# XML Based Visualization in Archaeology

Holly E. Wright

Department of Archaeology
University of York
York, UK
hew503@york.ac.uk

## Abstract

This paper centers on ways archaeologists can present spatially derived visual information on the Web, and discusses the challenges and opportunities surrounding the use of vector images, and their relationship to archaeology. In particular, it explores Scalable Vector Graphics (SVG), which is the XML application for describing vector graphics on the Web, as a potential tool for archaeologists. It includes some of the ways vector graphics are used in archaeology, and outlines the development and features of SVG, which are then demonstrated in the form of a case study.

## 1 Introduction

Archaeologists have been using vector-based digital tools for years (Eiteljorg 1989), and today they are used routinely in many areas of archaeological work. This method of organizing visual information using points, lines, and polygons is a unique system that can inform the archaeological process. Unlike raster images, which are made up of an array of disparate pixels (Jones 1997), vector images exist within a defined set of coordinates (Eisenberg 2002:2). The ability of a vector image to "know where it is" is significant for a spatial discipline like archaeology. Vector graphics are resolution independent, and display the same clarity at any level of magnification (Laaker 2002:7). Vector graphics can also be used to organize drawings into layers (Watt 2002:xvii), and the pan and zoom features make different forms of analysis possible as well (Eiteljorg 1989). Raster (images created using pixels) and vector graphics are not competing forms of visualization; rather, they are different formats for different purposes and are sometimes most powerful when used in combination.

Creating and interpreting excavation plans, sections, and site maps is an important part of visualization in archaeology. Without this fundamental spatial information it would be almost impossible to make sense of any archaeological fieldwork. The transition of this information to a digital format began in the late 1980s with archaeological survey teams using vector-based CAD programs (Eiteljorg 1989; Middleton 1998:6). CAD drawings can subsequently be incorporated into a variety of other formats that use vector-based information like GIS programs. Today, most GIS programs have the ability to create visualizations that are both raster and vector based, and to use them in combination (Wheatley and Gillings 2002:16). In many cases, CAD drawings are also the basis for three-dimensional Virtual Reality (VR) modeling, which also combines raster and vector technology to create a unique form of visualization (Terras 1999).

## 2 Archaeology, Vector Graphics, and the Web

### 2.1 Using Archaeological Vector Graphics

Archaeologists have been displaying rasterized vector images on the Web for years, but making them available as true vector images that are functionally comparable to the program used to create them can be problematic. The image may be created in layers that can be turned on or off, and the user controls the way elements are displayed. The ability to pan or zoom to create different views of an image without loss of resolution may be necessary in order to interpret the image. The image may reference a database, and interaction with the database through the image may be the most effective way of accessing the associated data. The image may be generated by a database itself and only exist based on a user-defined query. All of these functions are beyond the capabilities of any static raster image, but are fundamental to much of the vector-based work currently undertaken in archaeology. How this functionality can be presented on the Web is a question that needs more exploration. There are a variety of solutions available, each with their own advantages and disadvantages. The fact that no consensus has developed indicates that a satisfactory solution remains elusive (Cagle 2002:9).

The publication of large plan and section drawings has always been problematic in archaeology, both in print and on the Web. Using vector graphics can address some of these problems, but it is important to consider the inherent differences associated with use of this technology, as well. For example, a hard copy of a section drawing may be over a meter long, but once digitized into vector format it can be presented at any level of detail or in its entirety without loss of resolution, and the user can choose which areas are of interest (Figure 1). This is consistent with the trend towards publishing more completely in archaeology, and not just synthesizing conclusions (Clarke et al. 2003:402; Livingood 1996). Although only one of many decisions archaeologists make during the analytical process, by using

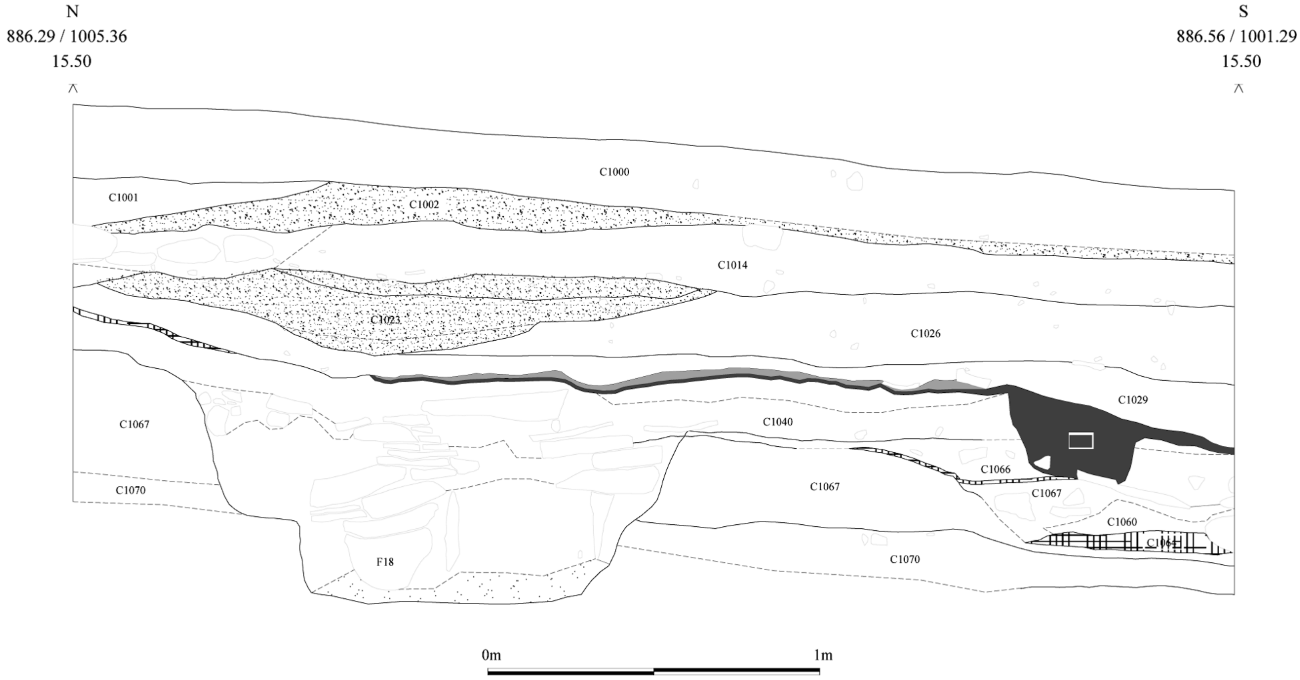


*Figure 1. Example of a section drawing created in AutoDesk's AutoCAD. Excavations in advance of an oil tank (Sector 2, Intervention 26) section view from the Tarbat Discovery Programme (Carver 1998), The Settlements at Tarbat, figure 5).*

vector graphics in this way, an author or editor would not have to pick and choose which details of an image to use, and so a layer of subjectivity may be removed.

Use of layers in archaeological vector drawings can be very helpful as an aid to interpretation, as well. They may help simplify a complex drawing, or highlight areas that may be of particular interest, without having to choose a detail (Judith Winters pers. comm., July 2003). Because layers can be turned on or off, the user can decide which is most important, and in what combination (Eiteljorg 1989). Layers can show how a site or section is organized or how it has changed over time; but it can also illustrate the process that went into creating a drawing (Figure 2). In particular, this type of visualization can be used as an important tool to help identify and communicate uncertainty in archaeological data, and provide alternative interpretations (Ryan 2002).

Computers are perceived as being more accurate than humans; therefore, digital visualizations tend to convey a sense of being more factual than images on paper. In reality, authors will frequently have to combine digital data in which they have varying degrees of confidence in order to create something that communicates the information they are trying to convey (Miller and Richards 1995:20). If a drawing is divided into layers reflecting how conjectural the data are, it leaves the user to decide whether they are in agreement with the ideas presented by the author and makes the interpretation more transparent. Although vector graphics divided into layers may be used to show uncertainty, the way in which layers are chosen and divided is still the result of a subjective decision.

Vector-based information is very significant to archaeologists, as shown by the results of *The Publication of Archaeological Projects: a user needs survey* (PUNS) report, published by the Council for British Archaeology. It places maps, plans, and sections as third in importance, only behind the introduction and conclusion in an archaeological report (Jones et al. 2001). Even photographic information is not rated as highly. This is even more significant, as the results of the survey indicate that very few people read a publication in its entirety (Figure 3).

The PUNS survey was meant to evaluate the usefulness of archaeological project publications generally, and reflects the way project reporting and analysis has been traditionally communicated. In contrast, the recent Historic Environment Information Resources Network (HEIRNET) User Survey was designed to assess the needs of individuals and organizations, specifically using digital resources for archaeology and the historic environment (Brewer and Kilbride 2006). This survey produced some very interesting contrasts between what archaeologists find useful generally, and what is useful when it is presented in an online format (Figure 4).

While the results are not exactly comparable, there are some significant differences that are of interest, especially with regard to archaeological visualization. Maps rated extremely highly, as would be expected based on the PUNS report. In contrast, graphics, which would be expected to include elements like plans, sections, and other types of vector-based spatial information, received the lowest rating. In fact, of the 118 individuals who identified themselves specifically as archaeologists, only five indicated online graphics were "very useful." Based on this information, it is possible to conclude that a significant gap has developed between the type of resources archaeologists rely upon for their research, and the ability of Web technology to deliver those resources in a useful way. If this is the case, considerable work needs to be done to improve how vector graphics are presented on the Web. The advent of mature Web technologies like XML and their related concepts that are now

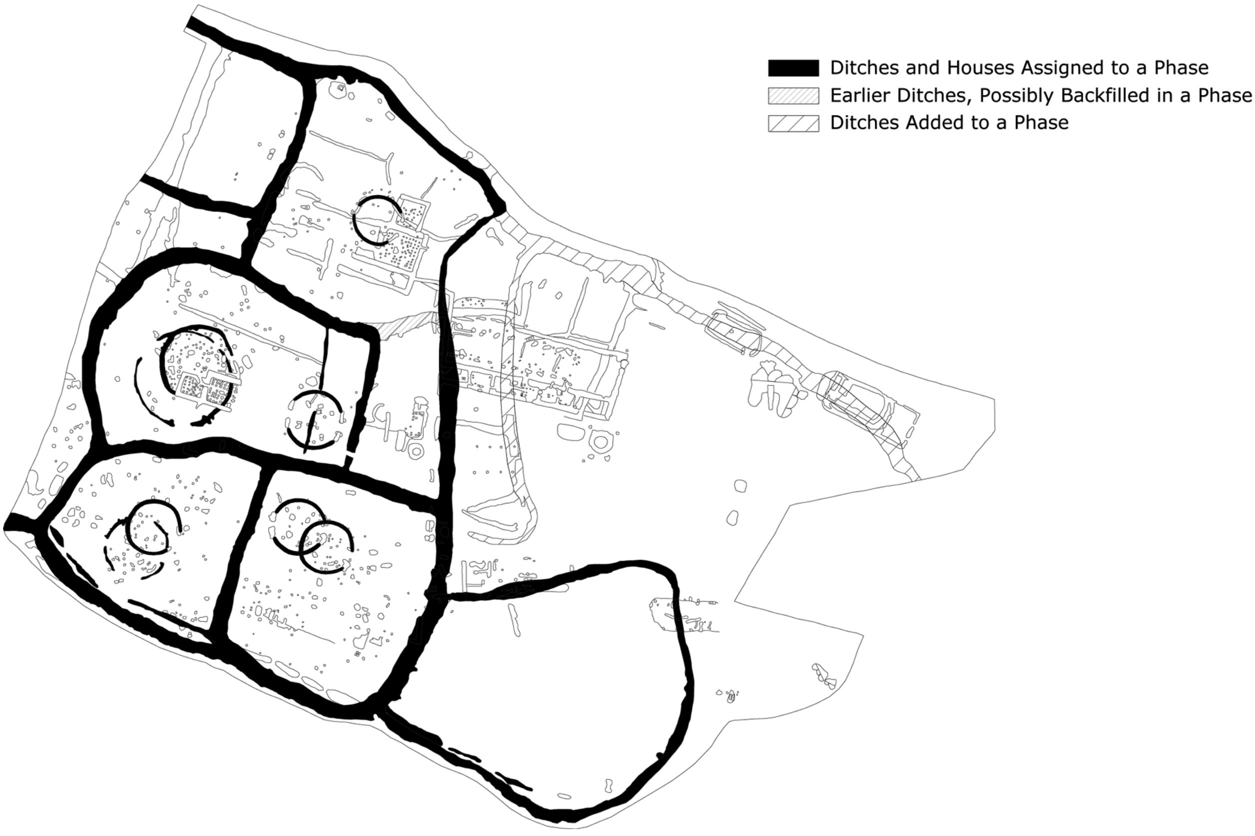


*Figure 2. Example of a layered plan drawing created in AutoDesk's AutoCAD. From the Dalton Parlours Iron Age Settlement and Roman Villa in the parish of Collingham, West Yorkshire. Drawing shows all five Iron Age settlement phases, as designated by post-excavation analysis by the West Yorkshire Archaeology Service. Digitized by the author from the Dalton Parlours excavation report (Wrathmell and Nicholson 1990).*

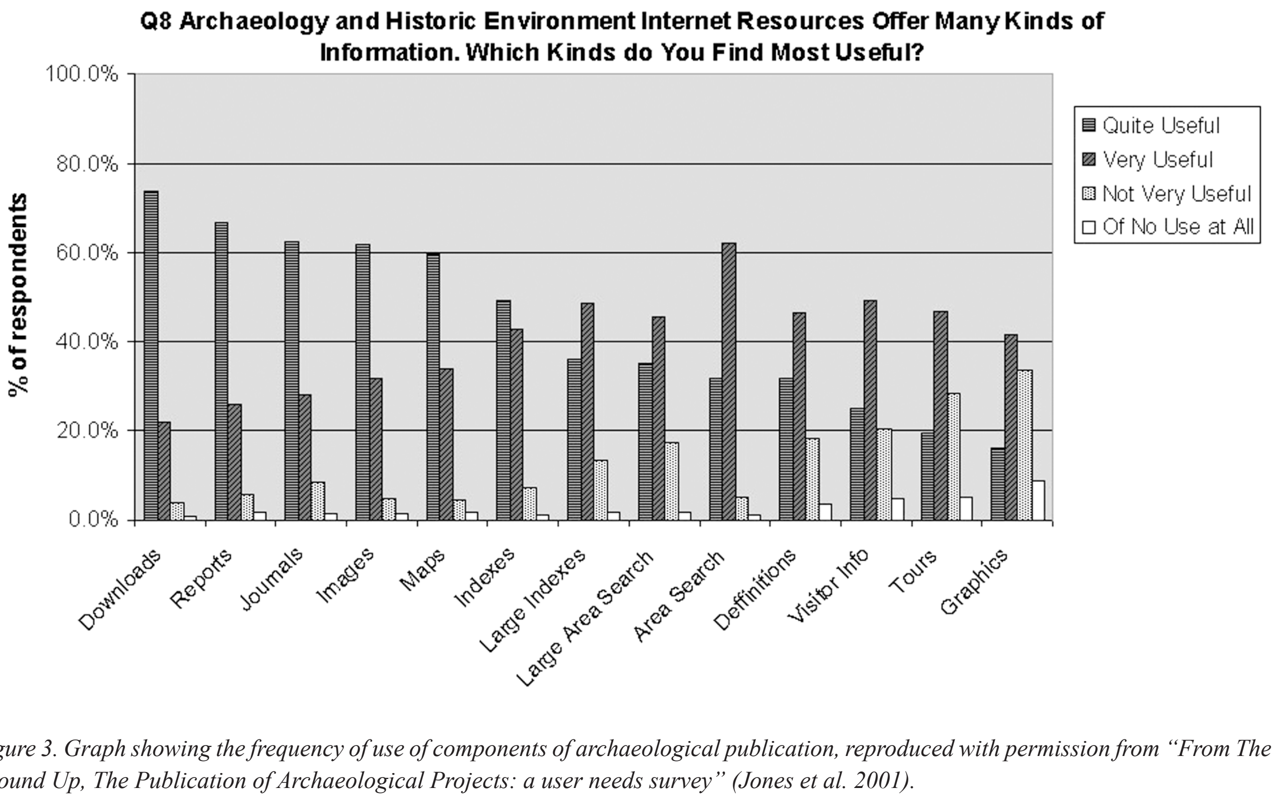


*Figure 3. Graph showing the frequency of use of components of archaeological publication, reproduced with permission from "From The Ground Up, The Publication of Archaeological Projects: a user needs survey" (Jones et al. 2001).*

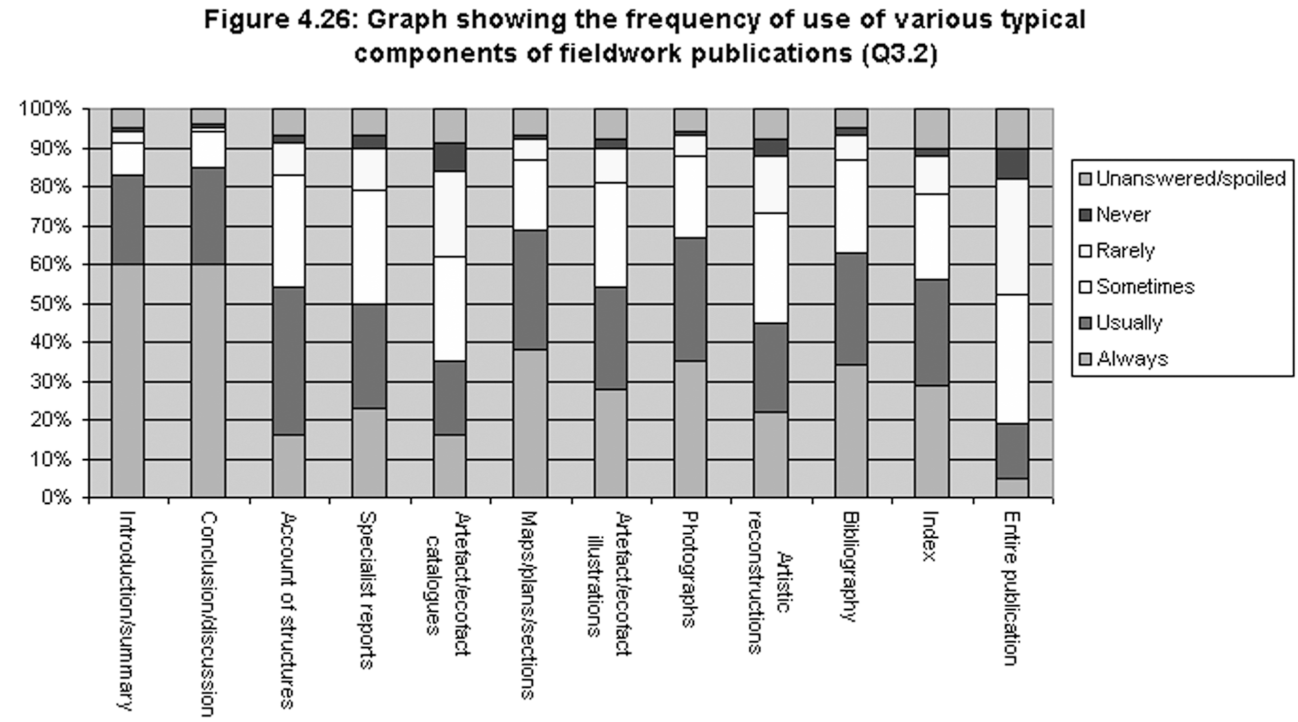


*Figure 4. Graph showing the usefulness of different kinds of Internet resources to practitioners working in archaeology and the historic environment, reproduced with permission from the HEIRNET User Survey 2005 (Brewer and Kilbride 2006).*

available not only have the potential to address this problem, but create archaeological visualizations that are more accessible, substantive, and dynamic than their non-digital counterparts.

## 2.2 Viewing Vector Graphics on the Web

Most of the vector graphics used in archaeology are created with some form of proprietary software that was either not designed with the Web in mind, or has features that cannot be handled by particular browsers. In general, files must be saved in special formats and a separate viewer or browser plug-in may be required to interpret these files on the Web. Often, these viewers have only limited functionality compared to the original software and only work with particular browsers or operating systems. While the software needed to create a vector image may be expensive, the viewers and plug-ins are usually available for download without cost to the end user.

For files created in Illustrator, there is no particular viewer needed. The Web format for Illustrator files is GIF, and it is a standard that works in virtually any graphical Web browser. Although Illustrator has the ability to divide drawings into layers, the GIF format merely rasterizes the vector document, so there is no way to access its functionality. For files created in AutoCAD, the main viewer currently available is AutoDesk DWF Viewer, which includes (like its predecessors, Volo View Express and Express Viewer) the functions of pan and zoom and the ability to control layers (Figure 5). AutoDesk DWF Viewer is a new product, which now allows the viewing of three-dimensional (3D) DWF files, as well as object properties. AutoDesk DWF Viewer is proprietary, and users are therefore limited to the compatibility choices made by AutoDesk. While AutoDesk DWF Viewer is freely available, it is only compatible with Microsoft Windows XP or 2000, and Microsoft Internet Explorer 6 (Anon. 2006a). Much like AutoCAD itself, which is only available for the Microsoft Windows operating system, this seems unacceptably exclusive, not only for those who choose to use the Apple, Linux, or any other operating system, but also for those who prefer to use Opera, Firefox, or any other Web browser. There are third-party options available for purchase that allow AutoCAD files to be shared over the Web, but most are expensive and also only available for the Windows operating system. It is unrealistic to expect a significant number of users to make a software purchase solely to view archaeological AutoCAD drawings on the Web, and once again the solution remains elusive.

Unlike programs such as AutoDesk DWF Viewer, which are designed primarily for creating shared work environments, Macromedia's Flash (now owned by Adobe) has long been established as the primary tool for creating vector graphics for the Web. Flash files are viewed with Macromedia's corresponding browser plug-in called Flash Player. This plug-in is available for all major Web browsers and operating systems, and is already installed in virtually all Web-capable computers (Laaker 2002:14). Flash is a mature technology with many features that appeal to archaeologists wishing to present vector graphics on the Web. Most people already use Flash, it creates small vector files, and the Flash Player plug-in loads images quickly and efficiently. Flash creates vector images that include

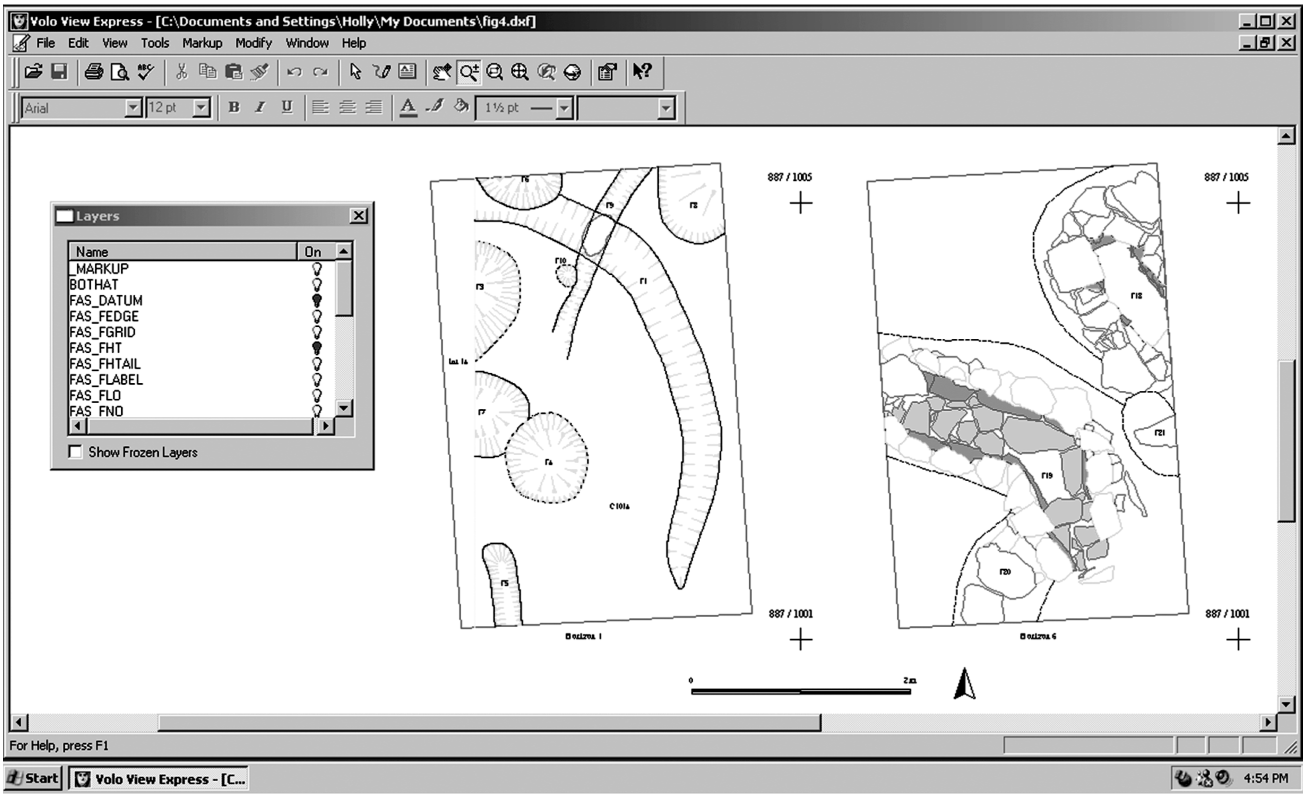


*Figure 5. Example of a CAD drawing displayed in AutoDesk's proprietary Web viewer Volo View Express. Excavations in advance of an oil tank (Sector 2, Intervention 26) plan view of horizons one and six from the Tarbat Discovery Programme (Carver 1998), The Settlements at Tarbat, figure 4).*

typographic tools, animations, and dynamically generated content (Watt 2002:504). More specifically, Flash allows Adobe Illustrator files and AutoCAD files in .dxf format to be imported and manipulated. Flash would seem to be an ideal solution for presenting vector graphics on the Web, and yet this technology is not widely used by archaeologists.

Looking more deeply at how Flash actually works, it is easy to see why. Flash only supports the importation of AutoCAD files in .dxf format (Release 10) and so is bound to .dxf's limitations. This includes lack of support for standard system fonts, and the inability to use fills. Flash only supports two-dimensional (2D) .dxf files, so any 3D AutoCAD files cannot be imported. In addition, Flash doesn't support scaling of .dxf files, which should be fundamental to any vector image (Anon. 2006b).

## 3 Scalable Vector Graphics (SVG)

### 3.1 The Development of SVG

Scalable Vector Graphics (SVG) is something entirely new. Far from being an inexpensive substitute for Flash, SVG belongs to the growing family of open-source World Wide Web Consortium (W3C) technologies that are actively shaping the way information is presented on the Web, and it is best understood when viewed within that context (Winter and Neumann 2003). To understand SVG and how it fits with other W3C recommendations, it is important to explore how SVG developed. When Tim Berners-Lee founded the W3C in 1994, his purpose was to help stabilize the implementations of his expanding HyperText Markup Language (HTML) used to create the World Wide Web (Berners-Lee 2000:143; Cagle 2002:8). By bringing together developers from disparate and often competing vendors, the W3C guides the intentionally unwieldy Web forward with some sense of cohesion. The W3C tries to be a voice of reason, and that voice has become increasingly authoritative (Castro 2003:16).

Through the W3C "official recommendations" system, commercial and non-commercial developers are brought together to give input into a standard, which creates a sense of where a technology is going and how to concentrate their own design efforts (Berners-Lee 2000:129; Watt 2002:xxiii). SVG was developed to address the lack of an alternative to the raster images that dominate the Web (Watt 2002:xvii). XML is an entirely text-based language that creates comparatively small file sizes, so calling on its extensible nature, developers looked for a way to develop a graphics format that could be text-based as well. For example, SVG is written in XML, so the SVG markup for a circle is simply <circle> </circle> which makes it perfectly understandable to humans. A circle, marked up in SVG, with a radius of 50 pixels and a black outline of three pixels looks like:

<circle r="50" style="stroke-width: 3: black: fill: none;"/ >

Because vector graphics render from text, they can be recalculated using different variables each time they are loaded (Watt 2002:xvi). By changing the radius from r=”50” to r=”100”, the size of the circle will become larger, but it will retain all of its other characteristics. This allows users to pan and zoom around an image at any magnification without loss of image quality, because the vector image just re-calculates each time. This ability is completely basic to the workings of a vector graphic, but impossible for a raster image.

The commercial sector was quick to see the potential for defining a graphic specification for XML. Adobe and Microsoft both submitted proposals to the W3C in 1998, the same year XML became an official recommendation. Adobe’s submission was called Precision Graphics Markup Language (PGML) and Microsoft’s was Vector Markup Language (VML) (Story 2000). Macromedia’s Flash specification was also released at this time in a non-XML binary format. Macromedia chose to throw their support behind Microsoft’s bid against Adobe, their traditional rival. The SVG working group took both specifications under consideration, and chose to merge the two technologies into what would become SVG. Through the working group, a wide variety of vendors were able to contribute to the process of creating the specification. These included AutoDesk, IBM, Netscape, Apple, Sun Microsystems, Xerox, Corel, Visio, Hewlett-Packard, and Quark. Despite SVG’s subsequent adoption as an official W3C recommendation, Microsoft, in typical proprietary fashion, has continued to use VML. The result of this is a browser plug-in that is only compatible with Microsoft Internet Explorer 5 for Windows (Cagle 2002:10).

The development of SVG also set historic precedents for the W3C. As stated by Chris Lilley, chair of both W3C SVG working groups, SVG was:

> a demonstration of the process that enables competing companies to come together in a vendor-neutral space and work on commonly agreed, open specifications for the benefit of the Web in general and to grow the market. SVG was the first specification to not only have a test suite, but also publish the results of testing on named implementations. During the Candidate Recommendation phase, implementers and content creators gave a large amount of valuable feedback that helped to improve the clarity and technical accuracy of the specification. As a result, compared to other specifications at an equivalent level of maturity, SVG was extremely well implemented by the time it became a W3C Recommendation on September 4, 2001 (Watt et al. 2003:xxiv).

### 3.2 The Features of SVG

It is important to remember that some of its most compelling features, such as platform independence and interoperability with other XML technologies, are not exclusive to SVG, they are just part of membership in the XML family. Because SVG is freely available, nobody “owns” it and no single software company can tailor it to their particular market. Like HTML, SVG is made up of nothing more than text, which can be viewed by developers and users alike, which makes it easy to see and share the way SVG documents are structured. For HTML, this resulted in hugely accelerated development, and this will probably be the case for SVG.

One of the primary complaints about Flash is that documents are created in the proprietary .swf file format, which is binary and therefore not meant to be read or accessed by people (Laaker 2002:13). Users and developers can view SVG, and the contents are understandable to humans and not just computers (Jackson 2002). SVG allows selective display of elements in an image. Because vector graphics can be created in layers that can be turned on and off, those layers can be preserved in SVG, and interacted with using a scripting language like JavaScript (or more accurately, ECMAScript) (Watt et al. 2003:17). The text that makes up an SVG graphic is also available to search engines able to read XML, so if a vector graphic contains text, it is still recognizable as such even if it is embedded into an image (Watt 2002:95).

This ability to recognize the textual parts of an SVG image also allows for internationalization of its content. If the settings in a browser are set for a particular language, the SVG image will display the same graphical elements, but the textual elements that are appropriate for that language will be chosen and displayed (Watt et al. 2003:19). Browser detection and SVG will also allow for greater flexibility in the future for designing accessible websites.. SVG’s resolution independence already means users with visual limitations can scale images to a level of magnification that is comfortable. For users that require images with different color contrasts or text only, SVG should be able to serve a version of the same page to meet their needs. Rather than designing websites for an accessible lowest common denominator, developers could create one site that can be viewed (or not, in the case of audio browsers) in a variety of ways. This will be somewhat in the future, however, since accessible SVG browsers and plug-ins will have to be created first (Watt et al. 2003:510). SVG can also create graphics that are data-driven and generated dynamically from a server. This can be done in a variety of ways using existing programming languages such as PHP, PERL, ASP, or JSP, but the result is a visualization that can be created “on the fly” based on user criteria (Watt et al. 2003:695-98).

While XML is a rapidly growing specification, all Web browsers do not support it natively, but this is changing. The most popular browsers that do offer such support are Mozilla’s Firefox 1.5 and Opera 8. These are both available for a wide variety of operating systems, including the most popular: Windows, Macintosh, and Linux. Recent testing of the most recent versions of these two browsers using Windows XP and Apple OSX revealed quite uneven implementation using Firefox, while Opera was much more solid. Because SVG is non-proprietary there are several companies and organizations with an interest in furthering its development, and they have added SVG support to their products or created plug-ins for other browsers. Adobe’s SVG Viewer plug-in is the most widely used, with the broadest distribution and compatibility with all major browsers or platforms.


Figure 6. Judith Winters, Editor of Internet Archaeology, with one of the original large-format permatrace drawings from the 1975 Cricklade excavation.

## 4 Cricklade: A Practical Archaeological Application of SVG

### 4.1 The Cricklade Publication

This section will look at some of the practical problems for converting archaeological vector drawings into SVG, and how to make the drawings more interactive by using JavaScript to control layers. To demonstrate, AutoCAD drawings prepared by Guy Hopkinson for the *Internet Archaeology* publication "Excavations at Cricklade, Wiltshire, 1975" by Jeremy Haslam (2003) were used. As published, the drawings are viewed with AutoDesk's Volo View Express software. Volo View Express (like AutoDesk DWF Viewer, which has recently replaced it) is designed solely for use with Microsoft's Internet Explorer Web browser, and Microsoft's Windows operating system. Because AutoDesk only makes its products available for Windows, it is often assumed that those interested in accessing CAD drawings on the Web will be Windows (and therefore Internet Explorer) users anyway (Watt et al. 2003:925). This is shortsighted at best, and at worst perpetuates the view that reliance on a particular vendor is sufficient for the healthy development of the Web. As an open source, XML-based markup language, SVG was explored as a possible alternative to this proprietary solution.

The site of Cricklade is part of the 9th-century system of Saxon urban fortresses and is located in Wiltshire, England. The defenses show the rectilinear planning typical of this type of defensive structure, and it is in a good state of preservation. The site has been well researched by both archaeologists and historians, and in 1975 excavations were carried out in advance of housing development (Haslam 2003). In July 2003, the article "Excavations at Cricklade, Wiltshire 1975" by Jeremy Haslam was published in Issue 14 of *Internet Archaeology*. Because this article published the findings of excavations carried out in the 1970s, inclusion in *Internet Archaeology* was an experiment in "retrospective conversion" or "retrospective publication", which is the process of digitizing existing hard-copy archaeological drawings (Eiteljorg et al. 2002). The twelve ink-on-permatrace drawings from Cricklade were large format, with some over 1.5 m long (Figure 6). If published at the time, the drawings would have been scaled down for inclusion in a print publication (Hopkinson and Winters 2003). In print, this publication would most likely have appeared in a regional journal, which would be small format (less than A4) and unable to accommodate large plan and section drawings because of the costs involved in producing foldout images. (Judith Winters pers. comm., July 2003). Electronic publication, specifically using vector graphics, has allowed the presentation of these drawings in their entirety.

### 4.2 The Cricklade Drawings

Guy Hopkinson prepared the drawings for publication, and his brief was to determine a way to scan the large permatrace drawings and clean the resulting images in a raster-editing program like Adobe Photoshop. The original ink drawings were monochrome, so color was added to take advantage of the digital medium and to facilitate better interpretation. Three of the section drawings were to be digitized in AutoCAD and saved in .dwf format for use with Volo View Express (Hopkinson and Winters 2003).

As this publication was meant to be an experiment in new forms of media publication, Hopkinson spent time exploring the best ways to approach the problem. Some of the lessons learned revealed the lack of flexibility in raster images. The publication went through an approval process with both an editor and an author, and changes were requested at several stages during image production. In particular, requests for size increases of the raster images after clean-up work had already been completed required a new scan at a higher resolution, and the work would have to be repeated. In retrospect, Hopkinson felt that a better solution would have been to digitize all of the drawings into vector format from the start, making changes easy to accommodate and saving time in the long run (Hopkinson and Winters 2003).

While not part of the final project, Hopkinson also felt that use of vector drawings could be further enhanced by other functions allowed by Volo View Express. This included embedding hyperlinks into an image to attach further visual information. This could potentially create an interface for the publication of an entire visual archive. He cites the main drawback as lack of support for other operating systems besides Windows (Hopkinson and Winters 2003). All of these things, and much more, can be accomplished using SVG, in a format that is not limited to a single operating system.

### 4.3 The Case Study

**Creating the SVG Images.** The layered drawings created by Guy Hopkinson for *Internet Archaeology* were prepared specifically for electronic publication. Having already gone through the editorial process, they were ideal candidates for demonstrating how SVG might be used to produce results that are comparable to, or better than, Volo View Express. In order to achieve this, a structure in the form of a website was created to house the drawings and provide a framework for interaction to occur.

The website created for this purpose was based on W3C design standards and validation, so that SVG elements would work in partnership with other open-source W3C technologies. It was created in XHTML 1.0, with all formatting controlled using external Level Two Cascading Style Sheets. The site was also designed to conform to the W3C Web Content Accessibility Guidelines 1.0 at the AA level and the index page includes the relevant Dublin Core metadata. Some concessions were made, owing to browser limitations and lack of access to specialized technologies, but they are minimal.

The tool used to create the SVG images was Adobe Illustrator 10.0.3 for the Apple OSX operating system, but other programs are available to create SVG files. Illustrator can be used to create original artwork that is saved in SVG or SVGZ (SVG with file compression) format. Documents created in a wide variety of outside formats can be opened, manipulated, and saved in SVG, as well. Layers created in drawing programs like AutoCAD can be preserved when brought into Illustrator, and subsequently saved into the SVG file. Layers are designated using the “group” element, and the original layer name becomes the group identifier. So a layer in Illustrator converted to SVG creates markup tags that look like:

```
<g id=”layer_name”></g>
```

It is important to note, however, that files saved as SVG from Illustrator do not retain SVG “primitives.” This is to say, while a circle may be created as a circle in AutoCAD or Illustrator, when it is saved in SVG, rather than using the <circle></circle> tags, it becomes a series of parameters in a path (Anon. 2005a). So a layer in an SVG file will consist of a group identity and series of paths holding the specific information for each set of points, lines or polygons for that layer (Figure 7). SVG code can be edited and optimized in many ways after it has been saved from Illustrator, so path tags could be replaced with SVG primitives if necessary.

The group tag allows human readers to recognize layers, so these can be easily identified and manipulated. The group designation can be used to work with any feature within a layer as well. The SVG code can be edited at any time to combine layers, isolate elements within layers or change layer order. SVG groups are read from the bottom of the file to the top, so the layer that is meant to display uppermost will be the last layer in the SVG file. Informal testing revealed that layer order was very important to those users of the Cricklade SVG website with archaeological experience, and that the uppermost layer to be displayed should always be the excavation layer. This is easily achieved by moving the excavation layer group to the end of the appropriate code in the SVG file.

Use of the group tag is not limited to layers. It can make any part of an image available for a wide variety of manipulations either within the SVG file, or by using an external

```
<g id="building">
    <path fill="none" stroke="#000000" stroke-width="0.1764" stroke-linecap="round" stroke-linejoin="round" d="M118.233,222.684
        l9.701-0.658l0.044,0.46l3.314-0.241l0.198,3.576l-3.007,0.264l0.022,0.482l-10.008,0.658L118.233,222.684z"/>
    <path fill="none" stroke="#000000" stroke-width="0.1764" stroke-linecap="round" stroke-linejoin="round" d="M161.275,54.491
        l2.409,0.476l-0.289,1.846l-2.553-0.447L161.275,54.491z"/>
    <path d="M118.233,222.684l9.701-0.658l0.044,0.46l3.314-0.241l0.198,3.576l-3.007,0.264l0.022,0.482l-10.008,0.658
        L118.233,222.684z M161.275,54.491l2.409,0.476l-0.289,1.846l-2.553-0.447L161.275,54.491z"/>
</g>
```

*Figure 7. Example of the code for a very simple layer in SVG, as exported from Adobe Illustrator.*

```
<g>
    <path fill="none" d="M59.223,54.336v-19h-35v19H59.223z"/>
    <text transform="matrix(1 0 0 1 24.2227 46.2109)" enable-background="new">
    <tspan x="0" y="0" fill="#FFFFFF" font-family="'Arial-BoldMT'" font-size="12" letter-spacing="4.8">
    <a xlink:href="http://intarch.ac.uk/journal/issue14/haslam_index.html">GO</a></tspan></text>
</g>
```

*Figure 8. Example of a simple interaction with an SVG group.*

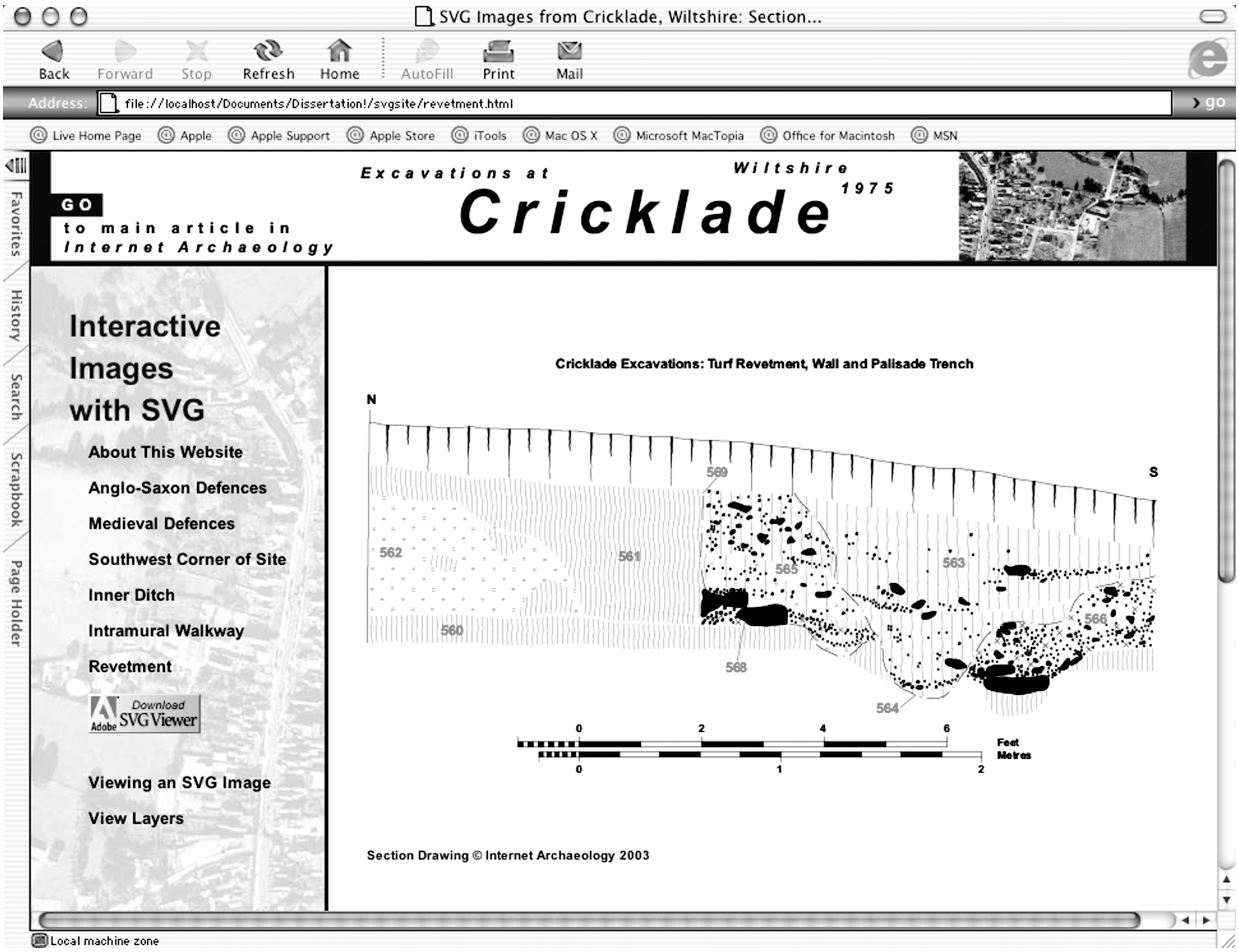


*Figure 9. The design of the Cricklade website showing the SVG image along the top. The Turf Revetment and Palisade Trench SVG image is in the content window. The positioning of the design is controlled using CSS, so the SVG and XHTML elements can be placed together seamlessly.*

scripting language like JavaScript. For example, a particular polygon can be isolated for use in an animation, and the animation will run automatically when the image loads because the code to control this behavior is part of the SVG file. In contrast, an external action written in JavaScript can be used to bring up an alert message when clicking on part of the SVG image. In either case, the reason is the same, because SVG is text that is human readable, information can be isolated and made interactive in a wide variety of ways.

**Creating the Website.** In addition to the Cricklade drawings, an SVG image was incorporated into the structure of the website itself to demonstrate its potential as a design element. This is the image that runs across the top of the website, showing the title (see Figures 9, 11, 12). To show how raster images can be incorporated into SVG, a small black and white detail from an aerial photograph of Cricklade is embedded in the right side of the image.

The left side of this image includes a hyperlink to the main Cricklade article in *Internet Archaeology*, and the link is embedded into the SVG image itself. This is done by isolating the word GO into a group and housing the SVG "xlink:href" attribute within it (Figure 8).

The formatting of the website is controlled externally using CSS, so the SVG image is part of the XHTML layout for the site (Figure 9). Because CSS allows elements to be placed on a page with pixel precision, the XHTML formatting fits seamlessly around the SVG. It would be imprudent to use SVG as part of the basic design for a website until all

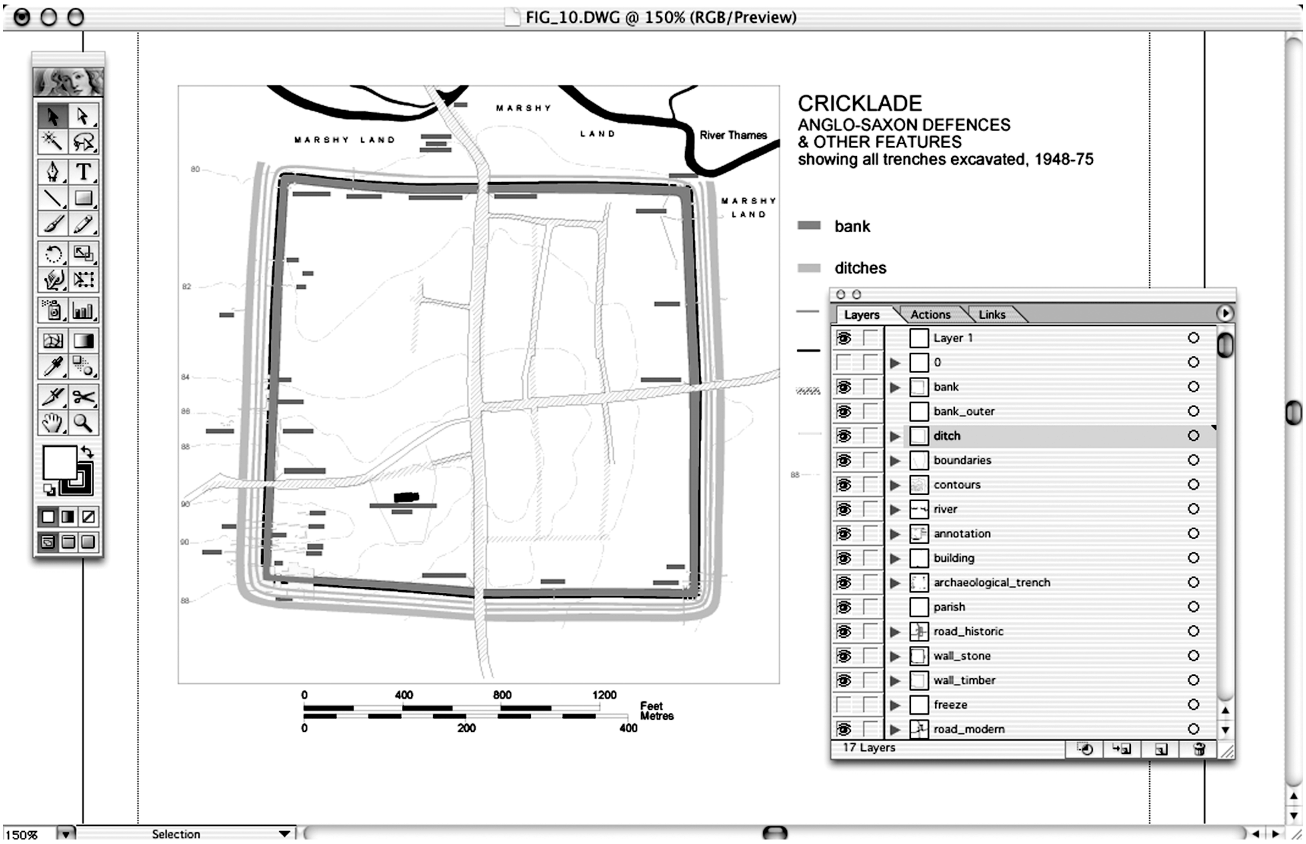


*Figure 10. Example of a .dwg file in Illustrator, showing the control palette and the layers palette for the drawing of Anglo-Saxon Defenses. The layer names are preserved from the AutoCAD file and can be manipulated in Illustrator.*

browsers implement SVG natively. More and more basic elements and functions could be handled this way in the future, and ultimately SVG could become a Web-authoring tool (Watt 2002:395). As it now stands, requiring a user to download an SVG plug-in simply to view the title bar of a website is unrealistic, so this is primarily a demonstration.

The six plan and section drawings in the website each went through a similar process to convert them into SVG. The drawings were opened in Illustrator in their original .dwg format, and the layers were examined for problems (Figure 10). The drawings were compared in AutoCAD to determine if any changes or data loss had occurred. If the drawing was unchanged, it was saved in SVG format. Experiments using SVGZ data compression produced undesirable visual changes and browser compatibility problems, so it was not used. Once in SVG format, the files were checked again for data loss and if they were found acceptable, the process of incorporating them into the website began.

To display an SVG image in a website requires use of the “embed” tag. This tag is deprecated (no longer valid) in XHTML in favor of the object tag, but it is widely held in the SVG community (and beyond) that use of the embed tag is a necessary evil. Until current browser manufacturers and the W3C sort out their differences over support for the object tag, using embed is the only reliable way to ensure SVG images will display in most browsers (Castro 2003:295; Eisenberg 2002:321; Watt 2002:321; Watt et al. 2003:486).

**Using the Website.** Once the images were placed in XHTML using the embed tag, it was possible to view them and adjust their size and placement. Adobe SVG Viewer has built-in controls for viewing an image using pan and zoom (Figure 11). These controls can either be activated with key commands, or through a pop-up menu brought up with a right-mouse click. This is one area where Volo View Express excels over the functions of Adobe SVG Viewer. The former allows the user to control the level of zoom incrementally by panning the cursor up and down the image, whereas the latter only allows four preset levels of zoom. This is generally sufficient, but an improvement in this area for subsequent versions of Adobe SVG Viewer would be welcome.

Further interaction was introduced by giving control of the layers in the drawings to the user. JavaScript was added to the XHTML to turn the visibility of each layer on or off through an external control panel (Figure 12). Layers are made visible by clicking the corresponding checkboxes in any order or combination, and they can be used when looking at any part of the drawing or at any level of zoom. This method offers far greater flexibility of presentation than Volo View Express. While the layer names used in CAD files may make sense for the creator of the drawings, they may not be understandable to a non-expert viewer, especially if they rely upon a coded system like the CSA Layer Naming Convention (Eiteljorg 2002). In order to change the way layer names appear in Volo View Express, it is necessary to return to the original CAD file and rename them.

With SVG, layer names can be left intact and changed when brought into Illustrator, or later in the code of the

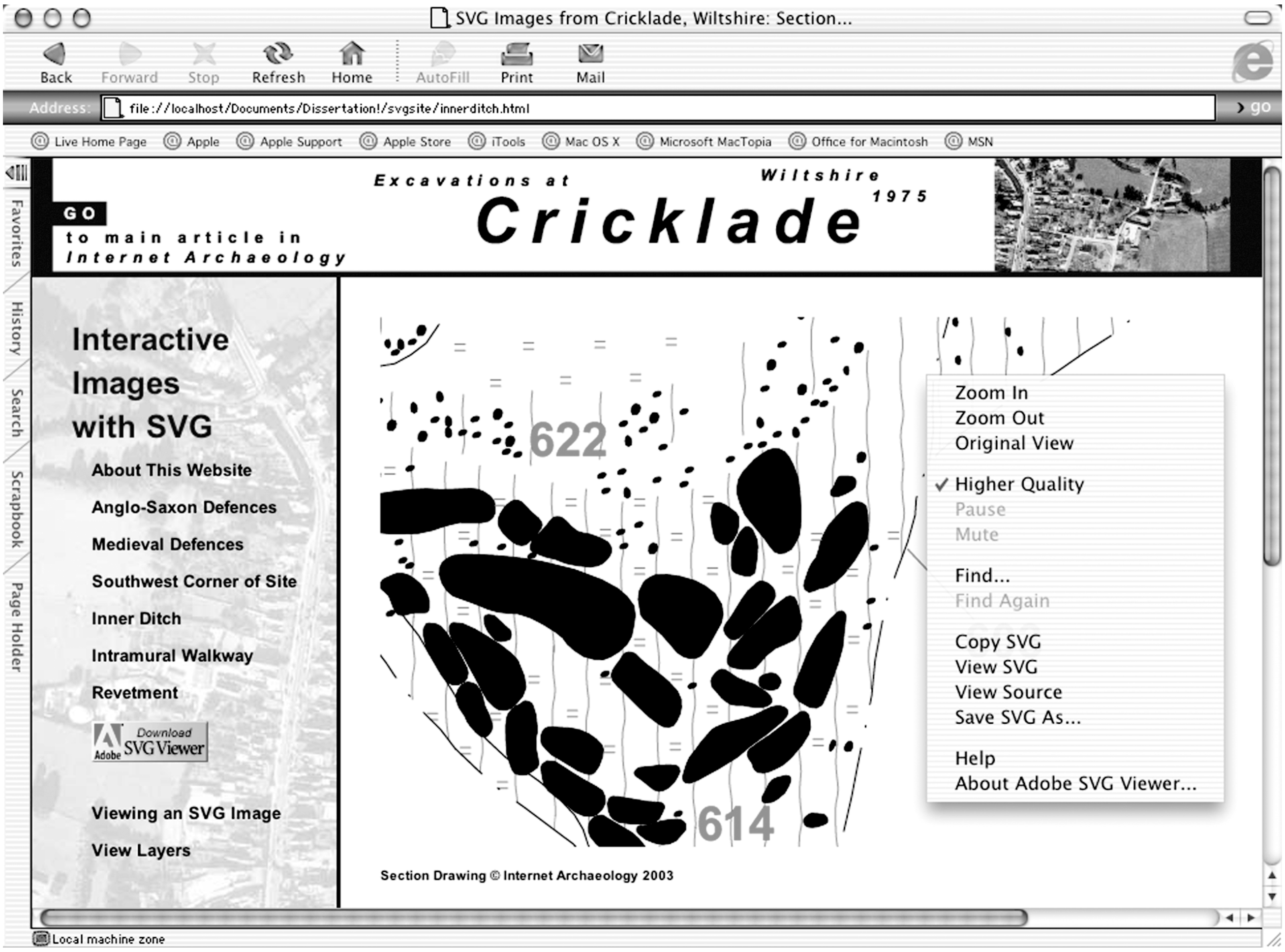


*Figure 11. Example of SVG image interaction using the built in Adobe SVG Viewer interface.*

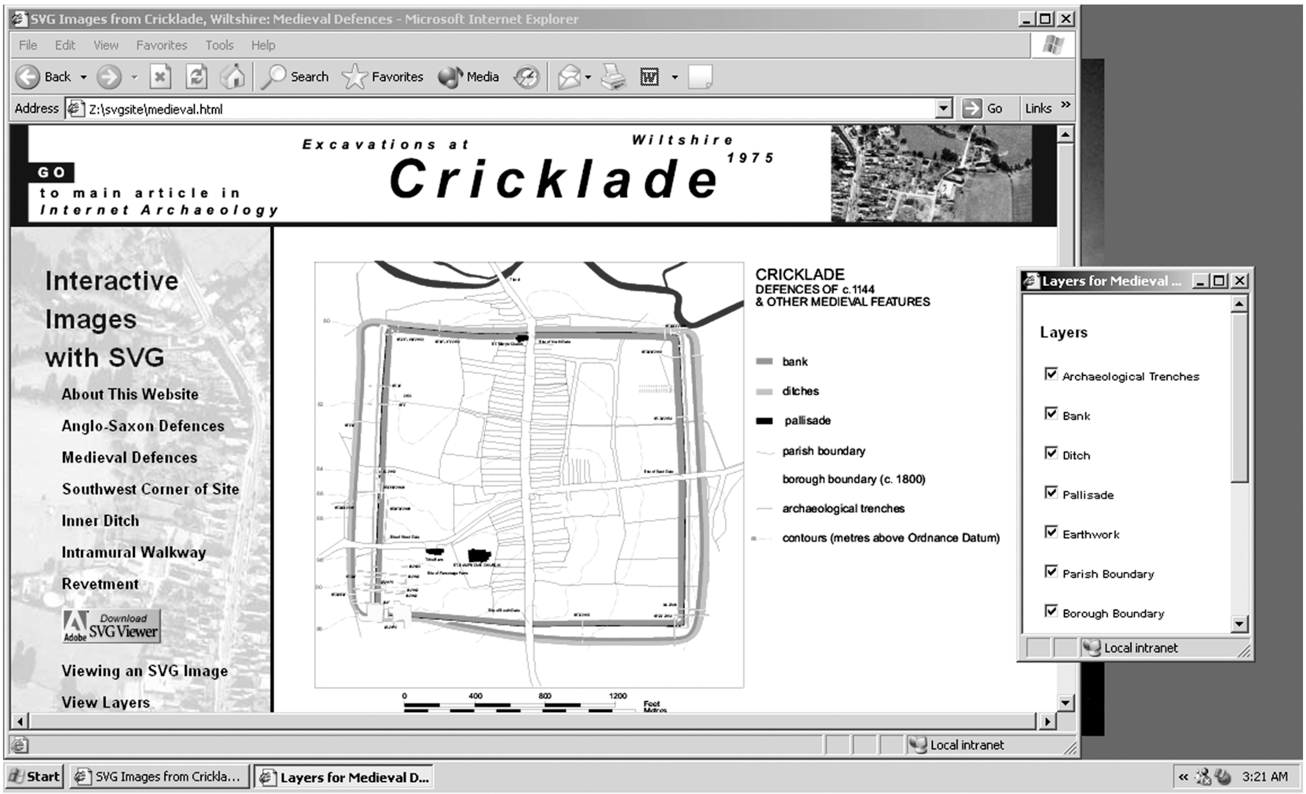


*Figure 12. Example of an SVG drawing in the Cricklade SVG website, showing the floating layer control panel on the right.*

SVG file itself. This step is not strictly necessary either, as long as the designer understands what the layers are. The names can remain completely oblique, as long as the text that corresponds to the checkbox in the layer control window is understandable to outside users. The order of the checkboxes can also be changed at any point in the process, without reference to the order of the JavaScript. The order of checkboxes does not need to correspond with the group order (and therefore layer display order) in the SVG file, either. The flexibility for organizing and presenting layer information is only limited by the functions of XHTML and JavaScript, and allows far greater control than Volo View Express.

While SVG opens up control of vector images to those using a much wider variety of Web browsers and operating systems, one current drawback is the uneven support for the use of JavaScript with SVG images. Despite various claims, testing of layer control for both the Apple and Microsoft Windows operating systems using Internet Explorer, Netscape, Firefox, and Opera returned disappointing results. Only Internet Explorer for Windows reliably implemented the layer control, so it was important to make sure the drawings could be understood without this function, and that users were made aware of the problem. While this is discouraging from a Web standards point of view, it is no worse than Volo View Express, which will probably never move beyond this same proprietary limitation.

Because images created in Illustrator can come from a wide variety of programs and file formats, SVG allows greater aesthetic flexibility, which can help to promote more effective communication. For the Cricklade drawings, Guy Hopkinson created two versions of the detailed plan for the excavation carried out in the southwest area of the site. The original .dwg file from AutoCAD was used to produce the .dwf file for use with Volo View Express. The drawing was meant to convey the differences between the excavated and conjectural information about the site, but because AutoCAD can only render lines and solid fills, the result was not as clear as one might wish (Figure 13).

To remedy this, Hopkinson took the .dwg file and brought it into Illustrator to enhance the drawing and make it easier to understand. The image uses transparency rather than hatching in various colors to show the differences between the excavated and conjectural parts of the drawing (Figure 14). The result is very effective, but because of the limitations of Volo View Express, the image could only be rasterized and used as a static .gif in the final publication of the article for *Internet Archaeology*.

SVG does not have these limitations, and the Illustrator version of the drawing was used in the Cricklade website. It effectively conveys the usefulness of layer control for vector images in archaeology, and the Illustrator image shows the differences between the excavated, observed, and conjectural areas far better than the AutoCAD version.

**Evaluation:** As an exercise, the Cricklade website shows some of the features of SVG and how it can be used as an effective communication tool for archaeological visualization. In particular, it shows the ease with which basic SVG functions can be introduced into virtually any website, when using a vector design tool like Adobe Illustrator. Illustrator can be used as a stand-alone program for creating vector graphics and saving them into SVG format, or as a pass-through program for taking vector drawing files made in other programs and converting them to SVG. As shown by the drawing of the southwest corner of the Cricklade site, the best results are often produced when used in combination. Many of the limitations of visual presentation in programs like AutoCAD can be addressed in Illustrator before conversion to SVG.

The site also demonstrates how SVG can be taken further, either with interaction built into the image itself or with other forms of technology like JavaScript. Virtually every element of an SVG image can be identified in the document's code and understood to be a separate object, which can be manipulated. The JavaScript interface used to create the layer control in the Cricklade website illustrates this. Each drawing has a unique set of layers that can be controlled by the user, allowing more complete access to drawings as originally digitized, and thereby clearer understanding of the archaeology. At the same time, the versatility of SVG when used with JavaScript and XHTML makes it far easier to adapt archaeological information for a broader Web audience, without having to make changes to original data.

This practical demonstration has only scratched the surface of what SVG can do, and how it might be applied to archaeological visualization. The *Internet Archaeology* publication "Excavations at Cricklade, Wiltshire, 1975" shows one way to present archaeological vector drawings on the Web, and provides a good foundation to contrast the functionality of SVG and JavaScript with Volo View Express. On the surface these may seem to have roughly compatible features, but they are fundamentally different. The power and flexibility of SVG, especially when used in combination with other elements like JavaScript, has the potential to outpace anything a proprietary program such as Volo View Express can achieve.

## 5 Conclusion

The development of XML has opened up many possibilities for working with archaeological information, both on and off the Web. While print designers have used raster and vector technology equally and in combination for years, Web designers have been without a comprehensive vector graphics solution. Forced to work around the particular issues associated with vector images for so long, SVG has remained somewhat below the radar of many, but this will almost certainly change. In addition to possessing the useful qualities found generally in vector technology, SVG opens up a host of additional possibilities. When combined with other forms of technology like databases and JavaScript, it is capable of interactivity that is both simple and complex. SVG can be used to create high-quality graphical interfaces for the visualization and analysis of archaeological information, and make primary data easier to understand. As archaeologists seek non-proprietary ways to preserve their

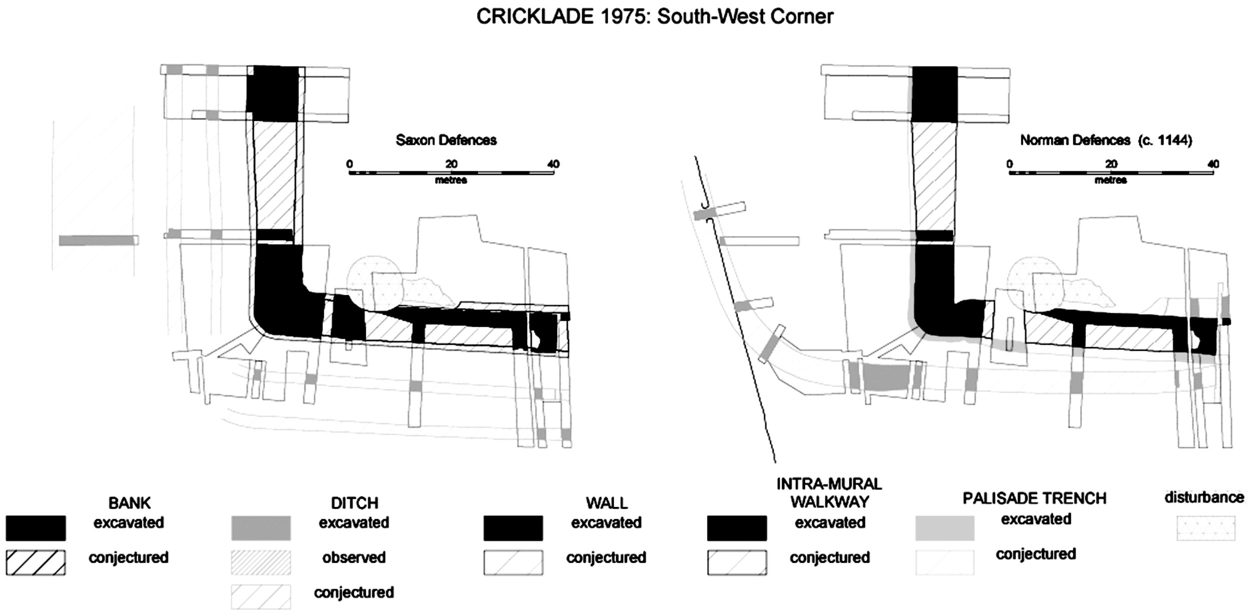


*Figure 13. The original *.dwg drawing of the southwest corner of the Cricklade site, where most of the excavation was carried out. This image shows the line and solid fills (when the image is rendered in color), that must be used to differentiate information in AutoCAD.*

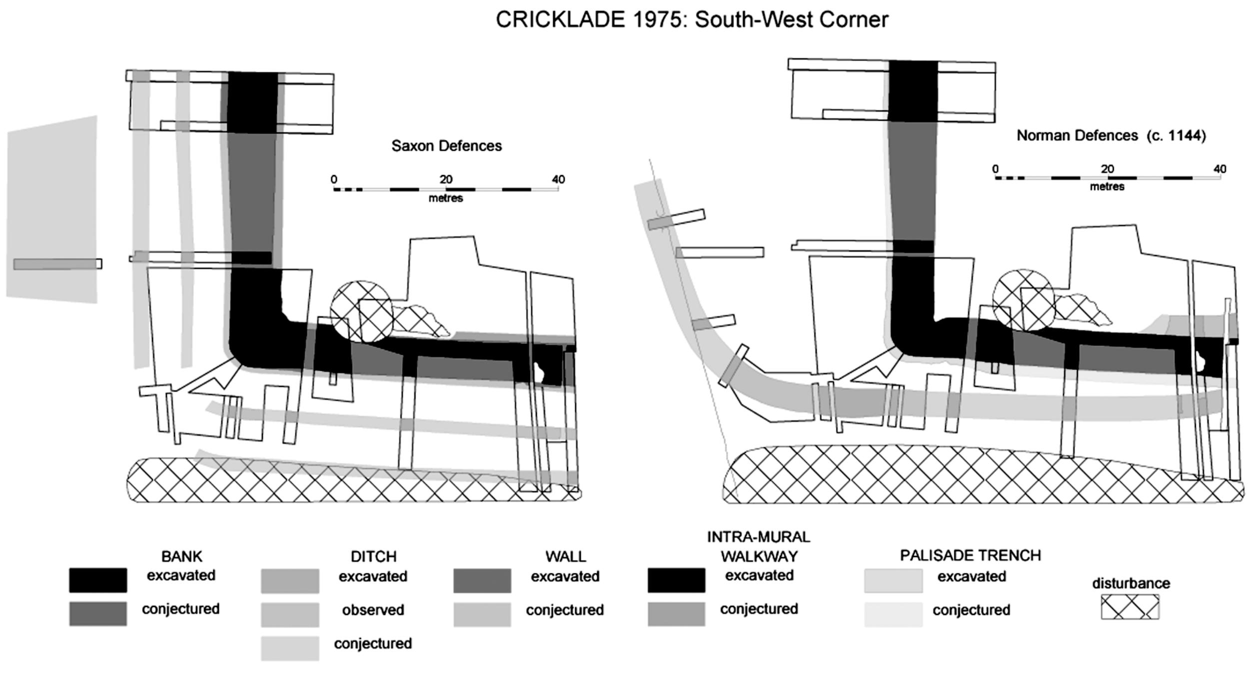


*Figure 14. The same drawing from Figure 13, showing the effective use of transparent color. The image shows the differences between the parts of the drawing that were excavated and the parts that are conjectural.*

data, either for further active use or in the form of an archive for future interpretation, forward migration of data is an ever important issue (Anon. 2005b; Zeldman 2003). Using standards-based W3C technology such as SVG will also allow archaeological information to be brought to the Web in ways that will produce data that can be better maintained, easily migrated, and can serve multiple purposes.

## References Cited

Anon. 2005a. Creating SVG with Adobe Illustrator CS2. Adobe SVG Zone. Updated January 2005, http://www.adobe.com/svg/pdfs/illustrator_svg.pdf (Accessed 15 January 2006).

Anon. 2005b. What is the ADS, and what do we do. Archaeology Data Service Website. Updated May 2005, http://ads.ahds.ac.uk/project/general.html (Accessed 20 February 2006).

Anon. 2006a. AutoDesk Website: AutoDesk DWF. Updated 2006, http://usa.autodesk.com/adsk/servlet/index?siteID=123112&id=6101753 (Accessed 15 March 2006).

Anon. 2006b. Adobe Website: SVG Zone. Updated 2006, http://www.adobe.com/svg/overview/svg.html (Accessed 22 February 2006).

Berners-Lee, Tim. 2000. *Weaving the Web.* London: Texere.

Brewer, Jeannette and Kilbride, William G. 2006. HEIRNET User Survey 2005 Report. Updated February 2006, http://www.britarch.ac.uk/HEIRNET/survey/section1.htm (Accessed 15 March 2006).

Cagle, Kurt. 2002. *SVG Programming: The Graphical Web.* Berkeley: Apress.

Carver, Martin O. H. 1998. Bulletin of the Tarbat Discovery Programme 4. Updated October 2003, http://www.york.ac.uk/depts/arch/staff/sites/tarbat/bulletins/bulletin4/contents.html (Accessed 20 March 2006).

Castro, Elizabeth. 2003. *HTML for the World Wide Web.* Berkeley: Peachpit Press.

Clarke, Amanda, Fulford, Michael, and Rains, Michael. 2003. Nothing to hide—online database pubication and the Silchester Town Life Project. In, *Proceedings of the 29th CAA conference held at Heraklion Crete, Greece, April 2002.* M. Doerr, and A. Sarris, eds., pp. 401-404. Hellenic Ministry of Culture.

Eisenberg, David. 2002. *SVG Essentials.* Sebastepol: O'Reilly and Associates, Inc.

Eiteljorg II, Harrison. 1989. Computer-Assisted Drafting and Design: New Techniques for Old Problems. Updated April 1996, http://csanet.org/inftech/cadbklt.html (Accessed 23 January 2006).

Eiteljorg II, Harrison. 2002. CSA Layer Naming Convention. Updated December 2002, http://csanet.org/inftech/csalnc.html (Accessed 10 February 2006).

Eiteljorg II, Harrison, Fernie, Kate, Huggett, Jeremy, and Robinson, Damian. 2002. CAD: A Guide to Good Practice. Updated 2002, http://ads.ahds.ac.uk/project/goodguides/cad (Accessed 13 February 2006).

Haslam, Jeremy. 2003. Excavations at Cricklade, Wiltshire 1975. Updated June 2003, http://intarch.ac.uk/journal/issue14/haslam_index.html (Accessed 10 March 2006).

Hopkinson, Guy, and Winters, Judith. 2003. Problems with Permatrace: a note on digital image publication. Updated February 2004, http://intarch.ac.uk/journal/issue14/hopkinson_index.html (Accessed 9 March 2006).

Jackson, Dean. 2002. SVG On the Rise. Updated June 2002, http://www.oreillynet.com/pub/a/javascript/2002/06/06/svg_future.html (Accessed 15 February 2006).

Jones, Siân, MacSween, Ann, Jeffrey, Stuart, Morris, Richard, and Heyworth, Mike. 2001. From The Ground Up: The Publication of Archaeological Projects: a user needs survey. Updated May 2001, http://www.britarch.ac.uk/pubs/puns (Accessed 18 March 2006).

Jones, Susan C. 1 997. Raster And Vector Images—An Important Distinction. Updated May 1997, http://csanet.org/newsletter/spring97/nl059707.html (Accessed 2 February 2006).

Laaker, Micah. 2002. *Sams Teach Yourself SVG in 24 Hours.* Indianapolis: Sams Publishing.

Livingood, Patrick. 1996. Electronic Publication in Archaeology. Updated 1996, http://www-personal.umich.edu/~patrickl/sthesis_2.htm (Accessed 25 March 2006).

Middleton, Mike. 1998. Opinion. Association of Archaeological Illustrators and Surveyors Newsletter. Updated October 1998, http://www.aais.org.uk/html/newsletter.html (Accessed 24 March 2006).

Miller, Paul, and Richards, Julian. 1995. The good, the bad, and the downright misleading: archaeological adoption of computer visualization. In, *Computer Applications and Quantitative Methods in Archaeology 1994.* J. Huggett and N. Ryan, eds., pp. 19-22. Oxford: TEMPVS REPARATVM.

Ryan, Nick. 2002. Documenting and Validating Virtual Archaeology. Updated April 2002, http://www.cs.kent.ac.uk/people/staff/nsr/cvro (Accessed 9 March 2006).

Story, Derrick. 2000. Extensible Graphics With SVG. Updated April 2000, http://www.oreillynet.com/pub/a/network/2000/04/28/feature/svg.html (Accessed 23 March 2006).

Terras, Melissa M. 1999. A Virtual Tomb for Kelvingrove: Virtual Reality, Archaeology and Education. Updated November 1999, http://intarch.ac.uk/journal/issue7/terras_index.html (Accessed 18 February 2006).

Watt, Andrew H. 2002. *Designing SVG Web Graphics.* Indianapolis: New Riders Publishing.

Watt, Andrew H., Lilley, Chris, Ayers, Daniel J., George, Randy, Wenz, Christian, Hauser, Tobias, Lindsey, Kevin, and Gustavsson, Niklas. 2003. *SVG Unleashed.* Indianapolis: Sams Publishing.

Wheatley, David., and Gillings, Mark. 2002. *Spatial Technology and Archaeology: The Archeological Applications of GIS.* London: Taylor and Francis.

Winter, Andréas M., and Neumann, Andreas. 2003. Vector-based Web Cartography: Enabler SVG. Updated March 2003, http://www.carto.net/papers/svg/index_e.shtml (Accessed 20 February 2006).

Wrathmell, S., and Nicholson, A. (eds.) 1990. *Dalton Parlours: Iron Age Settlement and Roman Villa.* West Yorkshire Archaeology Service.

Zeldman, Jeffrey. 2003. *Designing With Web Standards.* Indianapolis: New Riders Publishing.