org/Archives/Public/www-svg/) and Yahoo! SVG Developers (http://groups.yahoo.com/group/svg-developers/) are places where SVG users and developers can pose questions, problems and solutions, and express their views on SVG. They are relatively reliable resources when printed books do not offer any solution to a particular problem.

Colours in the Wind – lessons from the SVG alternative

As discussed previously, several issues encountered during the development of the Flash version of the virtual exhibition led to the present work in investigating SVG as an alternative technology for future projects. To accomplish this, we sought to develop the entire virtual exhibition in SVG and document areas of strengths and weaknesses of the technology. All sections of the virtual exhibition were created as SVG files and DOM manipulation through ECMAScript (also known as JavaScript) was used only when SVG could not provide the necessary level of interactivity. The remainder of this section describes the techniques used in developing the SVG version of *Colours in the Wind* as well as the issues encountered and lessons learnt during development. As such, sections of the exhibition that employ similar techniques will not be highlighted.

Overall design strategy
The overall look-and-feel of the SVG virtual exhibition was designed to be similar to the original exhibition, and only modified to address usability issues or when features could not be implemented exactly. With regards to the former, two distinguishing features were introduced (see Figure 6). These were a navigation bar and a customisation toolbar, and together they served as a common template used throughout the exhibition to establish a sense of consistency (Nielsen, 2000). In contrast, the individual sections in the Flash virtual exhibition did not follow a common design template.

Take in Figure 6 Navigation bar (top) and customisation toolbar (bottom)

The navigation bar is incorporated into the overall template and visible at the top left corner throughout the entire exhibition. This is an additional feature to enable users to jump from one section to another easily because the six sections are displayed in a single window, unlike the Flash version that opened a new window for each section. A simple customisation toolbar for changing font attributes is positioned at the bottom of the screen. This is to allow users to control their view of certain elements in the SVG exhibition, and partly to save them the trouble of zooming in to read the text of the virtual exhibition.

Most of the pages of the exhibition were designed to fit into the browser window regardless of the screen size so that users will not have to use the zooming and panning tools. Each section of the exhibition is contained in one SVG file so that the template and other common elements will not have to be reloaded again as each page is refreshed.

A new mindset required for SVG development
For graphic or Web page designers, developing SVG graphics requires one to think like a programmer. The developer has to plan ahead to prevent unnecessary changes to the design or the code since this could create more debugging problems. It is crucial to comment SVG code because the developer may not remember what was

done in the code, or someone else on the team may need to modify the files sometime in the future. Comments should be made visually prominent where possible because chances are that a complicated SVG file may contain a few thousand lines of code and tracing the bugs becomes an uphill task.

Time is needed to get used to SVG as its concept of rendering graphics through XML-based code is still relatively novel. This is true especially for a graphic designer with no prior knowledge of markup or programming languages. Issues long confined to the domain of software development, such as, for example, whether the code is syntactically correct (valid in SVG's case), or if the SVG MIME type is registered on the server, must be considered in order to get an SVG file to display correctly.

Layering in SVG

In image editing software, such as Adobe Photoshop, layers help designers to prototype the visual end product of an image while keeping all the elements separated in case amendments are required. On the Web, HTML layers using the <*div*> tag are used for interactive purposes or for positioning content. SVG also encompasses the concept of layering in that all elements are essentially free-floating, resulting in more flexibility than HTML-based Web pages. In addition, the opacity of an element can be adjusted so that other elements underneath it show through.

The opacity and layering features were employed in "A Time and a Place" to allow users to enlarge portions of the maps by dragging a magnifying glass. Here, each map consisted of a pair of identical images of different sizes – normal and enlarged (see Figure 7). The enlarged map was placed at the bottommost layer and rectangular elements were placed over it, obscuring the entire image save only for a small window that allows part of the map image to be visible. As the user drags the magnifying glass over the smaller map, the enlarged map is correspondingly moved in the same direction using ECMAScript, thus making part of the map visible through the window. The magnifying glass was created by defining a rectangular element and placing it at the topmost layer. To achieve translucency so that the normal map image underneath it could be seen, its opacity was set to 0.5 (1 being opaque and 0 being transparent).

Take in Figure 7 Using the layering and opacity features for "A Time and a Place"

Absence of scrollbars

The fact that scrollbars are absent from an SVG Web page may go unnoticed if the content fits entirely into the browser window. The developer, however, has to make sure that the viewing area is specified properly and the content is visible in its entirety, regardless of the browser window being used. If any part of the rendering bleeds off the edge of the browser window area, or if the entire SVG image takes up additional space outside the visible browser window, scrollbars will still be non-existent, unlike HTML pages. In this case, users will have to depend on the zooming and panning capabilities of the SVG viewer to see the obscured content.

In order to make scrollbars visible, the SVG DOM needs to be manipulated via a script. Since one of the goals of this project was to minimize the amount of scripting, the SVG exhibition was designed so that all the content fits into the browser window area. This was possible because SVG provides the capability to create content that stretches to fit a particular container element or the browser window.

Text editing

One difficulty in conforming to the design template is that SVG does not support word wrapping. Throughout the development of the SVG exhibition, word processing software such as Microsoft Word was used to "typeset" the length of each individual line of text, which was then copied and pasted into the text editor and enclosed with the appropriate SVG tags. Pixel measurements were used extensively while laying out the various elements of the exhibition. This is a major problem that most SVG developers and designers have highlighted – the tedious text editing and copy fitting that have to be performed whenever text is inserted or changed.

This problem was compounded because of the customisation options provided for users to change the type and size of the font (see Figure 6). Here, care had to be taken to ensure that the space allocated to the text should accommodate the extremities of values allowed in terms of font sizes. It was thus necessary to determine whether there was enough space to view all the text if the biggest point sizes or widest fonts were chosen, or whether there was too much white space if the smallest point size or narrowest font was chosen.

SVG viewer incompatibilities

In HTML, when a link is moused over, the cursor will change to a hand cursor by default. Likewise, various cursor types are included in the SVG Specification 1.0, but at the time of this project not all SVG viewers supported this feature except for the CSIRO SVG Toolkit (http://sis.cmis.csiro.au/svg/status.html). As a result, users will not know that certain SVG elements are links that will trigger an action when clicked. A fix that was done was to "trick" the SVG viewer into displaying the hand cursor using a dummy anchor (<*a*>) tag with no file specified.

A similar problem involved SVG's <*title*> tag. This tag is equivalent to HTML's *alt* attribute and provides the opportunity to display information about a portion of the SVG content typically through tooltips. Unfortunately SVG viewers are not obliged to render the contents of the <*title*> tag. To achieve this effect some amount of scripting had to be performed.

In general, because of the newness of the SVG standard, not all features listed in the specifications are implemented in the popular SVG viewers. There were several occasions when design decisions were made based on the assumption that certain listed features would be available in SVG viewers only to discover otherwise. This led to some amount of redesigning to work around the inadequacy and/or to find an alternative to that feature. The latter typically involved some amount of scripting. One example was the use of ECMAScript to create the tooltip feature described above. Consequently, careful planning is advised before embarking on any SVG Web project. It is imperative that a close examination of the targeted SVG viewers is conducted before any major design and coding is undertaken.

Finally, current Web browsers do not natively support SVG and users need to download and install an appropriate viewer. Although this is also the case for Flash, there is less confusion because the Flash player is developed by only one vendor – Macromedia. On the other hand, with SVG being a new and open standard, there are at least five different viewers available and users may not know which viewer to download. This problem is also compounded by the fact that the different viewers support the SVG standard unevenly so that certain features may work on a particular viewer but not on another.

Comparing the SVG and Flash versions

The SVG and Flash versions of the *Colours in the Wind* virtual exhibition were put to the test by conducting a qualitative comparative study. Although both versions had the same content, there were some visual differences and thus the aesthetic aspects of the virtual exhibition were not considered in this study. Instead, the purpose was to determine usability issues regarding the use of SVG as an alternative to Flash, as well as general opinions about the technology.

Participants
Three males (M1 – M3) and six females (F1 – F6), ages ranging from 24 to 38, took part in the pilot study. All had computer experience of between four to 20 years and used the Internet everyday. Their backgrounds were varied, coming from statistics, information studies, biotechnology, mechanical engineering, mass communication, architecture, finance and hotel management. Five had Masters degrees, while the other four had Bachelors degrees.

Protocol
Each participant was assigned a computer with both Macromedia Flash and Adobe SVG viewers installed. Participants were given two URLs for the two virtual exhibitions and asked to use Microsoft's Internet Explorer to access them. They were not told whether the exhibitions were Flash or SVG based. A facilitator, who was a developer in the SVG version of the virtual exhibition, worked with each participant individually to elicit opinions. The facilitator's role was to guide participants and make clarifications so that they could complete the sessions.

Each session was divided into three parts. In Part 1 participants were introduced to the two versions of the virtual exhibition and encouraged to explore them to gain familiarity. Part 2 required participants to complete a set of tasks in both versions. These included general site navigation, copying and pasting text, and controlling playback of the virtual exhibitions' animations. The facilitator avoided guiding participants in the tasks since individual reactions to SVG and Flash were crucial to the study and doing so would shape their impressions either positively or negatively. Upon completion of the tasks, participants' opinions on the ease of use of each virtual exhibition were elicited through a questionnaire in Part 3. A short open-ended interview was also conducted to give participants an opportunity to provide information not captured in the questionnaire.

Findings and analyses

Overall impressions
All participants, except one (M1), initially thought that both virtual exhibitions were Flash-based. One reason given was that some participants had only heard of Flash and upon seeing animation assumed the exhibitions were built with Flash. The only participant (M1) who could tell the difference pointed out the Flash animation "always has a loading prompt before running" while the other did not. Participants F5 and F6 had heard of SVG but they did not expect that current Web sites could have been developed using SVG. Further, they were not able to discern any significant variations between the exhibitions. This indicates that the quality of SVG graphics is comparable to Flash and that real-world projects can be developed using SVG.

Most users preferred SVG over Flash, attributing their choice to the faster downloading times and better responsiveness to user actions on the SVG version of the virtual exhibition. F4 noted that it "was so much faster to access the individual exhibits in the SVG site. Also, the response time for the individual exhibits in the Flash version was slow when I clicked on the links." Only one user (F2) felt that Flash was still superior to SVG on the whole. F2 commented that some Flash animation sequences flowed more smoothly than the SVG version. Further, while individual exhibits were being downloaded, the Flash version displayed a progress indicator while the SVG version did not, and this might cause uncertainty as to whether an error had occurred.

Playback and navigation
The ability to control playback of the virtual exhibition's animation sequences is crucial since users do not process information at the same rate. The task of pausing and resuming the playback of both versions of the virtual exhibition produced mixed responses. Four participants, M3, F1, F5 and F6, were able to pause the Flash animations. When asked to rate the level of difficulty of this task from 1 (easiest) to 5 (most difficult), M3 and F1 found it difficult (rating of 4), while F5 and F6 found it easy (rating of 1 and 2 respectively). On the other hand, six participants were able to pause the SVG animations, with two of them finding it easy (rating of 2), while the remaining four found it very easy to accomplish the task (rating of 1).

It is technically possible to pause animations in both versions of the virtual exhibition. In the SVG version the viewer provides a context-sensitive menu with a "Pause" option. Selection of this option places a check-mark next to it. From this paused state, selecting this option again resumes playback. On the other hand, Flash supports pausing but in a reverse direction. That is, it has a "Play" option that toggles between play and pause. Further, most users were easily able to rewind and fast-forward animations in the Flash version but were not able to do so for the SVG version. For example, F6 said that she "was not able to control the animation speed while using the SVG version". Taken together, the responses indicate that both the Flash and SVG viewers require improvements in their playback functions to give users greater control over how animations are displayed.

Besides manipulating playback controls, panning is another built-in feature common to both the Flash and SVG viewers. This dragging operation allows users to view graphics that are not visible within the viewing area. Seven participants figured out how to perform panning for the SVG version of the virtual exhibition. Of these, when

asked to rate the level of difficulty of this task from 1 (easiest) to 5 (most difficult), four said it was very easy to do so (rating of 1), while the other two indicated it was easy (rating of 2). Only one participant (M1) initially found it difficult to pan, as he did not know how to access this operation through the SVG viewer and thus gave a difficulty rating of 4. The Flash version fared worse as only two users could figure out how to perform the pan operation. One possibility could be that with SVG the panning function is available at all times, but with Flash users are only allowed to pan when the graphics have been magnified. Interestingly, in Flash panning cues are provided by the appearance of a hand cursor when this operation is available, whereas in SVG no such cues are available. Yet, a majority of the participants could actually perform panning in the SVG version of the virtual exhibition in contrast to the Flash version. This indicates that the introduction of SVG into Web sites is viable as users are able to adapt to the technology.

Copying, editing and saving

Editing and saving content is a double-edged sword. On the one hand, these operations allow coding changes to be made quickly and conveniently and also support learning as users (such as students) can copy portions of text to incorporate into other documents. A major problem with these operations, however, concerns intellectual property issues and the potential for plagiarism. Nevertheless, these operations were evaluated by the participants on the assumption that the advantages outweigh the disadvantages.

Participants were asked to go to "A Present from the Past" and transfer the introductory paragraph to a word processor (Microsoft Word) for editing. All users found it easier to complete the task successfully for the SVG version than for the Flash version. However, one user (F3) noted that this operation was not intuitive for the SVG version because the standard “copy” shortcut key (Ctrl-C) did not work. Instead, participants had to use the SVG viewer’s context-sensitive menu. Despite this, participants remarked that this operation was at least available in contrast to the Flash version. Here, participants had to retype text using the word processor in order to accomplish the task. F5 commented that “I can copy the text right off the page, even though it’s troublesome, unlike Flash where it’s almost impossible to do so.”

Saving a local copy of the SVG and Flash virtual exhibitions was the last task participants were asked to perform. Six users found it easier to do so for the SVG version, while one user found it easier to save the Flash version. The other two users had problems saving both files. This indicates that SVG files were easier to download from the Web to be made available for offline viewing or editing. The fact that SVG is XML-based makes code changes easier as well.

Preferences and opinions

The questionnaire asked participants their opinions on various aspects of SVG and Flash. Table I summarizes the responses and indicates that participants felt that SVG outperformed Flash in terms of downloading speed, quality of animation, response time and the ability to view source code. Participants also felt there were no major differences in the quality of textual elements and the menu options available for both the SVG and Flash viewers.

Take in Table I Participants' responses on SVG and Flash usability features

The most notable feature that participants seemed to like in SVG is that it downloaded faster than Flash and that they had access to the source code. F5 said that "SVG seems to be more open and more accessible." Even on a 56Kbps modem, F1 and F4 reported that SVG still downloaded faster than Flash. M2 also pointed out that SVG had a faster page download compared to the Flash file even when the files were already stored in the local cache.

Table I also shows that the number of votes in the Flash column was never greater than the number of votes in the SVG column, indicating that SVG was well received by the participants. Participants' comments also seemed to support this view. For example, F5 stated that "SVG is just as good if not better", while F6 concluded that SVG "will no doubt contribute to the betterment of the computer design industry". However, it should be noted that generalizations could not be made due to the small sample size. Nevertheless, the results do indicate that SVG is a potential alternative to Flash, and that users are able to successfully interact with the applications created with it.

Conclusion

This paper describes a project investigating the use of SVG as an alternative technology for developing virtual exhibitions for the National Archives of Singapore. As discussed, we found that SVG was relatively robust and, in general, whatever could be implemented in the original Flash version of the virtual exhibition could be replicated in the SVG version. Depending on the complexity of a SVG project, there are times when scripting languages (such as ECMAScript) have to be used in order to provide features intended by the developer. This partly arises from the fact that some parts of the SVG specification are not supported by existing SVG viewers, probably due to the newness of the standard. Another reason for the use of scripting is that certain features can only be supported through the manipulation of the SVG DOM elements. Fortunately, because SVG is XML and hence text-based, it is easy to incorporate code snippets from other developers. These are available through many SVG-oriented Web sites.

A comparative study of the usability features between the SVG and Flash versions of the virtual exhibition was conducted. The study found that participants were generally positive towards SVG and, when asked to provide ratings for a list of usability features, SVG consistently received more positive votes than Flash. Although this initial study used a small sample size, it suggests SVG's potential as a development platform for future NAS virtual exhibitions.

However, one of the few things standing in the way of the widespread adoption of SVG is that users are currently required to download and install an appropriate viewer. It is hoped that both Internet Explorer and Netscape will provide native support for it in the future, but in the meantime "this lack of browser support will create some hesitancy in the market" (Trippe and Binder, 2002). Some Web designers are also worried that the nature of SVG as an open standard will subject their art and

design work to plagiarism and that they will not be able to protect their intellectual property. It is also greatly desired for search engines to begin indexing SVG Web pages to extend the reach of the technology.

Nevertheless, there is a great potential for SVG in the future and the number of people contributing to its development is increasing. Many enthusiasts are providing constructive feedback and proposing improvements with regards to SVG's current problems and flaws. If these pending issues are resolved, and the standard continues to be backed by industry leaders and the W3C, SVG will be more accessible to the public, leading to a truly user-centred and user-driven open technology.

References

Bowler, J., Brown, C., Capsimalis, M., Cohn, R., Cole, L. Deweese, T.E., et al. (2001), *Scalable Vector Graphics (SVG) 1.0 Specification*, available at: www.w3.org/TR/SVG/

Heminger, A.R. and Robertson, S.B. (2000), "The digital Rosetta Stone: a model for maintaining long-term access to static digital documents", *Communications of the Association for Information Systems*, Vol. 3 No. 2, available at: cais.isworld.org/articles/default.asp?vol=3&art=2

MacTavish, S.H. and Pickard, M.R. (1999), *Electronic Digital Imaging Standards for Archiving Records*, available at: www.c3i.osd.mil/org/cio/i3/recmgnt/1june1999imaging.pdf

Nielsen, J. (2000), *Designing Web Usability*, New Riders Publishing, Indianapolis, IN.

Probets, S., Mong, J., Evans, D. and Brailsford, D. (2001), "Vector graphics: from PostScript and Flash to SVG", *Proceedings of the 2001 ACM Symposium on Document Engineering*, pp. 135-143.

Siam, K. (2001), "Transforming centres of excellence – the National Heritage Board experience", *Library Review*, Vol. 50 No. 7/8, pp. 366-73.

Sommerville, I. (2001), *Software Engineering*, 6th ed., Pearson Education Limited, Essex, England.

Trippe, B. and Binder, K. (2002), *SVG for Designers: Using Scalable Vector Graphics in Next-Generation Web Sites*, McGraw-Hill Osborne Media, New York.

W3C (2001a), *Fonts*, available at: www.w3.org/TR/SVG/fonts.html

W3C (2001b), *SVG Document Object Model*, available at: www.w3.org/TR/SVG/svgdom.html

W3C (2002a), *Graphics Activity Statement*, available at: www.w3.org/Graphics/Activity

W3C (2002b), *W3C Scalable Vector Graphics (SVG) – History*, available at: www.w3.org/Graphics/SVG/History

Watt, A.H. (2001), *Designing SVG Web Graphics*, New Riders Publishing, Indianapolis, IN.

Zand, M. and Samadzadeh, M. (1994), "Software reuse issues and perspectives", *IEEE Potentials,* Vol. 13 No. 3, pp. 15-19.

Figure 1 Front page of *Colours in the Wind*

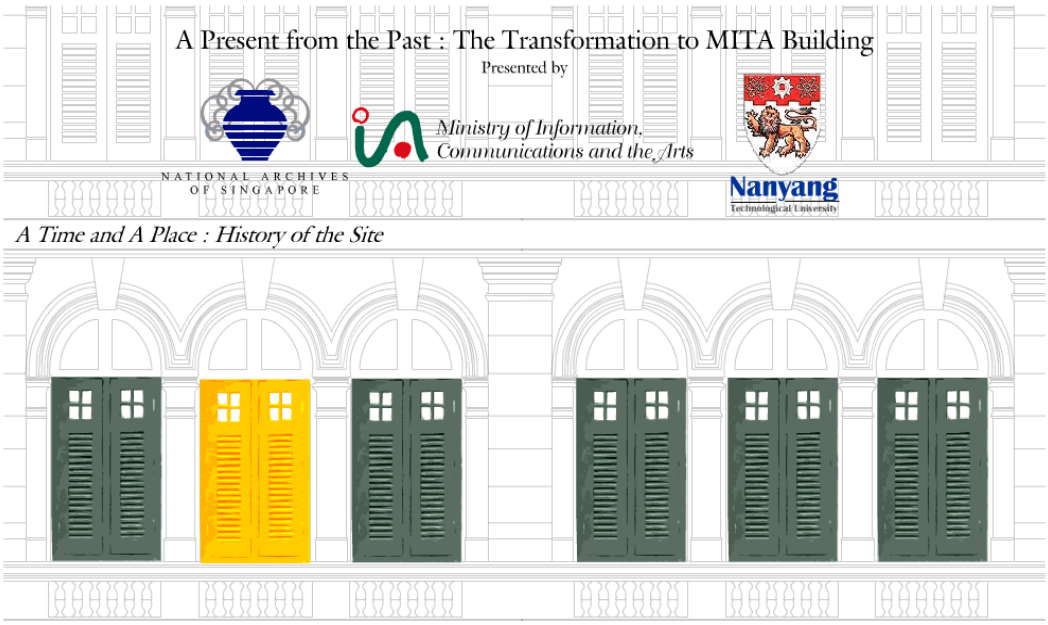

Figure 2 A Time and a Place

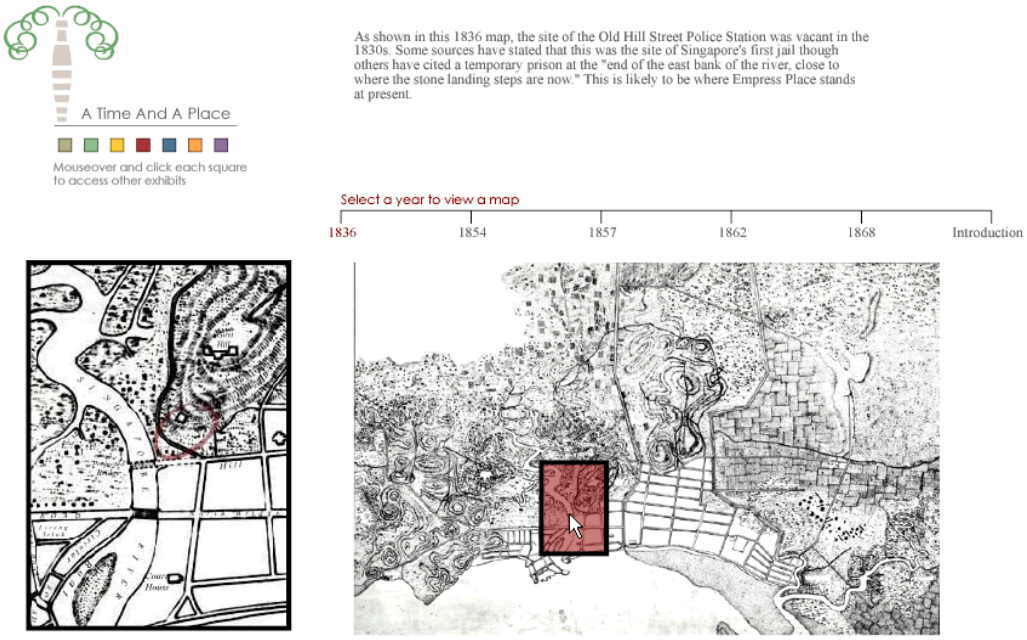

Figure 3. Here, There and Everywhere

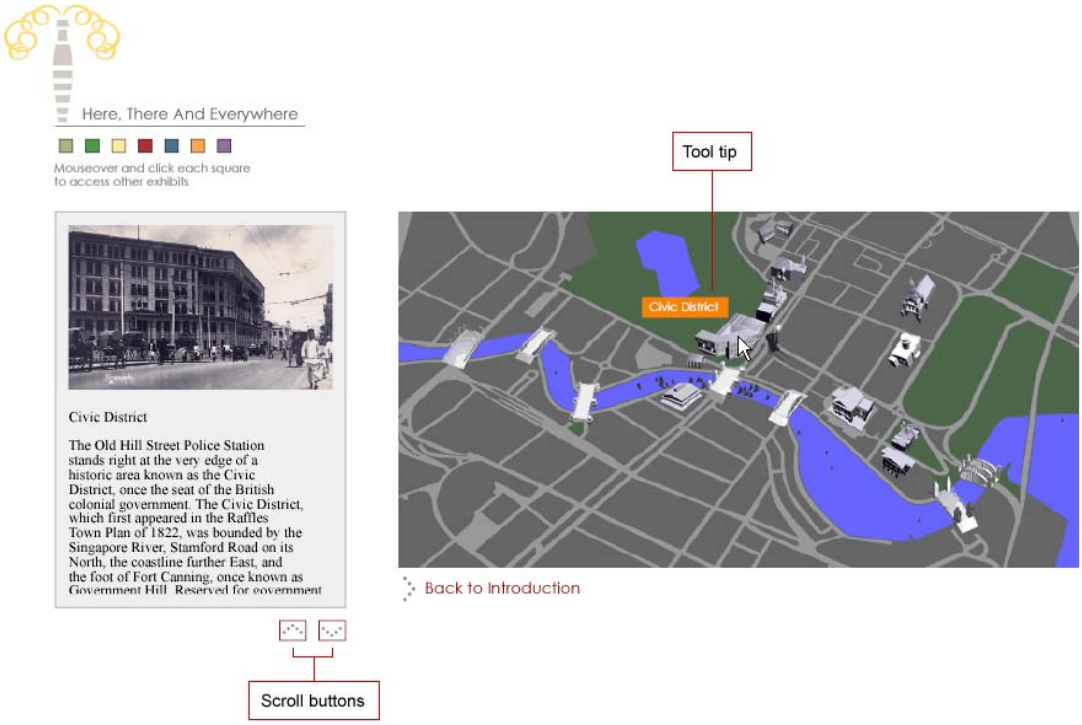

Figure 4 Memories in the Shell of Time

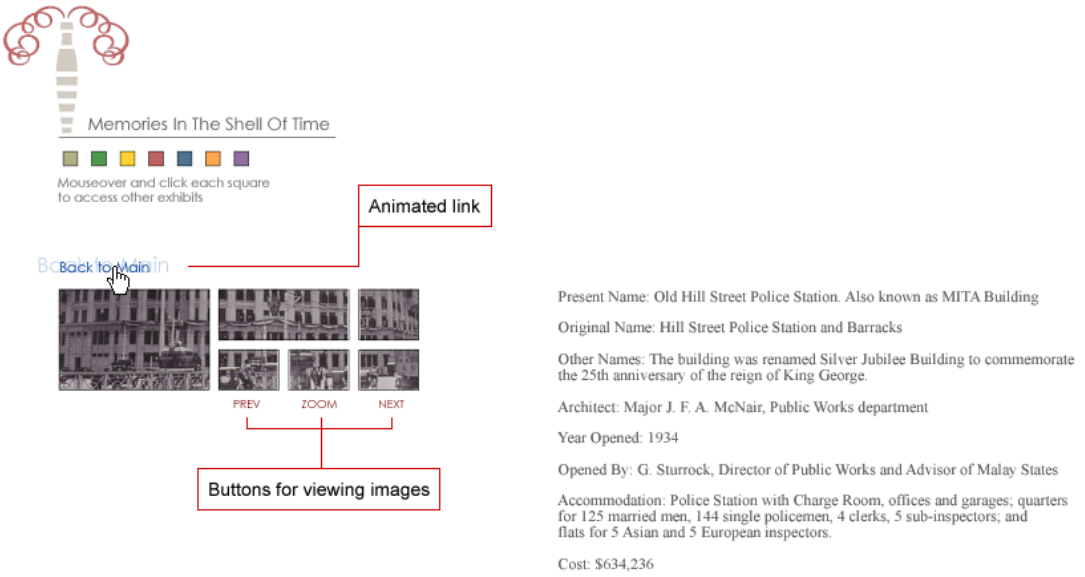

Figure 5 A Present from the Past

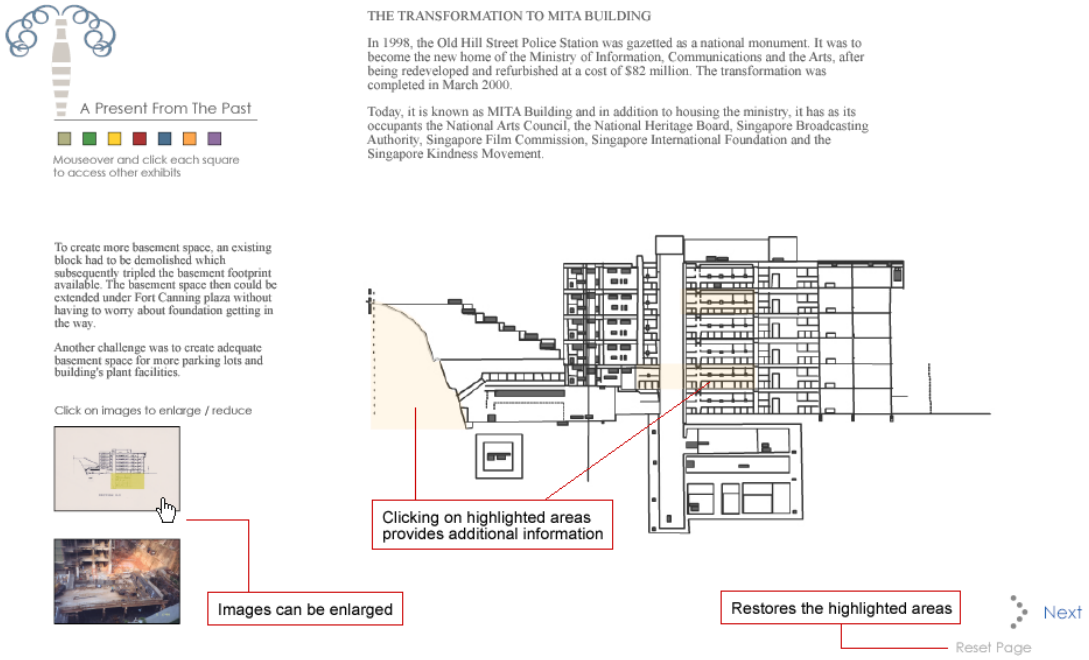

Figure 6 Navigation bar (top) and customisation toolbar (bottom)

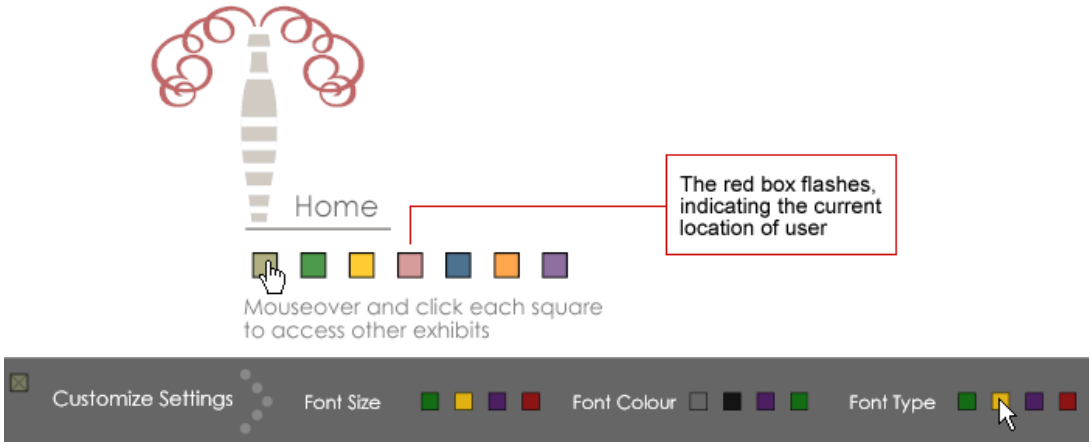

Figure 7 Using the layering and opacity features for *A Time and a Place*

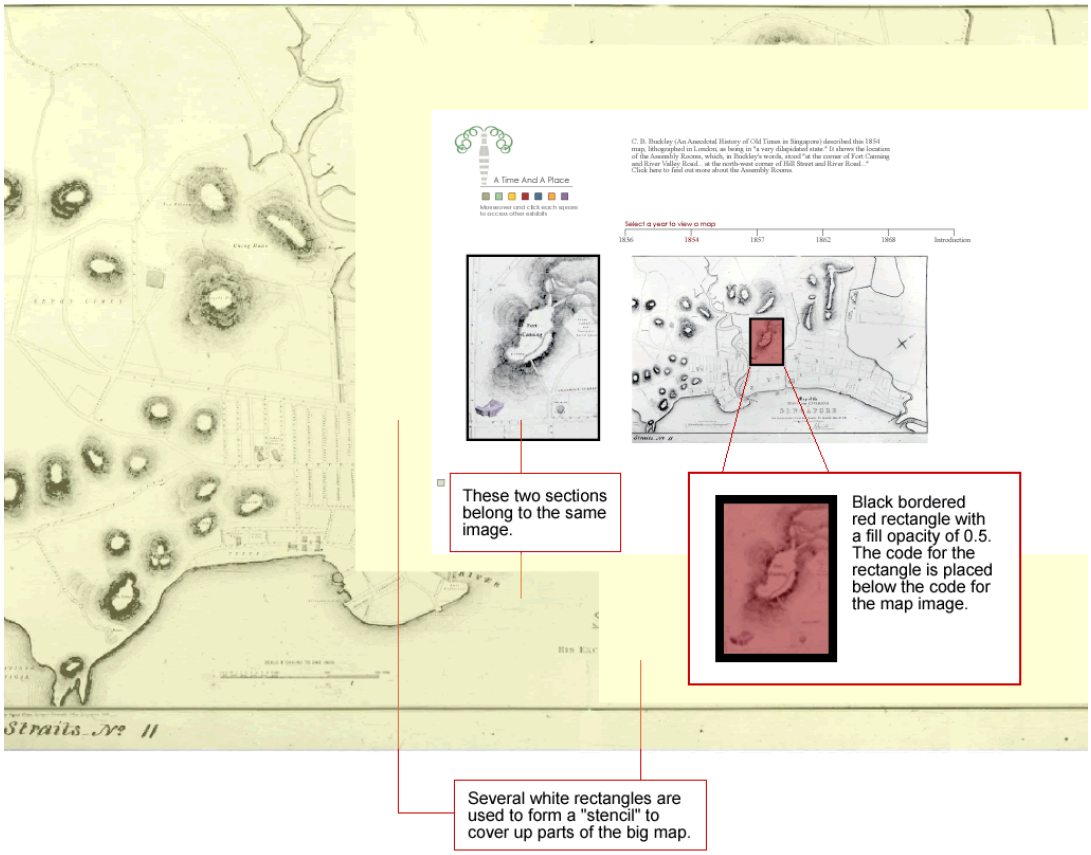

Table I Participants' responses on SVG and Flash usability features

Feature	SVG	Flash	No difference
Downloading speed	8	0	1
Quality of animation (e.g. smoothness of transition)	5	1	3
Quality of graphics (e.g. resolution)	5	1	3
Quality of textual elements (e.g. clarity)	3	2	4
Response time	7	1	1
Plug-in menu options	3	2	4
Viewing source code	7	2	0