

**to the National
Measurement System
Policy Unit, Department
of Trade & Industry**

**GUIDANCE AND TOOLS FOR
INTERACTIVE WEB PAGES**

**By Alister Cooper,
NPL**

September 1999

Software Support for Metrology

Guidance and Tools for Interactive Web Pages

Alister Cooper

Centre for Information Systems Engineering,
National Physical Laboratory.

Abstract: The World Wide Web has become one of the foremost means of communication and technology transfer available. The rapidly growing ability to provide sophisticated interactive content makes it even more valuable. This guide reviews the technologies presently available, providing a resource for anyone wishing to develop interactive web pages, and briefly looks ahead to what may be just over the horizon.

© Crown Copyright 1999
Reproduced by permission of the controller of HMSO

ISSN 1361-407X

Extracts from this report may be reproduced provided the source is acknowledged and the extract is not taken out of context.

Authorised by Dr Dave Rayner,
Head of the Centre for Information Systems Engineering
National Physical Laboratory, Queens Road, Teddington, Middlesex, TW11 0LW

Contents

1. INTRODUCTION.....	1
1.1. SSfM Web Site Project Description	1
1.2. Scope of the Document and Links to other SSfM Projects	1
1.3. What are Interactive Web Pages?	2
1.4. Servers and Clients	2
1.5. Browsers	3
2. INTERACTIVE WEB PAGES - THE TECHNOLOGY	5
2.1. Hypertext - HTML	5
2.1.1. The HTML language.....	5
2.1.2. Hypertext and Hyperlinks	6
2.1.3. Site Navigation.....	8
2.1.4. Mathematical Symbols on Web Pages	8
2.1.5. Good Practice	9
2.2. Multimedia	9
2.2.1. Images	9
2.2.2. Animation	10
2.2.3. Sound	11
2.2.4. Movies and Video	12
2.2.5. Good Practice	13
2.3. Client Side Scripting	14
2.3.1. JavaScript.....	14
2.3.2. VBScript - Visual Basic Script.....	16
2.3.3. Good practice	16
2.4. Server Side Scripting	17
2.4.1. The Common Gateway Interface - CGI	17
2.4.2. Server Side JavaScript.....	18
2.4.3. Good Practice	18
2.5. Java	18
2.6. VRML	21
2.7. Dynamic HTML	22
2.7.1. The Document Object Model (DOM).....	22
2.7.2. Cascading Style Sheets (CSS)	22
2.8. ActiveX.....	25
3. INTERACTIVE WEB PAGES - APPLICATIONS.....	26
3.1. Data Transformations	26
3.2. Charts and Graphs.....	26
3.3. Interactive Models.....	27
3.4. Interactive Control and Data Acquisition Systems	28

3.5. Database Queries.....	28
3.6. Miscellaneous Web Page Enhancements	28
3.6.1. Image galleries.....	28
3.6.2. Image Maps.....	28
4. TOOLS FOR BUILDING INTERACTIVE WEB PAGES.....	29
4.1. Hypertext - HTML	29
4.2. Multimedia.....	30
4.2.1. Images and Animation.....	30
4.2.2. Sound	31
4.2.3. Movies	31
4.3. Client Side Scripting	31
4.4. Server Side Scripting	32
4.5. Java	32
4.6. VRML.....	32
4.7. DHTML	33
4.8. ActiveX.....	33
5. THE FUTURE	34
6. CONCLUSIONS	36
APPENDICES	
Appendix A - Glossary of Terms	A - i
Appendix B - HTML Good Practice Guidance	B - i
Appendix C - Useful Resources and Bibliography.....	C - i

1. INTRODUCTION

This report has been produced as a deliverable of the DTI funded National Measurement System programme on Software Support for Metrology (SSfM) which runs from April 1998 until March 2001. It is a deliverable of the SSfM Web Site project.

1.1. SSfM Web Site Project Description

This project consists of three elements relating to the World Wide Web. The first concerns the dissemination of information about the SSfM programme and in particular technical reports that arise from it. The second and third elements address the possibility of extending the use of the web to provide interactive pages related to metrology, for example through the manipulation and visualisation of mathematical models.

Current web-based technology has now advanced to the point where interactive manipulation of complex mathematical models is now possible, e.g. using Java. This means that it is possible for external users to experiment with models such as those developed using visual modelling techniques (see the SSfM web pages for the project on Visual Modelling and Data Visualisation <http://www.npl.co.uk/ssfm/model/visual/index.html>). This has a number of benefits. Firstly, interactivity allows a much clearer understanding of the way a simulation or physical process may work, resulting in distinct commercial advantages as well as educational benefits. Additionally, there are possibilities of allowing various levels of access to a mathematical model. At the lowest level, a simplified data set could be used to demonstrate the types of input parameters, etc., required and their effects, whilst at the highest levels users could submit data sets relevant to their work and obtain practical results.

1.2. Scope of the Document and Links to other SSfM Projects

This document is designed to present the following information and guidance:

- Review of web based interactive technologies;
- Suggested good practice in the development of web pages;
- Review of the use of the technologies relating to use on NMS programme web sites;
- Review of the range of tools for developing interactive web pages;
- References to detailed materials and documents relating to each of the points above.

It is not within the scope of this document to present each of these topics in detail, as each could easily be a very large document of its own. It does mean to present enough detail for a user to understand the foundations of each topic and to reference a range of other documents, both within and external to the SSfM programme, that will provide a deeper understanding of the subject.

In addition this document is to provide generic guidance and will therefore concentrate on non-proprietary cross-platform interactive technologies where possible. The NMS programme uses nearly every conceivable platform and so platform independent solutions should be encouraged.

The following SSfM projects have or are expected to have a direct relationship to the content of this document:

- Visual Modelling and Data Visualisation
- Guidance on Developing Software for Metrology
- Web site - Work packages:
 - ▶ WP1: Web site development (<http://www.npl.co.uk/ssfm/index.html>);
 - ▶ WP3: Examples of interactive web pages
(http://www.npl.co.uk/ssfm/tt/ssfm_interactive_examples/index.html).

1.3. What are Interactive Web Pages?

The World Wide Web (WWW) has developed quickly into one of the most powerful tools for distributing information to a wide audience. The supporting technology has developed equally quickly, now offering a range of sophisticated techniques for allowing the user to interact with the information itself in real time, significantly enhancing the quality of information transfer, both in terms of understanding and content and in terms of attracting the audience.

A basic web page can offer a simple static document with text and maybe images to the user. However, as a computer is being used to display the information, much more than this can be achieved - programs can be run through the same interface. This allows a user to interact with the a web page as they would with any other visual program, i.e. the web page can be made to respond to actions taken by the user or other source, enhancing the page and offering content not available to “static” documents.

Interactive Web pages are then: *Web pages that can dynamically respond to external events.*

A web page can have a number of levels of interactivity, defined here as the ability to respond to user input beyond the basic functionality provided by a browser. Four levels may be classified as follows:

1. Non-interactive. The web page displays a fixed set of text and images and contains no mechanism for response to the user. This level is rarely encountered, as most web pages have some hyperlinks and implicit means of navigation.
2. Minimal interactivity. The web page contains fixed text and images with hypertext links to other web pages or file downloads.
3. Medium interactivity. The web page offers a significant level of choice by, or response to, the user. For instance, the web page could provide a form for user response to a questionnaire, or an image map that reacts to mouse pointer position by bringing up a set of new images or text etc.
4. High interactivity. The web page contains a mechanism for significantly reformatting its appearance or providing other sophisticated responses to user input. For instance this may include the direct manipulation of data, e.g. data base queries or data transformation, the reformatting of an image such as viewing three dimensional models, chart manipulation, or running an application, e.g. a simulation or model with real time response to user actions.

Figure 1.1 Levels of Interactivity of a Web Page

As the level of interactivity increases, as with any such scale, the complexity and care required to test and validate increases.

This guidance document focuses on levels 3 and 4.

1.4. Servers and Clients

Interactivity requires the use of a program to respond to the user’s actions. These programs may be developed and written using one or more of a number of different technologies. However all of them must run on a computer somewhere, either the “server” or the “client”. A server is a computer that provides information, files, web pages, and other services to a client, which is the combination of a user’s computer and the software it is using.

If the code supplying the interactivity on a web page is run on the user's machine, then it is called “server side”, processing. If however the code runs on the user’s machine, then it is termed as “client side”, processing. The following table lists the various strengths and weaknesses of each.

	Strengths	Weaknesses
Server Side Processing	<i>Download time</i> - large programs need not be downloaded. <i>Run time</i> - the server may be much more powerful than the client, running "processor heavy" programs faster. <i>Compatibility</i> - programs run on the machine they were written for. <i>Program security</i> - the program is kept on the server. <i>Client security</i> - the program does not run on the client machine.	<i>Server processing time(1)</i> - too many clients will cause big reductions in server speed <i>Server processing time(2)</i> - the number of clients may need to be limited to allow the program to run at all. <i>Response time</i> - communication is done over the Internet, which is very slow compared to internal communication. <i>Ease of implementation</i> - the user requires knowledge of and access to the server, for file locations, file permissions, server configuration etc.
Client Side Processing	<i>Server processing time (1)</i> - can greatly improve the server's performance when serving a number of clients. <i>Server processing time (2)</i> - removes the need to consider limiting client numbers. <i>Response time</i> - internal communication is vastly quicker than over the Internet. <i>Ease of implementation</i> - client side programs at most require an easily installable plug-in for the client browser to run.	<i>Download time</i> - the program must be downloaded - not significant with small programs. <i>Run time</i> - the client may not be as powerful as the server, increasing the run time needed. <i>Compatibility</i> - no longer a real problem with the development of cross platform technology. <i>Program security</i> - the program may be copied. <i>Client security</i> - the program runs on the local machine.

Figure 1.2 Comparative Strengths and Weaknesses of Client- and Server- side Processing

As can be seen from Figure 1.2 above, the main issues of server verses client side are processing and response times and security. Large, "processor heavy" and/or security sensitive programs are generally run on the server side, unless it is known that there is a trusted target audience using fast communications and powerful processors. Small, "processor light" and security non-sensitive programs are generally run on the client side. However the rapid development of very powerful PC processors and fast communications has meant that more and more client side processing is being used to free the servers to concentrate on data transfer to ever increasing numbers of clients.

1.5. Browsers

To view documents on the World Wide Web an application known as a Web browser is required, which acts as the client application to the various Web servers that will down load information on the user's request. There are a number of browsers in use, heavily dominated by two, Netscape Navigator and Microsoft Explorer. Others include Mosaic, Opera and HotJava. Some are written for a particular purpose, such as a number that are written in Java for their platform-independence.

The two main browsers tend to set the base line of functionality, in terms of file compatibility, multimedia viewing capabilities, the level of implementation of DHTML (Dynamic HTML see [Section 2.7](#)), the Java language etc. However no single browser has full compatibility with every technology used across the web. Netscape Navigator is at version 4.7 and Microsoft Internet Explorer is at version 5.0 (at the time of writing), but versions 2.0, 3.0, 4.0 and their various sub-versions are all still in use, each more sophisticated than the last and all with different implementations of the various technologies.

Two major issues arise out of this that should be borne in mind when authoring web pages:

- The first is browser compatibility. All through the guidance document reference will be made to the importance of writing as platform (browser and operating system) independent Web pages as possible, to allow accessibility to as wide an audience as possible.
- The second issue is providing functionality to those browsers that do not support a particular file type etc. This is generally done by using a “plug-in” which is a helper application that runs alongside a browser to extend its functionality. If a plug-in could be required to view a Web page, a link to a source should be given.

The browser problem will never really be resolved due to the international nature of the web, the lack of binding standards, the continual development of new technologies and the commercial fight for supremacy over providing the interfaces to what is *the* growth technology of the times: the WWW.

For links to browser resources see [Appendix C Section 1.5](#)

2. INTERACTIVE WEB PAGES - THE TECHNOLOGY

2.1. Hypertext - HTML

HTML is the backbone language of all web pages. Web pages are defined by a file containing a set of HTML codes that tell a browser how to display a page, and what to display on it. The components that go up to make the page, from images to embedded web applications, are referenced in some way from this HTML code.

2.1.1. The HTML language

It is beyond the scope of this report to go into great detail about HTML. For a list of useful references and links to documents on the web see [Appendix C Section 2.1](#) and for books see [Appendix C](#).

HTML consists of a set of instructions, known as *HTML tags*, that define the structure and function of a web page. A browser interprets this HTML code to format the appearance of a web page. The tags define the structure of the web page, leaving it to the browser to use this structure to format the appearance. HTML is not a programming language, such as C++ or FORTRAN are, as it has no commands or functions etc. HTML code defines the structure of a document and what it consists of, and can be interpreted by applications called “web browsers” to produce a final page.

A web page is defined by a file consisting of a number of HTML tags structuring the page, along with any resources (images etc.) it requires to display the page correctly.

HTML code is written into a file as text. It does not need to be compiled or processed for the browser to be able to use it. Therefore it is quite straightforward to produce web pages once the HTML language is familiar, and it does not require the use of complex or expensive tools.

The following diagram illustrates the HTML for a very simple web page:

```
<HTML>
  <HEAD>
    <TITLE>
      Example HTML for a simple web
page
    </TITLE>
  </HEAD>
  <BODY>
    <H1>
      Hello, this is a simple web
page.
    </H1>
    <BR>
    <P>
      It has very little to it.
    </P>
  </BODY>
</HTML>
```



Figure 2.1 A Simple HTML file for a web page and the appearance of the web page in Microsoft Internet Explorer

This web page will display the text “Hello, this is a very simple web page.” as a main heading, with “It has very little to it” as normal text on the next line. In general the appearance of a HTML file bears little relationship with the appearance of the web page. It should always be viewed through a browser to check its appearance after any editing is done.

HTML tags are contained within < >, the “less than” and “greater than” symbols. Most tags are paired up, as on and off switches for a formatting instruction.

For instance in the file in Figure 2.1 there are the <H1> and </H1> tags formatting the text between them to be “Heading 1”. The “ / ” in the second of the pair denotes the “off” tag, which should always be used to close a tag pair. Some tags do not require pairing up, such as the
 tag (used to insert a new line).

For a full listing of HTML tags and their usage, consult one of the resources noted in [Appendix C Section 2.1](#). However there are five important pairs of tags that should be used in all HTML files:

- The <HTML> ... </HTML> tags must be the first and last tags in an HTML file as they enclose the part of the file that the browser interprets;
- The <HEAD> ... </HEAD> pair of tags contain the header information for the file;
- The <TITLE> ... </TITLE> for the title of the page, always in the header section;
- The <BODY> ... </BODY> tags contain the HTML that describes the main part of the page;
- The <H1> ... </H1> heading one tags, that should be used once and once only in the document for the displayed page title.

The section on Authoring Good Practice will discuss these and other components of a web page further.

For links to general HTML guides and tutorials see [Appendix C Section 2.1](#) .

2.1.2. Hypertext and Hyperlinks

One of the great strengths of HTML is in the organisation of information in a non-linear, linked way, where one piece of information can index another directly. The way HTML does this is through the use of “hypertext”, or more generally “hyperlinks”.

A hyperlink is a piece of text or other object, such as an image, that is tagged in the HTML code as an index to some other piece of information somewhere on the web page or site (short links) or to another web site entirely (long links).

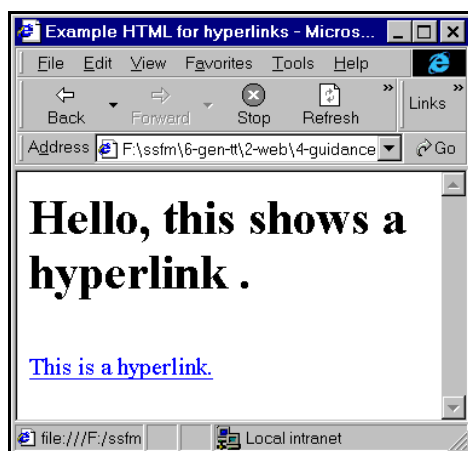


Figure 2.2 A web page containing hyperlinks

This may then be activated (typically using a mouse click) and the browser will jump to that resource allowing a web page author to link the information in a site together.

A hyperlink addresses its target using either relative addressing, where the target is referenced relative to the base directory of the web page using an absolute address where the target is referenced by its full web address.

The base address of a Web page is either its home directory or that defined using the `<BASE>` element in the `<HEAD>` of the document:

`<BASE HREF = "URL of base address of document">`

All relative addressing will be done relative to this base address. The default base address of a web page is the directory that the HTML file is located.

For instance the HTML code below uses relative addressing to create a hyperlink from a thumbnail image to the full image:

```
<A HREF="/images/image1.gif">  
  
<IMG SRC="/images/image1thumbnail.gif" ALT="Link to image1">
```

The HTML code below uses absolute addressing to do the same thing:

```
<A HREF="http://www.name.domain/images/image1.gif">  
  
<IMG SRC="http://www.name.com//images/image1thumbnail.gif" ALT="Link to  
image1">
```

The addresses used are called URLs (Uniform Resource Locators) which are basically World Wide Web addresses. They are written in the form:

`scheme://machine/path-of-file`

The “scheme” is most likely to be either “http” (hypertext transfer protocol) used for Web communications, “ftp” (file transfer protocol) a basic file transfer protocol, or “file” (a file that may be obtained by the client machine) which is for files on a local drive, though other schemes do exist. The “machine” is the name and domain of the server (or local drive) that the file is stored on, and the full path is the path to the file on the server (or local drive).

Hyperlinks can be used to reference files or other resources, so that a user may download them to their local machine. This is done by using a hyperlink to the file’s URL. The file should not be embedded on a Web page if it is to be downloaded.

For instance the HTML:

```
<A HREF = "/downloads/myfile.doc">Download myfile.doc</A>
```

will create a hyperlink to the file myfile.doc in the directory “downloads” of the Web page’s base directory.

A browser will open files if it “knows” how to, such as HTML or GIF files, but otherwise it will prompt the user to decide if they want to open the file using a user defined application, or save it to disk.

2.1.3. Site Navigation

Navigation around a site, either exploratory or in search of specific information, is an important factor to consider. As has already been discussed above, hyperlinks can be used to link two

objects together. Hyperlinks may also be used to link different pages together, allowing navigation page to page.

On a single site there should be one consistent method of navigation, apart from points of entry which will probably have their own custom navigation method. Navigation is often provided by a “navigation bar” in a left hand margin, such as in Figure 2.3, with a list of links, or using more sophisticated technologies (e.g. Java) to produce some kind of menu or site map.

Figure 2.3 Screen-shot of a Simple Navigation bar

However it is possible using other technologies, such as Java or JavaScript, to have more sophisticated navigation bars with components such as drop down menus or functionality like reading the site map from file, allowing all navigation bars to be updated from one source.

The simplest navigation bar consists

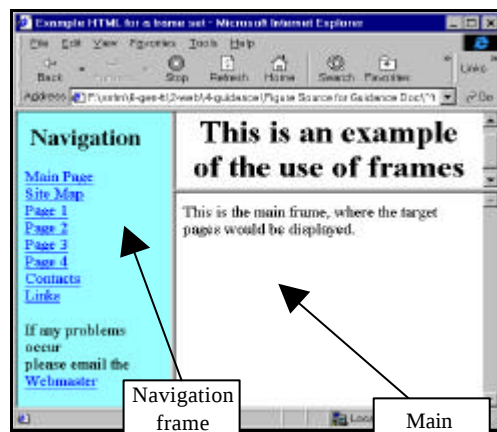


Figure 2.4 Screen-shot of a frame set with navigation frame

It is possible to display more than one web page at one time in a single browser window. This is achieved using what is known as a “Frame Set”, where the window is divided into independent frames, each of which can display a separate web page, as in Figure 2.4.

In this case navigation through the site is often achieved using one frame for the navigation bar, removing the need for a bar on each page (though all pages should provide an independent means of navigation for browsers that do not implement frames) with the pages navigated to appearing in another frame in the window.

Frame sets are standard only in HTML 4.0, and can be tricky to use properly. Care has to be taken to make sure that each window defined by a frame set has its own name, so that when fresh pages are loaded, they can be targeted at the correct one. Also it is easy to link out of the site, but for the new page to remain trapped in the old frame set. It is also hard for a user to bookmark or otherwise record the URL of a page other than that of the frame set document, as the URL displayed in the browser will remain that of the frame set rather than the pages displayed within it.

2.1.4. Mathematical Symbols on Web Pages

The W3C (World Wide Web Consortium) has recommended a mark up language for use on the Web, called MathML. MathML is intended to facilitate the use of mathematical and scientific content on the Web, and for other applications such as computer algebra systems, print typesetting, and voice synthesis. MathML can be used to encode both the presentation of mathematical notation for high-quality visual display, and mathematical content, for applications where the semantics plays more of a key role such as scientific software or voice synthesis. At

present several applets and plug-ins can render MathML in a browser, with browsers able to use MathML code as native in the future. Translators and equation editors which can generate HTML pages where the math expressions are represented directly in MathML will become available within the next couple of years.

Other technologies can be used to display mathematical symbols, such as using a Java applet or an ActiveX component. It is also possible for a GIF image of the desired symbols/equation to be made and displayed, though maintenance and updating of such an image would be relatively difficult.

See [Appendix C Section 2.1](#) for links on MathML and displaying mathematical symbols ..

2.1.5. Good Practice

Good practice in the development of HTML documents is an extensively written about subject, even if the majority of documents are not so titled. There are a number of organisations that are specifically dedicated to the HTML standards and good practice, particularly the W3C and the Web Standards Group.

The main good practice issues should be familiar to any software developer: test it, be consistent with good structure and style and provide documentation. For HTML, testing requires that a document be viewed in all the browsers that the target audience are likely to use. Further steps can be taken by using validation tools to check the structure and to report any invalid HTML. In addition, validation tools will inform the author of any non-standard HTML elements, which is useful for the documentation of any site and informing the user of the required browser or plug-in.

Good structure and style are important in both producing a robust document that will be likely to appear similar in different browsers, but they are also important to maintenance of the page by making the HTML easier to follow. Automatic code generators can produce terrible code by this standard, with a structure and format that, while correct HTML, is extremely difficult to read or modify.

Documentation, given a well structured and styled document, is trivial if necessary at all, as HTML is nearly self-documenting! However the use of non-standard constructs should be documented, if only to provide the user with a source of information about how to view the material.

A more extensive discussion of HTML document good practice is presented in [Appendix B](#).

2.2. Multimedia

2.2.1. Images

Images on the Internet are almost entirely one of two types: GIFs (file name: *.gif) or JPEGs (*.jpg or *.jpeg) though a third standard, PNG, is beginning to be supported. These images are produced by applying data compression algorithms to an image file, such as a bitmap, with the goal of making the image files small enough for practical communication rates.

For instance a half page sized image with just a few colour fields, say 512 by 512 pixels in 8 bit colour (= 256 colours) would be 256 kbytes in size if stored as a bitmap (i.e. every pixel is individually saved) but probably under 20 kbytes as a GIF. This is very large difference which improves transfer times across a network considerably.

GIF and PNG schemes use a lossless compression scheme with moderate compression ratios achievable, whilst the JPEG scheme use a compression scheme that allows varying compression ratios at the cost of an increasing loss of information from the image and therefore quality.

For web links to image information see [Appendix C Section 2.2](#)

- **GIF** (Graphics Interchange Format): GIF is the preferred choice for images with only a few distinct colours, such as line drawings and simple cartoons, or for high quality images. It

supports up to 8 bit colour. GIF is lossless (i.e. none of the original information is lost) and can offer very acceptable rates of compression for simple images.

There are two forms of GIF, the older GIF87a which offers 8 bit colour and interlacing, and the newer GIF89a which additionally offers background transparency and animation. An animated GIF comprises of a series of GIF sub-images which are played in succession like a movie. All modern browsers support GIF89a.

GIF is, however, a proprietary and patented scheme, putting a price on applications that produce and manipulate them.

- **JPEG** (Joint Photographic Experts Group): JPEG is designed for compressing either full-colour or grey-scale images of natural, real-world scenes and supports 24 bit colour. It works well on photographs, naturalistic artwork, and similar material; not so well on lettering, simple cartoons, or line drawings.

JPEG exploits the limitations of the human eye, notably the fact that small colour changes are perceived less accurately than small changes in brightness. JPEG is thus intended for compressing images for human viewers. If you plan to machine-analyse your images, the losses made in the compression will be a problem.

JPEG will offer superior compression ratios to GIF for full-colour or grey-scale images of "realistic" scenes such as scanned photographs and similar material whilst retaining acceptable image quality.

- **PNG** (Portable Network Graphics - pronounced "ping"). PNG format was designed to replace the older and simpler GIF format. PNG has three main advantages over GIF: alpha channels (variable transparency), gamma correction (control of image brightness), and two-dimensional interlacing (a method of progressive display). PNG also compresses better than GIF in almost every case, with an improvement of around 5% to 25%.

PNG's compression is fully lossless and it supports up to 48-bit true color or 16-bit grayscale so saving, restoring and re-saving an image will not degrade its quality, unlike standard JPEG (even at its highest quality settings). However it's compression ratio is not as good as JPEG at its best.

PNG is an open format, it is not patented, but at present browser and graphical support for PNG is patchy although improving quickly.

2.2.2. Animation

Animation on a Web page may be achieved using one of a number of different technologies:

- **GIF 89a.** As has already been described, the GIF 89a data format supports animation. This consists of playing a number of sub-images in order to produce movement.
- **Java** (see [Section 2.5](#)). The Java Application Programming Interface (API) has strong graphical tools, enabling a Java applet to be able to implement animation in a number of different ways, for instance through the display of a number of separate uploaded images in order, much like the GIF animation, or through generating images on the fly. It is good for producing animations of mathematical models etc.

The use of Java allows a lot of control over the animation either at design or run time. Parameters such as play speed, pause etc. can be controlled, and the applet may be written generically to take up and animate all images in a directory to a certain set of play parameters without having to design the applet specifically for each animation.

The standard Java language is only able to read certain image file types at present, namely JPEG and GIF, which are by far the most extensively used. However it has a number of sophisticated tools for manipulating images, such as scaling, grabbing regions of an image, colour etc., and other image file formats are very likely to be supported as they become popular.

- **Client side scripting** (see [Section 2.7](#) and [Section 2.3](#)). Client side scripts are capable of manipulating a document being displayed in a browser. This means that scripts can be used to move and change document contents producing animation, for instance scrolling text across the page etc. The technologies required to do this are collectively known as Dynamic HTML
- **VRML** - Virtual Reality Mark-up Language (see [Section 2.6](#)). VRML can include animation of objects contained in a virtual world. This allows objects to move and change, producing a 3D animation that can be viewed from any angle or distance. This can be very effective, for instance in the visualisation of a dynamic system or object.

It is possible to use Java applets to interface with a VRML viewer to allow sophisticated programmatic control of a virtual world beyond the normal limits proscribed by the world viewer software.

- **ActiveX Components** (see [Section 2.8](#)). ActiveX components can be written to implement animation, through uploading images or generating them on the fly and embedding them on a web page much as with Java applets.
- **Macromedia Flash and Director**. Macromedia Flash uses streaming technology to play an animation as it downloads and renders images using vector based graphics (as opposed to bitmap graphics). Vector graphics produce very small image files for images that can be built out of solid colour shapes, which are generally most useful for animation. They are limited in the complexity of the image they can be used for effectively, and are not useful with photographs etc.

Flash is a drawing tool and a movie editor, allowing the author to draw or import images and cut them together. The Flash plug-in is required to view Flash animations, except in Netscape Navigator 5.0 which will incorporate the "standard" into the browser. For links to resources on Flash graph see [Appendix C Section 2.2](#).

Macromedia Director is a more powerful tool, able to build entire multi-media productions, including animation, movies, sound, images etc. using a visual development interface. The Shockwave plug-in is required to view Director files.

2.2.3. Sound

Adding sound to a web site can be useful. It can provide accessibility for information to the visually impaired, and can be used illustrate some data more effectively than text or images. It is best used as part of a multi-media presentation where the different media support one another.

The need for data coding is similar to that needed for image compression. Non-coded audio data is processed with 8 or 16 bit linear data words sampled with the sampling frequency (e.g. 44.1kHz for CDs). This means that you end up with 1.4Mbit for one second of stereo music in CD quality ($16(16 \text{ bit words}) \times 44,100 \text{ (sampling rate per second)} \times 2 \text{ (stereo)} = 1.4\text{Mbits/s}$).

It can be seen that coding schemes are required to compress this to a manageable level for the Web, particularly if sound and video are going to be sent together. There are many coding schemes in use, but the following are the most widely used:

- **AU**. AU files are the file format common to most Sun Unix workstations, though they are supported by most audio players on PCs platforms as well.
- **WAV**. WAV (for wave) files are a file format developed by Microsoft common to most PC based applications.
- **MP3**. Layer1, Layer2 and Layer3 are MPEG (Moving Pictures Expert Group) audio compression standards. It takes advantage of the resolution limits of the human ear.

Using MPEG audio, one may achieve a typical data reduction of

1:4 by Layer1 (corresponds with 384 kbps for a stereo signal)

1:6...1:8 by Layer2 (corresponds with 256..192 kbps for a stereo signal)

1:10...1:13 by Layer3 (corresponds with 128..112 kbps for a stereo signal)

- **MIDI** - Musical Instrument Digital Interface. MIDI is a different coding scheme from those above. Instead of coding the sound directly, it codes commands to a set of “voices” available to a synthesiser. These commands effectively instruct the voices how to play. Thus MIDI is an interface language rather than a sound encoding language, and is really only useful for music. However it is used to achieve international “jam sessions” over the Internet and a lot of music editing software can work on MIDI files.

For web links to sound file information see [Appendix C Section 2.2](#)

2.2.4. Movies and Video

Using appropriate plug-ins, various movie formats may be viewed through a Web browser. All the movie formats use compression schemes to ensure that the size of the data files are kept reasonable, but even so a couple of minutes of high quality video is likely to be over a megabyte in size.

However some formats support streaming, where the video is buffered at the client end, and played directly from the buffer, so that the user does not have to wait until all the data has been transferred to view it.

Even so, narrow bandwidth communications are not really able to cope with the data rates required for acceptable size and/or quality for video, and this should be taken into account when offering movies to a target audience. In addition to the bandwidth issue, the size of movie files is large. Both the server and the user is going to have to use a significant amount of hard disc space to store them.

Each format requires the video data to be captured and compressed. This requires the use of video equipment and video hardware

The formats discussed here are MPEG, QuickTime and AVI.

- **MPEG (*.mpeg)** (Moving Pictures Expert Group). MPEG (pronounced “empeg”), is the name of family of standards used for coding audio-visual information such as movies, video, music etc. in a compressed digital format. There are 4 MPEG standards:
 - MPEG 1 refers to the delivery of video for a CD-ROM quality presentation.
 - MPEG 2 refers to broadcast quality compressed video.
 - MPEG 3 was targeted for High Definition Television (HDTV). However, it was discovered that with some tweaking MPEG2 would work for HDTV and so it is not used.
 - MPEG 4 is a very low bit-rate audio-visual coding scheme using audio-visual objects.

MPEG 1 and 2 are the only two standards in wide use at the moment as MPEG 4 is a relatively new standard (established end of 1998). MPEG 1 is the standard generally used to show movies across the World Wide Web.

To make MPEG movies an encoder is needed, which will take video data, from a digital video camera for instance, and convert it into an MPEG file. To read an MPEG file a decoder is needed, either as a plug-in for a Web browser or a stand alone application.

- **QuickTime (*.mov and *.qt)**. The Quicktime format was originally developed by Apple for use on their Macintosh and other platforms as a video and audio compression scheme. However it is now supported across most platforms. Various tools are required to edit and play Quicktime movies on a given platform, but Quicktime is most easily captured and edited using a Macintosh system as the system supports it directly.

The latest release, QuickTime 3, includes some powerful features for interactivity and low-bandwidth content. These include the ability to add URL linked-buttons within QuickTime movies, randomly access a large presentation, create a custom movie frame or controller, use

Flash-like vector media, etc. It also supports a number of other video and multi-media formats, such as AVI and MPEG, WAV, AU and MIDI.

In addition Quicktime for Java JDK has been developed by Apple, allowing Java applications and applets to access and show Quicktime movies.

- **AVI (*.AVI)** AVI stands for Audio Video Interleave. It is a special case of the RIFF (Resource Interchange File Format) and is defined by Microsoft.

It supports 256 (8bit) to millions (24 bit) colour encoding as well as sound from 5khz Mono to CD quality stereo sound. A small number of colours and low quality sound gives a better MBit/sec ratio. Using the minimum settings AVI can deliver video at ratios as low as .03MB/sec. However if you use maximum settings, it can run as high as .3 MB/sec.

For web links to movie and video information see [Appendix C Section 2.2](#).

2.2.5. Good Practice

Good practice in the use of multi-media generally consists of considering the best compromise between supporting the content and supporting wide accessibility for the target audience. Many multi-media objects require particular support from the browser it is being viewed in. This may consist of a plug-in, a specific browser etc., which should be considered, and the user notified of the requirement, and a source for necessary plug-in etc.

As with all Web pages, any multi-media pages should be viewed through all of the browsers that the target audience is likely to be using. In addition, it should be noted that the performance of the machine will also be an issue, especially with the processor-intensive movies, and if possible, some minimum specification for acceptable appearance should be found.

To ensure that images are relatively easy to handle, the following guidance should be noted:

- JPEG and GIF are widely supported, with PNG growing in support, so only the choice of format based on image type, quality etc. is at issue;
- Quite large images can be compressed to reasonable sizes, but if a number of these are to be presented, it is best that they are not all placed on one page. One way of dealing with this is to place “thumbnails” of the images on the page, which each link to the full image. A thumbnail is a small, low quality copy of an image, often only 1or 2kbyte in size. Just reducing the size of the image specified in the HTML document does not do this as all the information is still used. Instead, a tool must be used to actually shrink the image;
- If a large number of images are to be presented, consider using a “slide-show”, an application that allows a user to select images to view. Many are written in Java, JavaScript, ActiveX etc., and these often load images in the background while the user is examining one in particular, “reducing” the time that the user waits to see the set of images.

Animations can be more problematic. Animated GIFs are supported by all browsers that support the GIF89a standard, which most do. However animations that are implemented using one of the other technologies can require plug-ins etc. Whatever the requirement is, the user should be given access to information on how to view the animation. Sound and movies should also follow this paradigm.

In general, for capturing video and sound, the initial recording should be made as high quality as possible, using digital technologies. This will ensure that when the data is compressed, encoded and/or changed in resolution, that it is always being transformed from a higher quality source. If at all possible all editing should be done at the highest quality level before transformation is done.

If recording is done to analogue tape, avoid "scrubbing" through the master many times. Excessive playing of the masters will degrade their quality, so the material should only be reviewed a few times prior to capturing it. If it is necessary to review the material several times, it is recommend that a duplicate is made and this duplicate viewed instead of the master.

With video, to get the highest quality results, the video should be captured at full screen resolution (640x480 or 720x486). Even if it is intended that smaller final movies are to be delivered, a full screen capture will generally give better results, and doing so tends to improve the final image.

If it is possible to capture the video at full resolution, any editing and addition of effects should be done at this size. The video should not be resized with an editing or effects package. Also, be sure to render any effects with the highest quality possible.

This guidance is very basic, and any author intending to produce their own sound and/or video should consult a dedicated resource.

2.3. Client Side Scripting

Some applications, such as many modern browsers, are able to understand and carry out instructions written in one or more scripting languages such as JavaScript. Normally scripts are embedded directly into the HTML of a web page. The code is written into the file with the HTML, and does not need compiling.

Client side scripts are programs written in a scripting language which is embedded into the web page and is run on the user's machine by the browser. Some browsers are unable to run particular scripting languages, and may require a plug-in or a more modern browser.

2.3.1. JavaScript

JavaScript is an embedded object-oriented scripting language, normally written directly into the HTML file. It is not generally used to write large programs, and its main use lies in the control of the browser and the appearance of the web page, and it is regarded as a key component of the concept of Dynamic HTML (DHTML, see [Section 2.7](#)). It is widely supported.

As can be seen in Figure 2.6 the JavaScript is written into the HTML enclosed by a

`<SCRIPT> ... </SCRIPT>`

tag pair. This simple JavaScript also illustrates a number of key features of the language:

- JavaScript is object oriented, like Java, C or PASCAL, and so deals with objects, methods and properties. The JavaScript above invokes the "write" method of the "document" object (see the section on the [Document Object Model](#)), and passes it the variable "message". This will change the properties of the document object such that the contents of the variable

```
<HTML>
<HEAD>
  <TITLE>A Simple JavaScript</TITLE>
</HEAD>
<BODY>
<P>This is HTML. </P>

  <SCRIPT LANGUAGE = "JavaScript">
    <!-- Hide from non JavaScript browser

      var message = "This is JavaScript!";
      document.write("<H2>" + message +
"</H2>");

    // -->
  </SCRIPT>

<P>This is HTML again.</P>
</BODY>
</HTML>
```

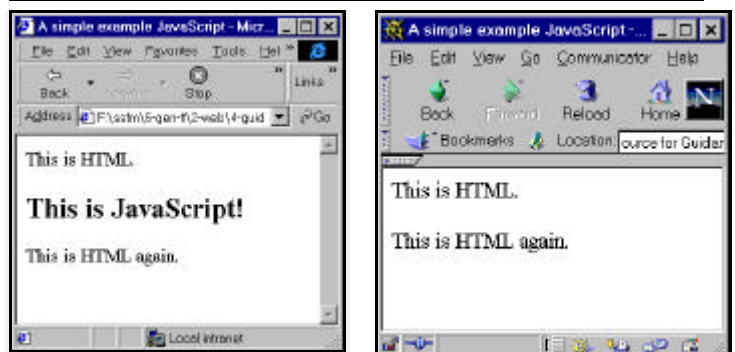


Figure 2.6 A Simple Piece of JavaScript and the resulting Web Page where JavaScript is enabled (left) or not enabled (right)

“message” will be displayed on the screen.

- HTML may be written to the document. The `document.write("<H2>" + message + "</H2>")` command sends the tags `<H2>` and `</H2>` to the document. As can be seen by the output, this puts the text in a `<H2>` element.
- JavaScript is not a strongly typed language, i.e. you do not have to declare the type of the variable (string, integer, float etc.), as this is inferred from the value assigned to it. An example of this is the string variable “message”.
- Not all browsers support JavaScript, and of those that do not all will have it enabled. In this case the browser will ignore the `<SCRIPT>` tags, and write the JavaScript text onto the web page, unless it is hidden between `<!--` and `-->` (HTML comment) tags as above. The JavaScript interpreter will ignore the comment tags.

One very important part of JavaScript is its ability to respond to document events generated by user actions, such as moving the mouse, clicking, writing text into a form field etc. It is this that makes JavaScript useful for interactivity.

The Small JavaScript in Figure 2.6 illustrates this ability to respond to events. The HTML and JavaScript implements a “no-click” hyperlink. The mouse just needs to be pointed at the link to activate it.

```
<HTML>
<HEAD>
  <TITLE>JavaScript No Click Hyperlink</TITLE>
  <SCRIPT LANGUAGE="JavaScript">
    <!-- Hide from non JavaScript browsers

    function go()
    {
parent.location="http://www.npl.co.uk/ssfm/";
    }

    -->
  </SCRIPT>
</HEAD>
<BODY>
  <A HREF="this" onMouseOver="go()">
    Point mouse here to go to SSfM site
  </A>
</BODY>
</HTML>
```

The function “go” is defined in the `<HEAD>` of the document. It can now be called from within another “active” element, i.e. one that can generate an event or another `<SCRIPT>`. In this case the element that calls the go function is an `<A>` hyperlink element. When the mouse moves over the link, the `onMouseOver` event is triggered, and the go function called. To write effective JavaScript a good knowledge of the events, objects and methods available is needed by the author.

When using JavaScript it is very important to test the page on a number of different browsers and platforms as they each

Figure 2.7 HTML and JavaScript for a “no-click” hyperlink.

behave slightly differently. This is particularly true if the JavaScript is written to interact with the DOM (Document Object Model - see [Section 2.7](#)) of a web page, as there are, in particular, significant differences between Netscape Navigator and Microsoft Internet Explorer, for instance. It is possible to set up logic traps based on browser type and version detection so that different JavaScript scripts are run for different environments.

For links to web sites with JavaScript resources see [Appendix C Section 2.3](#).

2.3.2. VBScript - Visual Basic Script

Visual Basic Scripting Edition (VBScript) is a programming language developed by Microsoft for creating scripts that can be embedded in HTML Web pages for viewing with Microsoft Internet

Explorer or other browsers with an appropriate plug-in. VBScript is very similar in functionality to JavaScript and is a subset of the widely used Microsoft Visual Basic programming language. However it is not widely supported.

This small piece of script Figure 2.8, implements a very simple function in VBScript: it adds one to the variable passed to it.

This function could be called from another part of the HTML file in response to some event.

VBScript is not commonly used due to its proprietary nature and platform dependence. The far more common JavaScript has

been integrated into the main modern browsers, and can be considered to be more platform independent.

For links to VBScript information see [Appendix C Section 2.3](#).

2.3.3. Good practice

As with any Web page, it is important to view the page in a number of browsers and provide the user with information on what is required to view the content if not universally accessible. The use of the comment tags within the script element is important to hide the text of the script from non-enabled browsers. If they are not there the whole script will be displayed on the screen as if it were text.

```
<HTML>
<HEAD>
  <TITLE>A Simple JavaScript</TITLE>
</HEAD>
<BODY>
  <P>This is HTML. </P>

  <SCRIPT LANGUAGE="JavaScript"
    SRC = "/JavaScripts/message.js">
  </SCRIPT>

  <P>This is HTML again.</P>
</BODY>
</HTML>
```

```
//JavaScript: message.js//
var message = "This is
JavaScript!";
document.write(message);
```

The external JavaScript file,
"message.js"

Note that C++ style comments // are valid in JavaScript. Also note that no comments are needed in the HTML script element.

Figure 2.9 HTML supporting an external JavaScript (for compliant browsers)

Good software development practices should be used, particularly for scripts longer than a few lines. Good style and structure, with comments etc. should be used to aid maintenance,

upgradability etc. Testing should be undertaken across the range of possible parameters to ensure that the code is reliable and correct.

Other SSfM projects are producing guidance on good software development and testing practices. In particular, a “Guide to Development of Software for Metrology” is being produced. See <http://www.npl.co.uk/ssfm/develop/index.html> for further information.

If a script is to be used on more than one page in a site it is a good idea to use an external script, and link to it from the HTML document, rather than embedding it on each page. This means that if the script is modified, it can be done in one place and aids maintenance etc. The way to do this with JavaScript is shown in Figure 2.x. It is useful to keep all *.js files in one directory, and note how it is no longer necessary to use the HTML comments to hide the JavaScript from non-Java- or VB- script enabled browsers.

2.4. Server Side Scripting

In this case the “script” is run on the server rather than the client. This has benefits and disadvantages as detailed in [1.4 Servers and Clients](#).

2.4.1. The Common Gateway Interface - CGI

The Common Gateway Interface (CGI) allows interactivity between a client and a server through the World Wide Web. It is a communications standard for programs on a server machine to interface with the Web server software.

The HTML in a document delivered by a Web server is static, it doesn't change. However a CGI script may be executed on the server which can output dynamic information, such as results from a database query for instance. Because HTML is text based such a CGI script may then generate a new Web page in reply and deliver it back to the client. The results are then viewed on the client's browser as a Web page unique to the user's initial request.

The common way of gathering request information is through the use of forms where choices are made by the user which are then, once the form is submitted, interpreted by a CGI script.

The term script is used very loosely. In fact a CGI “script” can be written in any language as long as the program uses the CGI standard to communicate with the server software and can run on the server machine. The term arises from the predominant language used to write CGI scripts called PERL, which is very powerful for text input, output and processing.

Examples of CGI scripts can be seen in the NPL remote access laboratory, where potential users for an experiment must necessarily see different pages depending on whether they control an experiment or are just observing one, and the NPL online application form, which posts the application form by e-mail to the correct recipient.

CGI may also be used to interface with databases, either allowing a user to add records or to make queries and view the results. The power of the CGI to create Web pages on the fly in response to user or otherwise generated data is extremely useful, allowing the data to be formatted depending on its nature.

Some very well known examples of CGI programs are the search engines and subject guides that are used to find resources on the web. A search engine automatically searches through a set of web pages and indexes all the titles, descriptions and keywords it finds. The large search engines generate huge indexes of web sites from automatic searches of as much of the World Wide Web as it can reach. When a user initiates a search, they submit a set of keywords to a CGI script, which then uses the index to generate and return a Web page listing the results.

This is the reason Web pages should contain <META> elements detailing title at least, and description and keywords if a major entry point to a site. Search engines will eventually update their index automatically if a web page changes. Search engines may be used on a web site to search that particular site, or even other linked sites and the web in general. Examples of search engines:

- Alta Vista (<http://www.altavista.com>);

- Lycos (<http://www.lycos.com>);
- Webcrawler (<http://www.webcrawler.com>);
- Google (<http://www.google.com>).

A popular search engine that may be placed on a web site for dedicated searches is:

- Excite (<http://www.excite.com/navigate/download.cgi>) (freeware at the time of writing).

Subject guides are a little different. These still consist of searches using CGI scripts and large compiled indexes of Web pages, but the compilation is done by “hand”, either through human searches or the submission of sites. However changes to a Web site are not automatically updated in the index unless a human reviews it once more. An examples of a subject guide is:

- Yahoo (<http://www.yahoo.co.uk>)

For links to examples see [Appendix C Section 2.4](#). For links to resources on CGI see [Appendix C Section 2.4](#)

2.4.2. Server Side JavaScript

The main server side scripting is done using the Common Gateway Interface (CGI). However some servers are able to run JavaScript (and a few - VBScript) directly, allowing server side scripts to be written without using CGI.

HTML pages that use server-side JavaScript are compiled into bytecode executable files. These executables are run in concert with a web server that contains the JavaScript runtime engine. The developer creates HTML pages (which can contain both client-side and server-side JavaScript statements) and JavaScript files. You then compile all of those files into a single executable.

When a client browser requests a page that was compiled into the application, the runtime engine finds the compiled representation of the page in the executable, runs any server-side JavaScript statements found on the page, and dynamically generates an HTML page. The result of executing server-side JavaScript statements might add new HTML or client-side JavaScript statements to the original HTML page, and when done the runtime engine sends the generated page to the client, which runs any client-side JavaScript and displays the results.

Using JavaScript in this way has similar advantages and disadvantages to CGI, except that a single language can be used to perform both client and server side scripting and the server side scripting is integrated directly into the HTML documents, aiding maintenance etc.

Java applets may be run on the server side, using a similar paradigm. When they are run in this way they are known as “servlets”.

2.4.3. Good Practice

See client side Good Practice, [Section 2.3.3](#).

2.5. Java

Java is probably the single most popular and important language being used to develop interactive functionality for the Web. This is due to the significant strengths it offers in platform independence and high levels of security. It is very difficult, some say impossible, to write a malign program (a virus for instance) in Java. There is a growing demand and market for numerical algorithms written in Java, and for services supplied via the web that require sophisticated programmatic support, with Java the only language really in a position to supply this demand.

The reason for this is that Java is a language specifically designed to be platform-independent, i.e. it may be able to run on any computer. This was achieved through the use of a Virtual Machine (VM) for each platform which interprets and runs the Java code, rather than having to compile and run the program as “native-code”. (Native code is code that has been specifically compiled to run on a particular type of machine and generally will run faster than interpreted code).

This means that once a Java VM is developed for a platform, applications written in Java will be able to run on it without any changes. The term “write once, run anywhere” was coined.

A new generation of Java VMs is near completion, called Just-In-Time compilers, which compile the Java on the fly into native code and run that, rather than interpret it. This offers an overall faster run-time, offering benefits of both platform independence and native code speed.

Java is an Object Oriented Programming (OOP) language, and is similar to C++ in style and syntax, using classes to define objects, methods and properties etc.. The main difference is in its

Example Package	Functionality
java.lang.*	Classes most central to the Java language - e.g. “String” class.
java.awt.*	Abstract Windowing Toolkit - classes used to do graphics, windowing and layout - e.g. “Window” class.
java.applet.*	Classes used for writing applets - e.g. “Applet” class.

Figure 2.10 Examples of Java Packages

lack of “pointers” (variables that “point” to a memory address allowing the program to read or write directly to it). This was done for security reasons since uncontrolled memory access potentially allows information to be retrieved from a client machine without permission, and dangerous code to be run, such as viruses. The security model used keeps all interactions with local files to a minimum and only allows secure interaction with the computer from which the Java program was downloaded or when given explicit and specific permission to do so.

The Java API consists of a large number of classes that are used to write a program. The classes are organised in a structured hierarchy of “packages”. A package is a set of classes with an overall purpose, like graphics and windowing, maths, input/output, networking etc. In addition a developer can create new packages of their own.

Three key packages are shown in Figure 2.10 above. The name of a package defines where it is stored on a machine, its functionality or origin and should indicate the source of the code relative to some library directory defined in the environment variable CLASSPATH of the computer, or the CODEBASE element of the <JAVA> HTML tag.

For instance java.lang.String is stored in the directory “/java/lang/” of the Java class library, and is a core Java API class. The environment variable CLASSPATH should point to the Java Class Library as a library directory. The class npl.cise.DataTable is found in “/npl/cise/” and is a class developed by the National Physical Laboratory (NPL), Centre of Information Systems Engineering (cise).

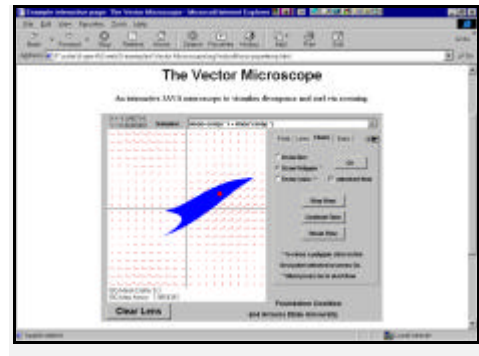


Figure 2.11 Screen shot of an example applet

The main use for Java is in downloadable, platform independent applications providing interactivity on web pages. These applications are known as “Applets” (little applications) and most modern browsers implement a Java VM capable of running them through the interface provided by the Applet class. All applets are sub-classes of the Applet class which is found, unsurprisingly, in the java.applet package.

Applets are generally quite small applications, but there is no imposed limit on their size. Their size is generally limited due to considerations of download time.

An applet is embedded into a HTML document using the <APPLET> element, as shown in the HTML document on the left, which illustrates a number of features.

```
<HTML>
<HEAD>

  <TITLE>
    Example of embedding an applet
  </TITLE>

</HEAD>

<BODY>

  <H1 ALIGN="center">My Applet</H1>

  <APPLET CODE="MyApplet.class"
  ALT="MyApplet"
    NAME="MyApplet" HEIGHT=100
  WIDTH=100>

    <P>
      Your Browser does not support Java<BR>
      Please enable Java or update your
  browser
    </P>

    <PARAM NAME="param1" VALUE="My applet">
  </APPLET>

</BODY>
</HTML>
```

<APPLET> elements should have at least the properties shown. The CODE property defines the path and applet class file from the document base URL. The ALT property defines the alternative text to be shown while the applet is being retrieved. The NAME property allows other objects in the document to identify and communicate with the applet. HEIGHT and WIDTH define the size of the applet.

HTML enclosed within an APPLET element is ignored unless the browser is not Java compliant. This means that a message may be inserted as above warning a user that their browser is unable to display the applet.

Figure 2.12 Embedding an applet in an HTML document

The <PARAM> element is an applet parameter, where the applet can retrieve the VALUE by using the parameter NAME. This is the mechanism by which an applet is externally parameterised.

All applets should have implemented “getParameterInfo()” and “getAppletInfo()” functions, for use by the browser and by other developers that wish to use the applet themselves. These functions should return all the information required to use the applet’s interface.

In addition, applets on a single web page may communicate with one another, allowing applets to “serve” the needs of others on a page. For instance a data server, generating data from some source, may be placed upon a page, and other applets may register with the data server to be supplied with some component of that data.

Java Beans

Specific to Java is the general specification of a component known as a “Java Bean”. These beans have a specific format to allow the use of component assemblers, known as “bean boxes” to be used to assemble them into a desired application, and compile them, without the need for further coding by the author.

They are not automatic code generators as such, as the important code is already written within the beans, but the bean box does generate code supporting the interaction between the components. However this is generally only a small part of the whole code.

The size of an application developed using beans can sometimes become an issue, as they are often larger than the code for an application developed from scratch. This is due to the code required to implement the Bean model, and the code that is required for a generic component, but not for the specific application. This is not normally a problem for small applications, but for larger ones, it can become significant, effecting download times, for instance.

To sum up, Java is a strong, secure language, providing a good API which is particularly strong in graphics and GUI development that allows developers to produce sophisticated interactive Web content.

For links to information on the Java language see [Appendix C Section 2.5](#).

2.6. VRML

VRML was developed to provide a generic language for defining a three dimensional interface, much as HTML can be used to define a two dimensional one. VRML is intended for the design and use of three-dimensional, multi-person, interactive simulations, otherwise known as Virtual Reality worlds.

```
#VRML V2.0 utf8

Group
{
  children
  [
    Shape
    {
      appearance Appearance
      {
        material Material
        {
          diffuseColor 1 0 0
        }
      }
      geometry Box {}
    }
  ]
}
```

Figure 2.12 A Simple VRML document defining a red cube.

of a “Children” field where the child components of the group are defined, in this case a Shape node.

The Shape node has an appearance field which contains an Appearance node, which itself has a material field containing a Material Node in which the colour is defined. VRML uses RGB colour definitions, with the three values equalling the (percentage colour saturation)/100. The Shape node also has a “geometry”, in this case a box, which will render a cube.

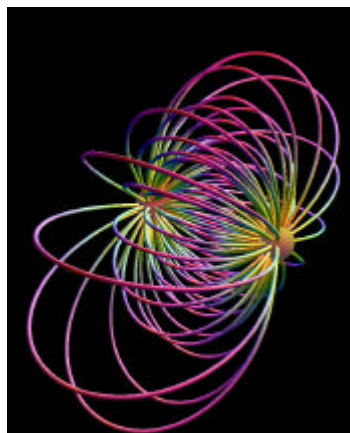


Figure 2.13 A screen shot of a VRML world of the magnetic field around a magnetic dipole

The VRML language is used to define objects, either simple geometric solids, such as spheres, cubes etc., or solids with an arbitrary set of vertices and faces, which may themselves be built up into more complex hierarchical objects. The objects can also have complex behaviours in response to internal and external stimuli.

An example VRML document is shown to the left. This VRML code defines a simple red cube floating at the default starting point.

The hierarchy of this code can be seen. Everything enclosed within the “Group” parenthesis are regarded as a single object by VRML. This is called a Group “Node”. The group node consists

The header line is a comment (denoted by #) declaring the VRML standard being used and the encoding scheme, in this case enabling international character display (utf8). This file is called a “VRML world”.

Far more complex objects can then be build up out of a set of simpler objects. Figure 2.13 shows a screenshot of a VRML world modelling the magnetic field around a magnetic dipole.

To view a “world” a VRML viewer (a browser plug-in like CosmoPlayer) must be used and a link to VRML file (*.wrl) activated. This allows the user to move their point of view (POV) around the “world” in three dimensions.

There are many examples of such “worlds”, used for a variety of purposes, from architectural visualisation or data visualisation to virtual cities or virtual training courses.

For web links to VRML resources see [Appendix C Section 2.6](#)

2.7. Dynamic HTML

Dynamic HTML (DHTML) is the set of technologies that enable the content of a Web page to change after the HTML document has been downloaded. DHTML is therefore not any one specific technology (such as Java or ActiveX). It is a concept that is made up of (to different extents in different browsers) by a number of technologies, including client side scripting, the Document Object Model (DOM), Cascading Style Sheets (CSS) and HTML itself. To write a DHTML page knowledge of each of these is required to develop a single page where these technologies interact to produce the desired effect. For links to DHTML information and examples of use see [Appendix C Section 2.7](#)

Client side scripting is discussed in [Section 2.3](#).

2.7.1. The Document Object Model (DOM)

The Document Object Model is the core of dynamic HTML, developed by the W3C. It enables scripting languages to manipulate the content of a page after it has been downloaded to the client. The DOM describes the hierarchy of elements that are present in the browser at any given time. This includes environmental information such as the current date and time, browser properties such as the browser's version number, window properties such as `window.location` (the page's URL), and HTML elements such as text, `<P>` tags, `divs`, or tables.

By exposing the DOM to scripting languages, browsers enable you to access these elements. While some elements such as the time of day can't be changed themselves, they can be used by scripts to modify other elements.

The part of the DOM that specifies which elements can trigger changes is called the “Event Model”. Events are generated by actions like moving the mouse over an element (`onmouseover`), loading a page (`onload`), submitting a form (`onsubmit`), clicking on a form input field (`onfocus`), and so on.

The triggering of an event can be keyed to a function written in a scripting language, so that for a specific event, such as a mouse click, the page is changed dynamically in some way, such as starting an animation for instance.

For links to Document Object Model information see [Appendix C Section 2.7](#)

2.7.2. Cascading Style Sheets (CSS)

Cascading Style Sheets (CSS) are a mechanism for separating formatting information from content to allow external maintenance of formatting information. This allows corporate styles to be easily maintained. A CSS is effectively a template defining one or more styles for HTML elements, which can then be applied to as many Web pages as is wished depending on how they are used. Both Netscape Navigator 4.0+ and Microsoft Internet Explorer 4.0+ support cascading style sheets, at least to some extent.

You can use style sheets in three ways:

- By **linking** to a style sheet from your HTML file using a `<LINK>` element. This method allows you to change the appearance of *multiple Web pages* by altering a single file. The CSS in this case is a text file containing the style codes.

```
<HEAD>
  <LINK REL=stylesheet
    HREF="path/MyStyleSheet.css"
    TYPE="text/css">
</HEAD>
```

- By **embedding** a style sheet in your HTML file using a `<STYLE>` element. This method allows you to change the appearance of a *single Web page* by altering a few lines. The CSS in this case is a block of text enclosed by the `<STYLE>` element.

```
<HEAD>
  <STYLE> H1{font-size: 15pt;
           font-weight:bold;
           color: maroon}
</STYLE>
</HEAD>
```
- By adding **inline** styles to your HTML file by using the style as a property of an element. This gives you a quick way to change the appearance of a *single tag*, a *group of tags*, or a *block of information* on your page. The CSS in this case is the text defining the style as a property of an element.

```
<BODY>
  <P STYLE="font-size:20pt;color:red">
    Paragraph of text.
  </P>
</BODY>
```

The term cascading is used to indicate the rules that apply when more than one Style Sheet is used on a single element. The rule is: a linked style sheet over-rides the browser default styles, an embedded style sheet over-rides a linked style sheet and an inline style over-rides an embedded style sheet.

An example of the use of a linked CSS is shown in figure 2.14 called “example.css”. The CSS file consists of a set of styles. The first one sets the defaults for the headings H1 to H3 as centred bold Arial font. However each heading type is then individually specified. Each inherits the earlier styles, but where a more specific style is used, such as for H3 where the alignment is set to left, this one takes precedence. Finally two “classes” are specified, (highlight and copyright) which may be applied anywhere as a property of an element.

The example HTML file is a simple one with three different elements used to format the text; the first three heading types and the paragraph element. The HTML file links to the CSS example.css, and the link must go in the header section of the document. The two screen-shots show the resulting web page viewed with a non-CSS enabled browser and then with CSS enabled.

To change the style of the Web page, and any other Web page using the same CSS now only requires changing one file rather than having to go through each document.

CSSs do allow far more than basic styles. They allow the positioning of text on the screen either through setting margins etc. or by direct positioning, control of the background (for instance so it does not scroll with the foreground), and a large number of other implemented and proposed functionalities.

Because they are part of the DOM, CSS properties are accessible to scripting languages, and it is therefore possible to change almost anything about the way a page looks. By changing the CSS properties of a page element (such as its colour, position, or even animation), you can do almost anything bandwidth and processor speed permit.

For links to Cascading Style Sheets information see [Appendix C Section 2.7](#)

<pre> H1, H2, H3 {text-align: center; font-family: Arial; font-style: bold} H1 {font-size: 40} H2 {font-size: 34; font-style: italic} H3 {text-align: left; font-style: italic; font-size: 28} .highlight {font-style: bold; background: yellow} .copyright {text-align: left; font-size: 22; font-style: italic} </pre>	<pre> <HTML> <HEAD> <TITLE>Using CSS</TITLE> <LINK REL=stylesheet HREF = "example.css" TYPE="text/css"> </HEAD> <BODY> <H1>This is heading style 1</H1>
 <H2>This is heading style 2</H2>
 <H3>This is heading style 3</H3>
 <P CLASS="highlight"> This is normal text using the CSS highlight class. </P> <P CLASS="copyright"> This is a copyright notice. </P> </BODY> </HTML> </pre>
---	---

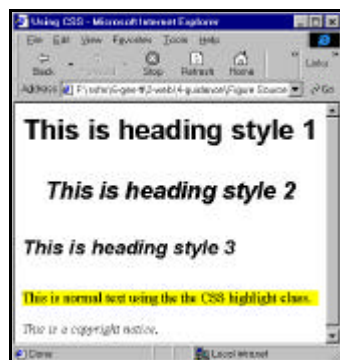


Figure 2.14 An example of a linked
Cascading Style Sheet.

From top left: The CSS, the HTML document, The
appearance of the Web page with the CSS and its
appearance without the CSS.

2.8. ActiveX

This set of technologies from Microsoft provides tools for linking desktop applications to the World Wide Web. Using a variety of programming tools such as Java, Visual Basic or C++, developers can create interactive Web content. For instance, ActiveX technology can allow users to view Word and Excel documents directly in a browser.

ActiveX is a stripped down version of OLE (Object Linking and Embedding) targeted for the World Wide Web. An ActiveX component or control is a software component written in whatever language that conforms to the ActiveX API (Application Programming Interface).

An ActiveX object or control resides on a Web page to be downloaded and run when Microsoft's Internet Explorer or other browser with an appropriate plug-in views the page. For instance an ActiveX control can be the ActiveMovie ActiveX control which plays AVI (Microsoft Audio Visual Interleave) and other video formats supported by ActiveMovie, or show a set of images or an animation, or whatever. In this sense, ActiveX objects are similar to Java applets. Unlike Java applets, an ActiveX control has been downloaded once, it remains on the downloading machine and can be used again without being downloaded again. However, ActiveX controls are tied to a particular architecture, Windows 95 or Windows NT, which is a major disadvantage.

For links to ActiveX resources see [Appendix C Section 2.8](#)

3. INTERACTIVE WEB PAGES - APPLICATIONS

The technology for a wide range of different forms of interactivity of the Web is available. Discussed here are a number of general areas of application that are relevant to the NMS.

3.1. Data Transformations

Interactive data transformation applications offer the provision of generic and specialised data transformation services, such as data fitting, data fusion etc.

These applications allow a user to explore the parameter space of the data transform, enabling them to choose an optimal solution, or produce a number of solutions for later comparison. For example in a fitting application the parameter space would consist of different orders of fit, different fitting regimes etc., which could be worked through, manually or automatically, to provide an optimal fitting parameter set.

There are many benefits in the provision of such a centralised service, most significantly the consistency of results across the client audience and the confidence that a single properly engineered and tested application provides. Also as only one copy of the application need be online at one time, version control and maintenance become more straightforward and efficient. This advantage is a feature of many of the example application areas detailed in this section.

Such a service could be provided in two ways, either client side processing or server side. The general benefits/disadvantages of these are discussed in [Section 1.4](#), such as the “cost” of the download time of the application against the “cost” of the response and data up/download times.

A particular issue, however, is the reuse of software. There is a lot of effective, well tested and validated software that is written in languages that are not platform-independent and so not appropriate for download. The NPLfit library routines, an extensive and powerful set of data fitting routines written in FORTRAN, are a good example of this. In general, for this kind of software, re-coding, into Java for example, is a huge piece of work. A better solution to this may be to provide a Web page interface, and for the processing to be done remotely. An example of exactly this type of Java applet has been developed for demonstration purposes and may be found on the NPL SSfM web site. To view this example visit:

http://www.draft.npl.co.uk/ssfm/tt/ssfm_interactive_examples/index.html

For links to examples of data transformation applications see [Appendix C Section 3.1](#)

3.2. Charts and Graphs

The charting and graphing of data is a key part of the work done by the NMS. Web based applications using this capability offer a number of possibilities that could greatly enhance NMS web sites and the information presented there.

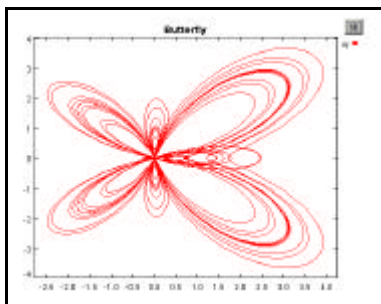


Figure 3.1 A Butterfly graph produced on-line using a Java applet

Interactive charting and graphing may be utilised in two generic ways.

- As a service to users, who may use it to find an optimal graphing solution for their data. This could be combined with a data transformation or modelling application to provide an enhanced service, allowing users to examine and contrast results from different parts of the transform or model parameter space.
- As a component in a document, allowing the reader to explore the data space of any relevant data graphically as they consider the information contained by the document. This would enhance their understanding and uptake of the information.

For web links to examples see [Appendix C Section 3.2](#)

3.3. Interactive Models

The use of mathematical and visual modelling is an important subject for all parts of the NMS, from producing test data for another system, through simulation and system visualisation, to theoretical model validation.

The use of web technology in this area includes:

- Data services to users, enabling them to set parameters and retrieve the resulting data set. This could either be a generic model, such as one to produce data sets exhibiting some specified mathematical behaviour (useful in producing test data), or one specific to a particular system, such as the prediction of system behaviour under a certain set of conditions.

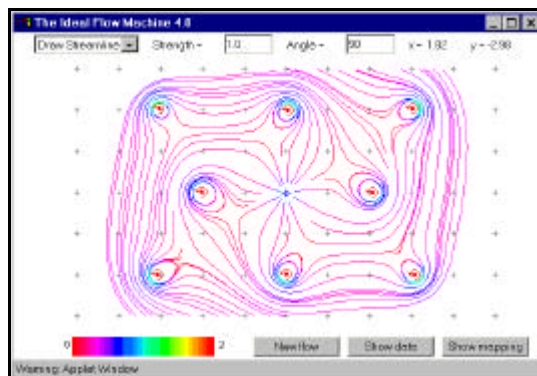


Figure 3.2 An interactive Ideal Flow Model implemented as a Java applet.

- Visualisation of system behaviour. A simulation of a system where the user may control the parameters and results are displayed in some way (possibly through graphical techniques, or an animation of a physical process) allows the user to gain a deeper understanding of its behaviour, and allows them to “target” specific parts of the system’s parameter space.

This not only enhances understanding, but would allow the development of particular experimental strategies, or would inform development of research into new areas.

An example of this would be a model of a CMM (co-ordinate measurement machine) allowing measurement strategies (position of sensors, movement etc.) to be tested and verified before attempting them using the real machine. Given that it would take only a short time to configure the model compared to the actual machine, this could save huge amounts of time. The model could be enhanced, for instance, to include estimation of errors with each strategy, so that an optimal strategy could be found.

- Dynamic Documentation. Embedding a model in a document is one of the most powerful ways to illustrate the behaviour of a system. Presenting a dynamic model that the reader is able to manipulate, particularly in real time, dramatically enhances the impact of a document and its ability to transfer knowledge. It can provide a learning experience second only to “hands on” experience, which itself may not be possible in theoretical or unrealised work.

An example of this has been developed for demonstration purposes and is on the NPL SSfM web site : http://www.draft.npl.co.uk/ssfm/tt/ssfm_interactive_examples/index.html

- Training models. Where it is desirable to transfer a skill as well as knowledge, it is possible to use a model to allow the user to “practice” this skill in some way. This could be as simple as a series of images demonstrating the skill to a fully interactive virtual reality simulation.

One example of this would be in the assembly of an instrument or tool. The simulation would consist of one or more “scenes” where parts would be assembled through some interface, leading to the assembled object. Of course without fully immersive virtual reality, which is currently not feasible, this would only be an analogue of the skill, the user would then have to apply what they learnt to the real situation. However, this style of skill transfer is far superior to static documents.

Another example of this is an instrument calibration model, designed to train a student in the correct procedure to perform the calibration using a “virtual instrument” and appropriate tools.

For web links to examples of interactive models see [Appendix C Section 3.3](#)

3.4. Interactive Control and Data Acquisition Systems

The use of online control and acquisition services is one that is presently being explored by a number of different groups, including NPL, from simple control and image acquisition with a digital camera, through the control of a robot's movement in real time to the implementation of a "remote lab", such as the one currently under development at NPL, where experimental procedures can be controlled and monitored from a remote machine.

A few specialised systems for this already exist, such as in the remote use of remote sensing instrumentation (astrophysical observatories, weather satellites etc.). The "On-Line Barograph" at NPL is a simple example of a data acquisition site, where the data produced by the barograph is displayed in real-time.

See the web page at: <http://www.npl.co.uk/npl/sections/pv/pressure.html>

For links to other examples of remote control systems see [Appendix C Section 3.4](#)

3.5. Database Queries

The provision of access to a database through a web page is already being used in a number of industries, particularly finance. In general this provides the user with access to a database that they would otherwise have to install on their machine, which is often impractical for the user (size, cost etc.) or desirable on the part of the server (in the case of proprietary data). Provision of a remote access interface provides the benefits of a centrally updated and maintained database which is under the control and security of the host server.

NMS provision could include access and search functions for databases of data sets.

For web links to examples see [Appendix C Section 3.5](#)

3.6. Miscellaneous Web Page Enhancements

For links to Miscellaneous Web page interactive enhancements see [Appendix C Section 3.6](#)

3.6.1. Image galleries

Web applications can provide a convenient way to present large sets of images to a user, whether these be photographs, illustrations, scanned images etc. When they are combined with search engines they are a particularly powerful tool for archiving images. These applications are generally known as Image Galleries or Slide Shows.

3.6.2. Image Maps

It is possible for an image to be used to implement multiple hyperlinks. When this is done the image is divided up into sections using a Cartesian co-ordinate system to define hot-spots or areas which are each tagged as a hyperlink. This resulting construct is called an "image map", and can be very useful as a context based navigation tool.

There are two classes of image map, server side and client side:

- Server side image maps are processed by the server with input data sent from the client machine and an image map file. This output is the target URL, which is used to retrieve the target document. This form of image map has become rare since HTML 3.2 was implemented in most browsers, which allowed the use of client side image maps.
- Client side image maps have all the information required to implement the image map locally. When the user clicks on the image, the URL is produced immediately from the HTML for the browser to use. This has considerable advantages over the server side type. It is processed on the client's machine, freeing resources on the server and the client machine does not have to send the co-ordinate data and then wait for a reply from the server machine to process the target URL.

It is also possible to implement image maps using embedded applications, such as applets, Java Script, ActiveX components etc., which will be discussed later.

4. TOOLS FOR BUILDING INTERACTIVE WEB PAGES

This section examines some of the range of tools available for the development of interactive web pages and their components. Tool categories are generically described, and then the generic tools type briefly discussed.

A software tool may be defined as a piece of software used to design, create, modify or parameterise a piece of software or data. This includes programming environments, compilers, data generation applications, editors of all sorts, and almost anything that is used to perform some process. Given this extremely wide definition, the tools examined here are those specifically designed to create and modify web pages and interactive components.

4.1. Hypertext - HTML

HTML editors split broadly into two main categories:

- Editors based primarily on HTML. Require significant knowledge of HTML;
- Editors which are WYSIWYG (What You See Is What You Get) editors. Do not require much knowledge of HTML - they use a visual development interface;

For links to HTML editors, see [Appendix C Section 4.1](#).

HTML Code Editors

An HTML file is just a text file and so may be created and edited with any text editor. However, dedicated HTML editors provide features such as error checks on the HTML tags, shortcuts for inserting and configuring common tags, provision of reference and “help” materials, templates, and site management and publishing aids.

If a web page is created by writing the HTML in this manner, a web browser is used to view the results of editing, though it can often be started directly from dedicated HTML editors.

Advantages: Writing the HTML directly allows the author to have in-depth understanding and control of the web page’s appearance and HTML code. With good coding practice the HTML code will generally be easier to follow and maintain than HTML code generated automatically.

Disadvantages: The author must be competent with HTML. If using a basic text editor, the file must repeatedly be saved and viewed through a browser. The user must “hand code” a significant amount of the HTML, which can be time consuming.

WYSIWYG Editors

WYSIWYG (What You See Is What You Get) editors utilise a GUI (graphical user interface) to allow the author to design a page in a similar way to DTP (desktop publishing) tools. In fact many DTP applications and even word processors now have HTML publishing tools integrated into them or available as “plug-ins”.

Pages are designed by placing objects (text, tables, images etc.) onto a work space representing the web page using mouse, keyboard etc. The application automatically produces HTML code as this is done, so that the web page viewed by a browser matches the page designed.

Advantages: A web page can be produced very quickly, and its appearance can be changed easily. This can be used to produce the skeleton code/template for a web page, then code based editing used to finish it.

Little or no HTML knowledge is required, as a web page can be produced using only the visual development environment.

Disadvantages: Code that is automatically generated is often poorly structured with many unnecessary or misleading items in the code, making it harder to follow and to maintain than well structured hand written code.

In addition, such editors often restructure or replace hand coded HTML tags with their own automatically generated code, which can ruin code structure or, even worse, implement unwanted code.

The nature of HTML at present is that different browsers will display the same HTML in different ways. To ensure that this is a minor issue can require careful consideration of the HTML code used, something that can be a problem with automatically generated code.

It is also true that what-you-see in the editor can be different to what-you-see in a browser. The only guarantee that what you see *is* what you get is to preview the page in all the different browsers that users are likely to have. If the appearance is wrong, there is a good chance that the only remedial action is to alter the HTML by hand.

Hybrid Editors - both HTML code and WYSIWYG.

Almost all editors have some kind of mixture of the code based and WYSIWYG editing, but some have made hybrid editing a distinct feature.

These editors will allow the user to edit a web page using one or more of a number of different screens, where a change on one will update the others simultaneously.

Advantages: Authors with a good knowledge of HTML can benefit greatly from the multiple views of the HTML document. They can change something in one window and immediately see what happens in the other. This brings the advantages of both previous types together.

Disadvantages: As they bring the advantages together they can also bring the disadvantages. The multiple windows can present a confusing interface for novice authors, and to benefit from the added functionality authors need to have familiarity with HTML.

Document Conversion - to Web page and other Web presentation formats.

A number of tools exist for converting documents into HTML format. The benefit of this is to allow documents to be written using some straightforward common tool, such as a word processor, and then the HTML document be published by an “expert” using a conversion tool. This allows the task of publishing Web documents to be divided effectively between the expert writer and the expert publisher.

However, conversion tools can be unpredictable in their output, and will generally suffer from the same failings as HTML code generators do, in that the HTML will be messy and difficult to maintain. Also it is generally true that documents designed specifically for the web will make the best web pages. Ones converted from other file formats will not be of as high a quality.

It is often good to present a document in “*.pdf” format (that produced and read by Adobe Acrobat tools) in addition to any other method. This will allow almost anyone (the Adobe Acrobat reader is widely available) to access a base line of the information you wish to present. The Adobe Acrobat reader does not allow the document to be revised, but does allow it to be saved and printed, and it will support hyperlinks, either internal to the document or external. Allowing the document to be downloaded and saved in this format also allows it to be read by those who do not have or are not permitted (such as in many security areas) access to the Internet, but wish to be able to read the document.

4.2. Multimedia

4.2.1. Images and Animation

Almost any browser has support for viewing JPEG and GIF images, including animated GIFs. There are a large number of applications for importing, manipulating and exporting image files, from almost any one format to any other.

Generally the tools will allow the author to resize and crop the image, with tools for manipulating the contrast, brightness, colour depth etc. of the image. Many will also allow re-touching, composite pictures, etc.

Most will also allow the author to export the file in a number of formats, with control over the parameters of such an export. For instance with an export as a JPEG the author will have control over the amount of compression, and thus the quality of the compressed image.

While there are many applications, one is often quoted as being the forerunner in quality and features, and is often recommended: Adobe Photoshop.

There are many tools available capable of producing GIF animations. A tool will take a number of images, either generated using the tool itself or imported from a 3^d party source, which are sequenced together and saved as in GIF89a format as a set of subimages. The image update frequency can often be set, controlling the rate at which the GIF animates. The size of the GIF will be approximately the sum of the sizes of the images (in GIF format) that make it up. The restrictions of the GIF format as regards to the compression rates achievable based on the type of the image still apply as with a normal GIF.

4.2.2. Sound

Most browsers support playback of most sound formats. Where a sound format is not supported there is often a plug-in or other application available that may play the format.

Microsoft Windows, for instance, has the Microsoft Media Player (which can also be downloaded for the Apple Macintosh) which can play most formats, such as *.au, *.wav., *.mp3 etc. If an application is not already available then there are many that may be downloaded.

There are also many applications for the encoding of sound, either taking the input from an external source through a sound card or a specialised sound capture card, or from a sound format file.

4.2.3. Movies

Most browsers require a plug-in to be able to play movies, or there must be a system application that is capable of doing so. Microsoft Internet Explorer Active Movie player for instance is capable of playing most movie formats. Apple Quicktime 3 is also capable of reading and playing most movie formats, and is available on Windows and Macintosh platforms.

Encoding applications are capable of importing video data, either captured directly from a video capture board connected to an external source (e.g. a digital video camera) or from a video file. The encoder can then be used to process and export the video data, often giving the option of a number of formats. Adobe Premiere, for instance, may export in Quicktime, AVI, and GIF animation formats. It also provides a number of tools for processing the image, through filters, compositing, speed etc. It can export audio only files in a number of formats also.

See [Appendix C Section 4.2.](#) for links to Multimedia tools.

4.3. Client Side Scripting

Scripts provide interactivity or multi-media access to a web page, for instance through the provision of a navigation bar, image map, animation etc. Most HTML editors provide some level of support for writing scripts, and many also provide the ability to use pre-written script "templates" or may generate code automatically for the author, based on some user defined parameters.

These benefit the author by freeing them of the burden of hand coding, but automatic code generators often produce code that is difficult to read and maintain.

Beyond these automatic script generation tools within Web page development applications, there are few tools supporting the development of scripting languages.

See [Appendix C Section 4.2.](#) for links to Client side scripting tools.

4.4. Server Side Scripting

CGI scripting may be performed in any application programming language that the server software may interpret or run. All that is required is that the CGI interface is used. This means that the tools required for CGI scripting depend on the language being used to program the script. The common languages used are: PERL, Visual Basic, C and C++ and sometimes Java.

- **PERL** - PERL is an interpreted language. The server must have a application capable of interpreting the script. The PERL script may be written using a text editor, and a PERL interpreter is required to run it.
- **Visual Basic** - Visual Basic (VB) is a programming language developed by Microsoft which provides a powerful Visual Development Environment for Microsoft Windows platforms. The programs are compiled into native code, and so will run without the need for an interpreter. However a run-time library in the form of one or more *.dll (dynamic link library) files is required. A copy of Microsoft Visual Basic is required to develop and compile VB CGI “scripts”.
- **C and C++** - C and C++ are similar languages, C being a subset of C++. C++ is an object oriented programming language and is very powerful, being able to operate at a very low level. It is a compiled language, producing native binary code for whatever machine it is compiled for. The code may be written using a simple text editor, and compiled using a command line compiler, such as g++ from GNU, or one of the powerful visual development environment tools used, such as Microsoft Visual C++ or Borland C++.

See [Appendix C Section 4.2](#) for links to Server side scripting tools.

4.5. Java

Recent developments have allowed web pages to expand their interactivity through the ability to run certain applications through a browser.

The predominant language used to develop such applications is Java. All modern browsers implement the Java Virtual Machine which allows specially configured Java programs called “Applets” to run on them, if it is enabled.

Developing Java applications and applets requires that an author has a development tool available to them, allowing them to compile Java code. Such tools range from the JDK (Java Development Kit) produced by Sun Systems consisting of DOS command line tools, providing the basic level of development environments, to full visual development environments such as Symantec VisualCafe, Kawa or Microsoft Visual J++. Each has its own strengths and weaknesses.

These visual development tools offer features like a visual development interface for quickly creating and modifying a user interface, file and project management tools, online help, compiler control etc. They can greatly improve the speed of application development once the author is familiar with the interface. The more sophisticated of these tools also support the development and use of Java Beans. Most also have a set of standard templates and wizards to aid in the start up of new classes.

See [Appendix C Section 4.5](#) for links to reviews of a number of Java development tools.

4.6. VRML

VRML is a mark-up language, so it is possible to hand code a VRML “world” (*.wrl file extension). However for anything but the most simple of “worlds”, consisting of more than a few objects, this can quickly become difficult. There are a number of tools that allow an author to construct a VRML “world” either through drawing and positioning 3D objects in a similar way to a 2D drawing package, or from transforming data into a VRML representation.

Some tools allow the import and export of different 3D formats. Some tools are explicitly conversion tools, where others are 3D graphics tools that allow the generation of the 3D graphics, such as Iris Explorer which is specifically designed for 3D data visualisation and animation and allows the results to be exported in VRML format.

A VRML viewer is needed to be able to view a VRML “world”. There are a number of plug-ins for browsers that are required for on-line viewing, such as CosmoPlayer and WorldView.

There are a few VRML parsers that can be used to check the format of a VRML file.

See [Appendix C Section 4.6](#) for links to VRML tools.

4.7. DHTML

As has been stated earlier DHTML is a combination of other technologies, HTML, JavaScript, DOM and CSS. As this is the case, no one tool can be said to be a DHTML tool. However a number of HTML tools market themselves as supporting the development of DHTML in some way. Particularly marketed in this way is Macromedia’s Dreamweaver.

See [Appendix C Section 4.7](#) for links to reviews of a number of these.

4.8. ActiveX

ActiveX components are not written with a specific tool, rather they may be written in any language as long as they use the ActiveX API.

- Visual Basic. The Microsoft Visual Basic language is well suited to ActiveX programming, as it directly supports the ActiveX API and has a wizard for creating new components.
- C++. C++ may be used to write ActiveX controls, and Microsoft Visual C++ is probably the best environment to do this is with, as it supports the ActiveX API directly, and provides such tools as the ActiveX test container for viewing the component..
- Java. ActiveX components may also be written in what could be termed their direct competitor, Java. Normally Java is platform independent, but by tying itself to the ActiveX API it will only run on Microsoft Windows platforms or those that have extension applications allowing the control (the term for an ActiveX component) to run.

ActiveX is strongly tied to the Microsoft Windows platform, and Microsoft products in general. This platform dependence should be borne in mind when developing interactivity for Web pages.

See [Appendix C Section 4.8](#) for links to ActiveX tools.

5. THE FUTURE

The pace of development of the technologies supporting the World Wide Web and interactivity continues unabated. Cutting-edge technology, supported by only the most sophisticated tools, quickly becomes the accepted standard. Presented here are some of the “On the Horizon” developing technologies. Some are already close to implementation.

XML - Extensible Mark-up Language

The Extensible Mark-up Language (XML) is a subset of SGML (Standard Generalised Mark-up Language), an international standard for defining the structure and content of electronic documents. Its goal is to enable generic SGML to be served, received, and processed on the Web in the way that is now possible with HTML. XML is already being used in some applications, but is generally not implemented on the Web.

SGML is a meta-language, i.e. a set of generalised rules used to write other more specific mark-up languages. For instance HTML is a mark-up language created and defined using SGML. It is used primarily for textual interchange, but can call upon other data types. SGML is an excellent at the structuring and storage of data and is completely extensible but is hard to implement fully for network use, such as with the Web. This is a shame, as SGML allows for many options to be used that are closed to HTML.

SGML is non-proprietary and is system- and platform- independent, so SGML documents created on obsolete computer systems can be transferred easily to other systems. However, the software implementation of SGML is difficult, due to its richness. It is difficult to encompass all of the options available to SGML in one application.

To deal with this XML has been developed as a subset of SGML. XML only specifies the parts of SGML that are used by the majority of SGML users. This selection makes XML more flexible and easily deliverable over a network such as the Web. XML is a meta language that can be used to specify a Mark-up Language, i.e. a Document Type Definition (DTD), which can then be used to define the Web page. Once the data held by an XML document is downloaded it is up to the client application how to use the DTD to display the data contained.

XML has many options unavailable to HTML, such as multi-directional links, and combined with the use of Java makes nearly any functionality possible - in fact Jon Bosak, the chair of the World Wide Web Consortium editorial review board, believes that “XML gives Java something to do”.

XML is a very large topic, and it is difficult to distil down into a few paragraphs. However, it is set to become the web language of the future, and its power in the manipulation and storage of data, particularly when combined with the platform independence and power of Java, make it have a strong potential for use in the work of the NMS. Already there are a lot of support and a number of applications available for the design of XML “solutions”.

See [Appendix C Section 5.1](#) for links to XML information.

Synchronised Multimedia - SMIL

To enable simple authoring of TV-like multimedia presentations such as training courses on the Web, W3C has designed the Synchronized Multimedia Integration Language, pronounced “smile”). The SMIL language is an easy-to-learn HTML-like language. Thus, SMIL presentations can be written using a text-editor. A SMIL presentation can be composed of streaming audio, streaming video, images, text or any other media type.

The proposed SMIL standard (again by the W3C) defines an XML-based language that allows control over the positioning and timing of elements in a multimedia presentation using a simple, markup language similar to HTML.

In a SMIL presentation, the media elements are referenced from the SMIL file, in a similar manner to the way HTML pages reference images, applets etc. The plain-text nature of the SMIL file means that it is easy to create and to edit, and can be assembled on-the-fly by Java servlets or CGI scripts as can an HTML page. It is also a relatively low bandwidth way to provide multimedia compared to streaming images and text as redundant frames of encoded video, as the images can be downloaded once and reused wherever necessary, rather than being discarded once viewed.

There are already applications that support SMIL, including Microsoft Internet Explorer 5.0. See [Appendix C Section 5.1](#) for links to SMIL information.

SVG - Scaleable Vector Graphics

The W3C have released a working draft document on SVG for consultation. SVG is a language for describing two-dimensional graphics in XML. SVG allows for three types of graphic objects: vector graphic shapes (e.g. shapes made up of straight lines and curves), images and text. Graphical objects can be grouped, styled, transformed and composited, so that a complex object can be produced from a set of simpler ones.

SVG drawings can be dynamic and interactive, and will become part of the DHTML concept, or whatever supersedes it with the advent of XML. The Document Object Model (DOM) for SVG allows simple vector graphics animation via scripting, using a rich set of event handlers such as “onmouseover” and “onclick” which can be assigned to any SVG graphical object. This makes the task of using graphics in interactive applications significantly more straightforward and direct. In addition SVG significantly reduces the communications and data storage overhead of interactive applications.

See [Appendix C Section 5.1](#) for links to SVG information.

Immersive Virtual Reality

With the rapid development of desk top processor power and communications technology, at some point “immersive virtual reality” will become possible through the Web, most likely developing out of the present VRML standard. Immersive VR is 3D modelling taken a step further than VRML viewers do at present, exposing the user’s senses to the model, to the partial exclusion of “reality”, such as the use of a head-set to display stereoscopic real time interactive graphics.

While this sounds a bit “science fiction” many research establishments are already developing and using such systems, and many commercial games already use such technology. Holding it back from becoming widely available as an interface is the processing power, data storage and communication rates required to support the systems. At the moment only the top-end graphics servers are capable of the processing speed required. However, it is only a matter of time before PCs become sufficiently powerful to use it.

The full potential of the use of such systems in the visualisation and manipulation of data, whether from experiment or from models, cannot really be imagined as it is still an immature technology, particularly when the use of 3D or higher dimensional projections onto 2D displays is only now being researched and developed. Investigation of the scope for applying the technology to the NMS could make a useful forward looking topic for the next SSfM programme.

See [Appendix C Section 5.1](#) for links to information on this topic.

Digital Signatures

Security is becoming more and more important when using the Internet. One issue is the security of information being transferred from point to point, as it is quite straightforward to “listen” in to data. This is being dealt with using encryption technology directly applied to the data.

Another issue is trust. How do you know to trust the information, or perhaps more importantly the software, that you have just downloaded? There is every possibility that it will contain a malign virus, or be otherwise corrupted in some way.

The solution being developed is that of digital signatures, where an electronic “signature” is sent as part of the original message, so that the recipient knows first who sent it, but also whether the message has been changed since it was sent.

This is done by calculating a mathematical value from the source message and encrypting it. This is the digital signature. When the message is received the signature is decrypted using the public encryption key and the resulting value is compared with a value calculated from the received message. If they are the same then the message is genuine and un-changed, so you can trust it as much as you can trust the source.

Various versions of this technology are already implemented and in use, but there is no accepted standard for the transfer of data, which means the use of digital signatures is still restricted to those users who use a similar system by agreement. Digital signature technology could be very important to the NMS in such areas as remote calibration.

See [Appendix C Section 5.1](#) for links to information about digital signatures.

6. CONCLUSIONS

The past few years have done nothing to slow the pace of the development of Information Technology, quite the opposite. Hardware has become much more powerful, particularly on the desktop, software and development tools more sophisticated. Perhaps most significantly the use of the World Wide Web has, by any measurement, exploded as a means to disseminate information. It has perhaps become the most powerful way to spread knowledge now available in terms of access, size of audience, speed and presentation.

While the Internet has been a means of scientific communication almost since its inception, it now has capabilities far beyond what has been possible. For instance, not too long ago, documents were constrained to text, and images needed to be downloaded and viewed off-line. Now images are almost an expected part of any Web page, and whole movies, sound tracks and applications may be run directly through the interface software.

This opens a huge arena of possibilities to the Web author. Documents now no longer have to be static, single view presentations. They can be enhanced using multimedia and interactive technologies, so that the information they contain becomes dynamic and interactive; the reader can explore the information in their own way more and more, allowing a single presentation to have many paths through it, each specific to the reader.

Systems can be demonstrated and manipulated through interactive models, data explored through sophisticated visualisations, data accessed through sophisticated interfaces, and entire worlds explored in 3D, to name but a few. The boundaries of what is possible have expanded to the point where what *is* done is but a small part of what *could* be done. However the boundaries are still expanding. With the continuing development of the technologies supporting the Web it is difficult to predict what will be possible, certainly without sounding like an episode of Science Fiction.

It is fair to say that the NMS lags behind some other fields in seizing the opportunities made available to it by interactivity on the web. More attention will need to be paid to introducing the application of such technology into the formulation of NMS programmes.

APPENDIX A - GLOSSARY OF TERMS

Term	Definition
ActiveX	A standard interface for software components that may be “plugged” into Microsoft or other ActiveX enabled products. They may be referenced from web pages to run in a web page.
API - Application Programming Interface	A series of functions that programs can use to make the operating system do their “dirty work”. Using Windows APIs, for example, a program can open windows, files, and message boxes.
AVI - Audio Visual Interleave	A multimedia file format.
Applet, Java	A program written in the Java in a format that may be referenced from a web page and is runnable by browsers implementing the Java Virtual Machine.
Bean, Java	A Java class written to be compliant with the Java Beans model, so that it may be used as a “plug-in” component to construct a larger application.
Bean Box	An application allowing a user to plug Java Beans together to produce an application or applet.
Browser	An application that can navigate the Internet and interpret HTML files to display web pages.
C and C++	Similar high level programming languages (C is a subset of C++).
Cache	Fast memory used to temporarily store data or instructions that are likely to be used again in the near future, increasing the speed of the computer.
CGI - Common Gateway Interface	This is a standard of communications between applications and web servers allowing dynamic response to and construction of web pages.
CGI Script	When a program is written for CGI is known as a “CGI script” as a common term. This is because most programs written for CGI are “script” languages that do not need compilation, e.g. PERL.
Client	An application or system (e.g. a web browser) that may seek some service from a “server” application or program (e.g. a web server). It may be used to refer to the client software OR hardware depending on context. See Server.
CSS - Cascading Style Sheets	The technology that allows formatting a web page independently of the document’s structure.
DHTML - Dynamic HTML	The name for a collection of technologies that may be used to make a web page change its appearance after downloading. The technologies are HTML, JavaScript, CSS and DOM.

DOM - Document Object Model	The technology that describes a web document in terms of object-oriented objects which may be accessed via a scripting language allowing the document to be manipulated after downloading.
Freeware	Software that is supplied free of charge. This does not necessarily mean it is copyright free, or that there is no licence agreement for its use. See Shareware.
FTP - File Transfer Protocol	A communications protocol for the communication of a file in either ascii or binary formats across a network.
GIF - Graphics Interchange Format	A very popular image file format used on the Internet. May be used to produce animations.
HTML - Hypertext Mark-up Language	The language used to define and structure the content of a web page.
HTTP - Hypertext Transfer Protocol	A communications protocol designed for the communication of HTML files across a network. It is versatile enough to be used to communicate any file.
Hypertext	Text that is marked so that it provides a link to another page of text or bookmark.
Immersive Virtual Reality	Virtual Reality (VR) is the term applied to a simulation representing a complete "world". Immersive VR is where the interface to that world is sophisticated enough to give the user the impression that they are "there" in some sense.
Interactive	Where the system responds to the actions of the user.
Internet	A widely distributed international network of interconnected servers.
ISP - Internet Service Provider	An ISP provides a link into the Internet for a client machine.
Java	Platform independent programming language that can be used to write "applets" or applications.
JavaScript	A scripting language that may be embedded in a web page. Regarded as one of the key components of DHTML
Java Virtual Machine (Java VM)	An application that interprets compiled Java code and runs the instructions locally. Each specific platform must have their own Java VM so that the code is translated correctly for it.
JPEG - Joint Photographic Experts Group	A very popular image file format used on the Internet. Used mostly for photographic quality images.
Local	The term used to describe activity on the users own computer. (See Remote)
MIDI - Musical Instrument Digital Interface	A music file format.

MP3	MPEG audio compression standard layer 3 for Stereo quality sound.
MPEG - Moving Pictures Expert Group	A set of standards for audio and video compression.
PERL	A scripting language, very powerful at text handling. The most common language to write CGI applications for the web.
PNG - Portable Network Graphics	A relatively new file format for graphics being promoted and gaining support for internet use.
QuickTime, Apple	A multimedia file format for movies and sound.
Remote	Activity that takes place on a computer other than the user's one. See Local.
Server	An application or system that provides a service for client applications or systems. It can be used to refer to the server software OR the server hardware depending on context. See Client.
SGML - Standard Generalised Mark-up Language	A language used to define mark-up languages for actual use. This was used to define HTML. SGML is a superset of XML.
Shareware	Software that may be distributed and trialed freely, but a full licence must be purchased for further use.
SMIL - Synchronized Multimedia Integration Language	An XML based language for structuring multimedia content on a web page.
SSfM - Software Support for Metrology	A DTI funded NMS programme providing generic support to the NMS in software and mathematics.
SVG - Scalar Vector Graphics	A format for defining graphics as a set of simple objects which may in turn be defined in a text file, making them very much smaller than image files.
URL - Universal Resource Locator	Effectively an address for web pages and files.
User	A person using an application.
VBScript	A scripting language based on Microsoft Visual Basic that may be embedded into web pages and run by VBScript compliant browsers.
Visual Basic, Microsoft	A programming language and visual development interface.
VRML - Virtual Reality Modelling Language	A standard and language used to define 3D Virtual Reality "worlds" for display in VRML viewers.
W3C - the World Wide Web Consortium	The W3C is an international industry consortium founded in October 1994 to "lead the World Wide Web to its full potential by developing common protocols that promote its evolution and ensure its interoperability."
WAV - Waveform Audio	An audio file format.
Web page	A page on the World Wide Web.

Web page application	A program embedded in a web page that can be run by the browser. E.g. a Java applet, JavaScript or VBScript.
WWW - the World Wide Web	The network of web pages on the Internet. Not all of the Internet is technically part of the World Wide Web, as some parts use different communication protocols. However in practice “WWW” is often used synonymously with “Intenet”.
WYSIWYG - What You See Is What You Get	A term for editing applications where what you see on the screen at edit is what you get in use (e.g. when printed out, or viewed as a web page)
XML - eXtensible Mark-up Language	A language that is used to define mark-up languages. It is a subset of SGML which was used to define HTML.

APPENDIX B - HTML GOOD PRACTICE GUIDE

This section examines good practice guidelines for the development of interactive web pages, both the web page itself and its components.

The guidelines set out here come from three principal sources:

- “Standards for HTML Authoring for the World Wide Web” WDG 1997,
- “Web Content Accessibility Guidelines” W3C 1999,
- The author’s experience.

The guidelines are specifically relevant to the NMS and interactive web pages. For the URL links to the full documents see [Appendix C Section 2.1](#).

In the table below the first column identifies the area to which guidance applies, the key is as follows:

V	validation and testing
D	document information
C	colours and backgrounds
I	images
A	applications
S	structure
C	cascading style sheets
F	frames
G	general

The second column is the specific guideline and the final column contains general background notes.

	Guideline	Background note
V1	All HTML documents should be checked with an HTML validator and validated successfully.	<i>Validation is probably the easiest and most important thing an author can do to ensure that an HTML document is accessible and usable by as wide an audience as possible. Validation tools will check the syntax of the HTML used against the HTML version document type definition, ensuring that there are no authoring errors. This can be crucial, as even if the browser is able to recover, which most of the time they will, the manner in which they do this tends to differ.</i> <i>In addition, validation will identify the exact parts of the document that are non-compliant, and the required documentation of this will provide valuable information to the user if they are not initially able to access the component. This identification of non-compliance can often also provide valuable insight into ways to expand the accessibility of the page.</i>

V2	HTML documents should be tested by being viewed in several browsers, especially those expected to be used by the client audience.	<i>Different browsers implement HTML and recover from errors in different ways. Browsers also have a range of proprietary extensions or higher versions of HTML available to them, which, if used, can make the appearance of those features unpredictable in other browsers.</i> <i>It is therefore necessary that documents not only be validated, but they also be viewed in a range of browsers, including all those reasonably expected to be used by the client audience.</i>
V3	Script pages should be viewed in browsers not supporting the scripting language to test satisfactory appearance.	
D1	All HTML documents should include an appropriate title.	<i>All HTML documents should have a clear and relevant title defined in the HEAD section of the code.</i>
D2	Documents that are a “front page” or other principal entry point for a system should include the following: KEYWORDS and DESCRIPTION meta elements for the benefit of web indexers.	<i>Documents that are principal entry points for a system should use meta tags with the DESCRIPTION and KEYWORDS elements. This allows indexers (such as search engines) to identify correctly and categorise the site and its content.</i> <i>The author or maintainer of the system should be identified clearly and linked to from such a page, allowing direct feedback from users. This will benefit both user and maintainer, as bugs will be quickly identified and user satisfaction enhanced</i>
D3	All documents should clearly display an “expiry date” for the page to inform the user of its continued relevance.	
D4	All HTML documents should include the date of the last change to the document.	
D5	Web pages should contain version control and update information, along with an expiry date for use by both the webmaster and the user.	<i>This allows the user to assess how up to date the information is and keeps the webmaster informed of updates that are required as time passes.</i> <i>It is possible to insert a tag into a web page that will force a browser to revisit a page to read the page fresh from the remote server rather than merely reload it from cache after some date or time has passed. This provides automatic version control for the user.</i>
C1	Pages should not depend on a particular browser window, font size or colour table to be readable, or any visual presentation whatsoever, except where the information is inherently visual in nature.	<i>Colour can be used for emphasis and also makes information easier to absorb quickly, though too many colours make it more confusing. In general two or three colours and two or three fonts should be used.</i>

C2	Where colours are set by an author, they should ensure a strong contrast between text and background. This implies that an author sets an appropriate background if the foreground is set, and vice versa.	<i>The content of a page should not be obscured by inappropriate choice of foreground and background colours. There should be an enforced high contrast between these, so that any and all information is clearly visible. Also the use of a “conspicuous” background, particularly a very colourful one can make reading text very difficult.</i>
C3	Background images (where used) should be small, and of a similar colour to the background colour specified.	
C4	Conspicuous background images should be avoided in pages containing text.	
I1	Images should not be used to replace text, they should complement it.	<i>The use of images needs care. Some browsers do not load images, either because they are text based, or because they have the image loading turned off to speed upload of documents. In the second case users often only load those images they decide that they need to see. This is most often true of users accessing the internet via low speed modems.</i>
I2	All images should have ALT texts.	<i>All images, therefore, should try to supplement text, rather than replace it, unless the information is inherently visual. In addition, brief relevant alternative text should be provided for each image, allowing users to select those images they wish to download.</i>
I3	ALT texts for images which are also Hyperlinks should be descriptive of the purpose of the link.	<i>“Home”, “Next”, “Previous” are examples of good ALT texts; “Click Here”, “Home Icon”, “Binocular Icon” are not.</i>
I4	ALT texts for larger images (25Kb +) should warn of their size.	<i>For instance “Image of Equipment (34Kb)”.</i> <i>Some images are, by necessity, large and can take significant amounts of time to download. When the image size is larger than 10Kb, the size of the image should be included in the alternative text. This allows the user to decide whether they wish to wait to download it based on its size.</i> <i>It is possible to write a page to pre-load images that are likely to be used soon. Use of the time a user takes viewing a page to load images for following pages makes access time appear faster.</i>
I5	Where imagemaps are used, alternative means of navigation should be made available to readers.	<i>When an image is used as an image map for navigation, on either server or client side, there should be an alternative means of navigating the same links provided on the page.</i>
I6	Images should use height and width attributes.	<i>The size of the image should also be declared, allowing the browser to format the document correctly with respect to the image.</i>

A1	All applets and other programmatic objects should have alternative texts, either through the ALT attribute of , or within the content of <APPLET> or the content of <OBJECT>.	<i>The text should describe briefly the purpose of the object.</i>
A2	Client side scripting, such as JavaScript, may be used provided it does not detract from the page's accessibility to browsers not supporting or enabling this feature.	
S1	<BLINK> should not be used.	
S2	 should not be used in place of <H1> to <H6>.	
S3	Emphasis tags such as , or should not be applied to extended passages.	
S4	<H1>...</H1> should be used exactly once in an HTML page.	
S5	HTML documents should not be dependent on proprietary extension mark-up. All key functionality should be available to an HTML compliant browser not supporting the extension mark-up.	
S6	HTML containers, such as <P> (paragraphs) or <TABLE>, should be explicitly closed.	
C1	Authors may use style sheets to enhance web pages, and are encouraged to do so when seeking to determine document appearance.	
C2	Style sheets should not be visible to browsers that do not support them.	<i>Use <!-- --> to hide them.</i>
F1	Information made accessible through a frame-set should also be made accessible unframed by the NOFRAMES section.	<i>The NOFRAMES section should provide readers with a complete alternative.</i>
F2	Use of more than one frame-set based layout on a site should be avoided.	

F3	All external Hyperlinks in a frame-set should use the target attribute to avoid embedding another site in a frame.	<i>It should be clear that another site has been entered.</i>
G1	Web pages should be laid out clearly with a minimum of 'clutter'.	<i>It is easy to put too much onto each page, making it confusing to view and often slow to load. Many users are put off by this whatever the page's content.</i> <i>The pages should be split up as much as possible into sub-sections, each expressing one particular concept with links to associated pages. The non-linearity of such a structure also allows the user to browse through a site in an effective user-centred manner.</i>
G2	Organise the files for a web site into a directory tree mirroring the desired structure of the site.	<i>This allows for easier site management and expansion.</i>
G3	The 'entrance' page for a site, or a large sub-section of it, should be simple and fast loading, holding the minimum to allow a user to recognise and link to the content they wish to access.	<i>This page is often called the index page. Many users are frustrated by slow loading pages they do not regard as having significant content. Frame sets should generally be kept simple, with as little simultaneous page loading as possible and it should be borne in mind that applets require the Java Virtual Machine to load and run, which can take several seconds.</i>
G4	A consistent convention of navigation between web pages and/or sections of a site should be implemented.	<i>Making the user familiar with one set of navigation 'principles' for the site will allow easier and more efficient access to the information it contains. Use of a navigation menu in a set place on each page (generally the left 'margin'), perhaps using a frame set, is a convention that is often used.</i>
G5	Each section of a site should have an easily identifiable point of contact for the user.	<i>In general this will be the webmaster for the site, but in the case of large sites representing a number of different groups within a single organisation this is likely to be one of a number of individuals. A link is best placed on the navigation bar, either directly to the person via e-mail, or to a contact page, with the e-mail link or a reply form on it.</i>

APPENDIX C - USEFUL RESOURCES AND BIBLIOGRAPHY

Links to useful resources on the Web are categorised by the section they relate to.

All links were active at the time of writing, but any may cease to be available at any time, and NPL has no responsibility nor control of the content or validity of any web pages presented here outside its own web site. This is only a guide to possible sources of information.

1. INTRODUCTION

1.1. SSfM Project 6.2 Description

The National Physical Laboratory Software Support for Metrology home page	http://www.npl.co.uk/ssfm/	
---	---	--

1.5 Browsers

See Web Browser tools		
Yahoo's "Browser Wars" section	http://headlines.yahoo.com/Full_Coverage/Tech/Browser_Wars/	
ZDNET's product guide to browsers	http://www.zdnet.com/products/internetuser/browsers.html	
BrowserCaps information on browser capabilities	http://www.browsercaps.com/picker.ptx	

2. INTERACTIVE WEB PAGES - THE TECHNOLOGY

2.1. Hypertext - HTML

The home of HTML at the World Wide Web Consortium (W3C)	http://www.w3.org/MarkUp/	
A library of on-line texts	http://www.itlibrary.com/reference/dir.webse rvices.html#html1.html	
A good site with lots of web resources, HTML and others	http://wdvl.com/Authoring/HTML/	
A basic HTML tutorial	http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimer.html	
A good basic tutorial	http://www.ncsa.uiuc.edu/General/Internet/WWW/HTMLPrimerP2.html	
An excellent HTML language resource	http://www.blooberry.com/html/	
Index of HTML tags	http://www.willcam.com/cmat/html/crossref.html	
A stripped down tag list	http://werbach.com/barebones/download.html	
Reference material on the use of frames	http://www.htmlhelp.com/design/frames/	

Authoring standards document by the Web Design Group	http://www.htmlhelp.com/design/standards.html	
Accessibility guidelines document by W3C - select current draft.	http://www.w3.org/WAI/GL/	
Accessibility guidelines document by the Web Design Group	http://www.htmlhelp.com/design/accessibility/	
HTML validation service by W3C	http://validator.w3.org/	
HTML analysis tool	http://www2.imagiware.com/RxHTML/	
HTML Authoring Reference materials	http://www.htmlhelp.com/	
Very good guidance on web page design	http://info.med.yale.edu/caim/manual/pages/page_design.html	
Very good guidance on site design	http://info.med.yale.edu/caim/manual/sites/site_design.html	
Discussion on maths typesetting on web pages.	http://world.std.com/~wij/web-design/typesetting.html	
W3C's recommendation for a Math Mark-up Language for use on the Web	http://www.w3.org/Math/	
MathML viewable browser	http://www.icesoft.no/DOWNLOAD/ELite.html	
WebEQ applet for viewing equations	http://www.webeq.com/webeq/	

2.2. Multimedia

A discussion with examples of what image format to use, focused on Adobe's own product, Photoshop.	http://www.adobe.com/studio/tipstechniques/GIFJPGchart/main.html	
Information about the JPEG format.	http://www.cis.ohio-state.edu/hypertext/faq/usenet/jpeg-faq/top.html	
Information about the GIF format	http://members.aol.com/royalef/gifabout.htm	
Guidance on the use of graphics.	http://info.med.yale.edu/caim/manual/graphics/cs/graphics.html	
Guidance on the use of video and audio	http://info.med.yale.edu/caim/manual/multimedia/introduction.html	
JavaScript and Sound - a Tutorial	http://www.hotwired.com/webmonkey/97/50/index3a.html	

2.3. Client Side Scripting

Introduction to JavaScript - a Tutorial	http://rummelplatz.uni-mannheim.de/%7Eskoch/js/script.htm	
JavaScript for the Total Non-Programmer - a Tutorial	http://www.webteacher.com/javatour/frameh01.htm	
Java Script reference from Netscape	http://developer.netscape.com/docs/manuals/communicator/jsref/index.htm	Navigator specific information
Miscrosoft VBScript documentation	http://msdn.microsoft.com/scripting/default.htm?scripting/vbscript/	

2.4. Server Side Scripting

Script references

General CGI reference material	http://hoohoo.ncsa.uiuc.edu/cgi/	
The CGI Resource Index	http://www.cgi-resources.com/	
PERL Tutorial	http://www.ncsa.uiuc.edu/General/Training/PerlIntro/	
CGI tutorial (PERL)	http://www.cgi101.com/	
CGI tutorial (C++)	http://hjs.geol.uib.no/Cplusplus/	
CGI tutorial (C++)	http://monarch.papillion.ne.us/~stthomas/cgi_in_c.html	
CGI Tutorial (Visual Basic)	http://www.users.dircon.co.uk/~tbrown/vb/cgi.htm	
Writing Server side JavaScripts from Netscape	http://developer.netscape.com/docs/manuals/enterprise/wrijsap/index.htm	

Search Engines and Guides

Search Engine	http://www.altavista.com/	
Search Engine	http://www.lycos.co.uk/	
Search Engine	http://www.hotbot.com/	
Search Engine	http://www.webcrawler.com/	
Search Engine	http://www.excite.com/	
Huge Subject Guide	http://www.yahoo.com/	
Meta search engine - simultaneously uses 7 search engine	http://www.debriefing.com/	

2.5. Java

Sun Systems own Java programming tutorial pages for Java	http://java.sun.com/docs/books/tutorial/index.html	
Sun Microsystems developer home page	http://developer.java.sun.com/	
Sun Microsystems Java Beans home page	http://java.sun.com/beans/index.html	
Good Java tutorial	http://java.sun.com/docs/books/tutorial/	
Extensive set of Java resource links	http://www.apl.jhu.edu/~hall/java/	

2.6. VRML

NPL Centre for Materials Metrology example web site.	http://sulu.npl.co.uk/netshare/guest/welcome1.html	
A number of VRML objects created from Maple	http://www.cybermath.com/samples.html	
Planet9 present a range of VRML worlds they have created	http://www.planet9.com/indexie.htm	
A gallery of machined parts modelled in VRML	http://www.cosmosm.com/vrml_gal.htm	
Galley of VRML worlds from NIST.	http://vislab.cam.nist.gov/~lipman/vrml/vrml_other.html	

2.7. Dynamic HTML

General DHTML reference and resources from Netscape	http://developer.netscape.com/tech/dynhtml/resources.html	
General DHTML reference	http://www.htmlstylehouse.com/dhtml.htm	
DHTML tutorial	http://www.hotwired.com/webmonkey/dynamic_html/tutorials/tutorial1.html	
Client Side Scripting	See section above	
Document Object Model	http://www.w3.org/DOM/	
Cascading Style Sheets	http://www.w3.org/Style/CSS/	
Cascading Style Sheets - resources	http://www.htmlhelp.com/reference/css/	
Cascading Style Sheets - a tutorial	http://www.hotwired.com/webmonkey/stylesheets/tutorials/tutorial1.html	

2.8. ActiveX

General ActiveX resource	http://awdsites.com/library/index.htm	
General ActiveX resource	http://www.shorrock.u-net.com/index.html	

3. INTERACTIVE WEB PAGES - APPLICATIONS

3.1. Data Transformations

SSfM Examples of Interactive Web pages	http://www.npl.co.uk/ssfm/tt/index.html	
A “focal point for information on numerical computing in Java”.	http://math.nist.gov/javanumerics/	
GAMS, the Guide to Available Mathematical Software, is a cross-index and virtual repository of mathematical and statistical software useful in science and engineering.	http://math.nist.gov/HotGAMS/	
General resource of java applets and applications	http://www.jars.com/	

3.2. Charts and Graphs

ptolemy plot - a Java library of graphing tools and demonstrations	http://ptolemy.eecs.berkeley.edu/java/	
GAMS, the Guide to Available Mathematical Software, is a cross-index and virtual repository of mathematical and statistical software useful in science and engineering.	http://math.nist.gov/HotGAMS/	
General resource of java applets and applications	http://www.jars.com/	

3.3. Interactive Models

SSfM Examples of Interactive Web pages	http://www.npl.co.uk/ssfm/tt/index.html	
NPL Materials interactive site	http://sulu.npl.co.uk/netshare/guest/	
A simulation of real gas flows with various physical processes	http://www.csa.ru/~greg/tryJava/DSMC/	

A Simulation of the detection of planets around other stars	http://zebu.uoregon.edu/nsf/wobble.html#BB	
A Virtual Wind Tunnel	http://raphael.mit.edu/Java/	
“Virtual Lego” as an example of a “virtual world” allowing the user to practice perform a task.	http://www.pncl.co.uk/~fdanbury/JVL.html	
General resource of java applets and applications	http://www.jars.com/	

3.5. Database Queries

the NOAA Environmental Services Data Directory	http://www.esdim.noaa.gov/NOAA-Catalog/	
Physical Reference Data	http://physics.nist.gov/PhysRefData/contents.html	
Databases for Atomic and Plasma Physics	http://plasma-gate.weizmann.ac.il/DBfAPP.html	
Matweb online materials information database	http://www.matweb.com/	

3.6. Miscellaneous Web Page Enhancements

Volcano watch animations from sets of images.	http://www.ssec.wisc.edu/data/volcano.html	An example of use
General resource of java applets and applications	http://www.jars.com/	A resource of many, many applets etc.

4. TOOLS FOR BUILDING INTERACTIVE WEB PAGES

Web Browsers

Netscape Navigator	http://www.netscape.com/	
Microsoft Internet Explorer	http://www.microsoft.com/windows/ie/	
NCSA Mosaic	http://www.ncsa.uiuc.edu/SDG/Software/MacMosaic/MacMosaicHome.html	
Opera	http://www.operasoftware.com/download.html	
Browser Capabilities	http://www.browsercaps.com/picker.ptx	Resource on browser HTML compatability

4.1. Hypertext - HTML

HTML code editors:

HotDog (Sausage Software)	http://www.sausage.com/	
HomeSite (Allaire)	http://www.allaire.com/	
Arachnophilia	http://www.arachnoid.com/arachnophilia/	freeware
GhostHTML	http://mirrors.kasnet.com/tucows/adnload/dlghosthtml.html	freeware
Misc. freeware selection	http://www.freesitertools.com/editors.html	Various freeware editors for download
Misc. selection	http://mirrors.kasnet.com/tucows/htmledit95.html	
Any text editor	n/a	

WYSIWYG Editors

Microsoft FrontPage	http://www.microsoft.com	
Netscape Composer	http://www.netscape.com	Free with Netscape Communicator
LiquidFX	http://www.psylon.com/	
Misc. Selection	http://mirrors.kasnet.com/tucows/htmledit95.html	

Hybrid Editors

HotMetal Pro	http://www.sq.com/	
Macromedia Dreamweaver	http://www.macromedia.com/software/dreamweaver/	
Misc. Selection	http://mirrors.kasnet.com/tucows/htmledit95.html	

4.2. Multimedia

Macromedia - Flash, Director, Shockwave, Deamweaver	http://www.macromedia.com/	A range of applications for multi-media authoring and viewing.
Apple Quicktime	http://www.apple.com/quicktime/	
Adobe Photoshop	http://www.adobe.com/prodindex/photoshop/main.html	One of the most recommended image manipulation tools.
Alchemy Mindworks	http://www.mindworkshop.com/alchemy/alchemy1.html	A number of image processing and viewing tools
Web Image, Group42	http://www.group42.com/webimage.htm	Image Manipulation

Microsoft Media Player for Windows and Mac	http://www.adobe.com/prodindex/photoshop/main.html	mp3, au, wav, qt, mov,
mpg123 for UNIX	http://dorifer.heim3.tu-clausthal.de/~olli/mpg123/	
WinAmp player for Windows	http://www.winamp.com/	
MacAmp player for Apple Macintosh	http://www.macamp.com/	
Various graphics tutorials on Animation, Photoshop, and others	http://www.freegraphics.com/10_Tutorials/	
Adobe premiere	http://www.adobe.com/prodindex/premiere/main.html	Video encoder, with support for import/export to/from many formats.
Scour resources	http://www.scour.net/resources/	Resources for multimedia tools, but type or file format

4.4. Server Side Scripting

The CGI Resource Index	http://www.cgi-resources.com/	Very comprehensive
PERL CGI programming library	http://cgi-lib.stanford.edu/cgi-lib/	Bills itself as the “standard” PERL library for CGI.

4.5. Java

Sun Microsystems Java Development Kits (JDK)	http://www.sun.com/java/products/platform.html -	freeware PC, Mac, UNIX
Sun Microsystems Java Bean Development Kit (BDK).	http://java.sun.com/beans/software/bdkdownload.html -	freeware PC, Mac, UNIX
Sun Microsystems Bean development tools link page	http://java.sun.com/beans/tools.html	
Microsoft Visual J++ home page	http://msdn.microsoft.com/visualj/	PC
Symantec Visual Café	http://www.symantec.com/	PC, Mac
IBM VisualAge for Java	http://www.software.ibm.com/ad/vajava/	PC
Kawa	http://www.tek-tools.com/	PC

4.6. VRML

Cosmo Software - VRML tools	http://cosmosoftware.com/	Particularly Cosmo Player (VRML viewer),
InterVista WorldView	http://www.intervista.com/worldview/	

The VRML Repository. General resource for VRML software - Building Tools, Viewers, Converters (called Translators), Parsers, Documentation, Libraries etc.	http://www.sdsc.edu/vrml/	Very good set of VRML resources.
Iris Explorer licensed by NAG.	http://www.nag.co.uk/visual/IE/iecbb/Pro duct.html	

4.7. DHTML

Macromedia DreamWeaver	http://www.macromedia.com/	
------------------------	---	--

4.8. ActiveX

Source of ActiveX controls	http://www.download.com/PC/Activex/	
Microsoft	http://www.misrosoft.com/	Use the search tool

5. THE FUTURE

SVG from the W3C.	http://www.w3.org/Graphics/	
Learn SMIL	http://www.empirenet.com/~joseram/index.html	
SMIL at HELIO	http://www.helio.org/products/smil/	
XML from the W3C or XML in 10 points from the W3C	http://www.w3.org/TR/REC-xml http://www.w3.org/xml/1999/xml-in-10-points	
XML, an introduction from Webmonkey.	http://www.hotwired.com/webmonkey/98/41/index1a.html	
XML, a document by Richard Lander.	http://www.csclub.uwaterloo.ca/u/relander/XML/Wave/index.html	
XML, a document by Robin Cover.	http://www.oasis-open.org/cover/xml.html	

APPENDICIES

Glossary

Glossary of terms from CNET	http://www.cnet.com/Resources/Info/Glossary/	
Glossary of Academic Information Technology Terms	http://www-rohan.sdsu.edu/glossary.html	
The Matisse Glossary of Internet Terms	http://www.matisse.net/files/glossary.html	
BABEL Glossary of Computer Related Abbreviations	http://www.access.digex.net/~ikind/babel.html	

Interactive Connections Internet Glossary	http://www.interactive.com/ internetglossary.html	Searchable
--	--	------------

HTML Good Practice

The Web Design Group “help” pages.	http://www.htmlhelp.com/	
The World Wide Web Consortium	http://www.w3.org/	

Useful Resources and Bibliography

Indexed source of Java, JavaScript etc.	http://www.jars.com/	
Indexed resources	http://www.developer.com/	
The Home page of the World Wide Web Consortium	http://www.w3.org/	
Source of downloadable software for the web	http://mirrors.kasnet.com/tucows/	
Web Developer’s Virtual Library, extensive resources and tutorials	http://stars.com/	

Books

O’Reilly Series	http://www.ora.com/	An excellent set of reference books
Amazon - on-line Bookshop, search engine and reviews	http://www.amazon.com/	