NPL
National Physical Laboratory

Report

NPL REPORT
DEM-ES 002

Software Support for Metrology – Good Practice Guide No. 14

Guidance and Tools for Interactive Web Pages

S C Gardner

April 2005

dti
National Measurement System

National Physical Laboratory | Hampton Road | Teddington | Middlesex | United Kingdom | TW11 0LW

Switchboard 020 8977 3222 | **NPL Helpline** 020 8943 6880 | **Fax** 020 8943 6458 | **www.npl.co.uk**

dti

The DTI drives our ambition of 'prosperity for all' by working to create the best environment for business success in the UK. We help people and companies become more productive by promoting enterprise, innovation and creativity.

We champion UK business at home and abroad. We invest heavily in world-class science and technology. We protect the rights of working people and consumers. And we stand up for fair and open markets in the UK, Europe and the world.

The National Physical Laboratory is operated on behalf of the DTI by NPL Management Limited, a wholly owned subsidiary of Serco Group plc

Software Support for Metrology
Good Practice Guide No 14

Guidance and Tools for Interactive Web Pages

S C Gardner
Mathematics and Scientific Computing Group

April 2005

ABSTRACT

This guide describes good practice in the use of web technologies for the development of interactive web pages. The emphasis is on technologies appropriate for presentation of scientific content, including mathematics and data visualisation, and on technologies that make content easier to navigate on the web.

© Crown copyright 2005
Reproduced by permission of the Controller of HMSO

ISSN 1744–0475

National Physical Laboratory
Hampton Road, Teddington, Middlesex, United Kingdom. TW11 0LW

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

We gratefully acknowledge the financial support of the UK Department of Trade and Industry (National Measurement System Policy Unit)

Approved on behalf of the Managing Director, NPL,
by Jonathan Williams, Knowledge Leader of the Electrical and Software team

Contents

1. Introduction

This guide is part of a series of good-, and best-, practice guides from the Software Support for Metrology (SS*f*M) programme. The guide describes good practice in the use of web technologies for the development of interactive web pages. The emphasis of the guide is on technologies appropriate for presentation of scientific content, including mathematics and data visualisation, and on technologies that make content easier to navigate on the web.

This guide is a deliverable of the DTI-funded National Measurement System programme on Software Support for Metrology (2004-2007). An earlier version of this document appeared as Guidance and Tools for Interactive Web Pages (NPL Report CISE 41/99) by Alister Cooper – a deliverable of the first SSfM programme (1998–2001). This guide is an updated and revised version of that document, including more detail on the expanded range of web technologies that have become available, and "re-packaged" as an SSfM good-practice guide.

1.1 SS*f*M Web Site Project Description

This guide is a deliverable from the SS*f*M-3 project 5.3: *SSfM and EUROMETROS websites and other delivery mechanisms for SSfM outputs*. This project exists primarily to support the websites established by earlier SS*f*M programmes and to develop other deliverables from the website material.

The SS*f*M website (www.npl.co.uk/ssfm) contains information about the programme and projects, including previous deliverable documents and pages for SS*f*M club members. The EUROMETROS website (www.eurometros.org) is a repository of algorithms and software for metrology – together with test sets and guidance material. The project aims to keep the SS*f*M and EUROMETROS websites up to date. The project also works to promote the websites nationally and internationally, through awareness activities and through using mechanisms to ensure that search engines readily find and prioritise the pages on these websites. The project reports on the use of the websites and has developed mechanisms to get

1.2 Scope – 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 in 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.

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.

1.3 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 Web site operator'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	*Download time* - large programs need not be downloaded. *Run time* - the server may be much more powerful than the client, running "processor heavy" programs faster. *Compatibility* - programs run on the machine they were written for. *Program security* - the program is kept on the server. *Client security* - the program does not run on the client machine.	*Server processing time (1)* - too many clients will cause big reductions in server speed *Server processing time (2)* - the number of clients may need to be limited to allow the program to run at all. *Response time* - communication is done over the Internet, which is very slow compared to internal communication. *Ease of implementation* - the user requires knowledge of and access to the server, for file locations, file permissions, server configuration etc.

Client-side Processing	*Server processing time (1)* - can greatly improve the server's performance when serving a number of clients. *Server processing time (2)* - removes the need to consider limiting client numbers. *Response time* - internal communication is vastly quicker than over the Internet. *Ease of implementation* - client side programs at most require an easily installable plug-in for the client browser to run.	*Download time* - the program must be downloaded - not significant with small programs. *Run time* - the client may not be as powerful as the server, increasing the run time needed. *Compatibility* - no longer a real problem with the development of cross platform technology. *Program security* - the program may be copied. *Client security* - the program runs on the local machine.

Table 1.1 Comparative strengths and weaknesses of client- and server-side processing

As can be seen from Table 1.1, 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.

1.4 Browsers

To view resources 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 download information at the user's request. There are many brands of browser in use, dominated by Microsoft Internet Explorer; others include Firefox, Opera and Netscape.

Internet Explorer tends to set the baseline of functionality in terms of file compatibility, multimedia viewing capabilities, scripting languages (see Section 2.7) and the Java language. However, no single browser has full compatibility with every technology used across the Web; and the different brands of browser have different implementations of the various technologies. Furthermore, within each brand there are differences between successive releases: there is more than one version of Internet Explorer in use, with each release being more sophisticated than the last.

Two major issues arise out of this varying level of support for different technologies that should be borne in mind when authoring web pages:

- The first is platform independence. All through this guidance document, reference will be made to the importance of writing Web pages that are as independent as possible of the platform (browser and operating system). This will allow accessibility to as wide an audience as possible.
- The second issue is providing replacement functionality to those browsers that do not support a particular file type, etc. This is generally done using a "plugin": a helper application that runs alongside a browser to extend its functionality. If a plugin might be required to view a Web page, there should be a link to a source of the plugin.

Interoperability requires adherence to standards both by content producers and browsers.

2. INTERACTIVE WEB PAGES – THE TECHNOLOGY

2.1 Markup Languages

Markup languages are data formats allowing text data to be mixed with tags that add special meaning. The most famous example of this is "hypertext", which adds "hyperlinks" to text for Web navigation. However, there are many applications of markup languages beyond hypertext. These range from other types of human-readable document (diagrammatic, mathematical) to data files without implicit rules for rendering for human view: IT applications can encode many types of abstract data as markup.

The evolution of the Internet, the process of making the Web experience richer, involves creating new markup languages. To make this easier, generic markup formats are used, of which the most prominent is XML.

2.2 Generic Markup Formats

The process of defining new markup languages would be very slow if every language had to be defined in its entirety without reference to any other format. This would also make development of supporting software very time-consuming. Generic markup formats provide standard structures that other languages can use as a baseline.

SGML

SGML (Standard Generalised Markup Language) is a generic markup language dating from the mid 1970s. It defines a particular format for markup using tags in angle brackets:

```
<tag>Text data</tag>
```

The most famous language based on SGML is HTML. HTML defines a number of tags with specific meanings for hypertext documents, including <p> for paragraphs and <a> ("anchor") for hyperlinks.

SGML is rather a complicated language, however. Its use has declined in favour of a simpler alternative: XML.

2.3 XML

XML (eXtensible Markup Language) is a simpler, restricted successor of SGML. Omission of unnecessary SGML constructs has made XML easier to learn and supporting software easier to write.

Here are some of the many applications of XML:

Language	Purpose
XML Schema	Defining languages based on XML
XHTML	Hypertext document structure
XUL	User interface descriptions
SVG	Vector graphics
MathML	Mathematical notation
XSLT	XML-to-XML Transformations
XSL-FO	Document presentation

All of these use tags of the same general form, as defined by XML. However, the actual tags they use, and their meanings, are different.

2.4 Hypertext – HTML and XHTML

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.

2.4.1 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 B Section 2.4.

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. HTML is not a programming language such as C++ or Fortran, as it has no commands or functions, etc. HTML defines the structure and content of a document, and can be interpreted by web browsers to produce a final page.

A web page is defined by a file consisting of a number of HTML tags describing the structure of the page, along with any resources (images etc.) required 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. However, many tools of varying sophistication (and cost) do exist for "authoring" HTML.

```
<html>
  <head>
    <title>
      Example HTML for a simple web page
    </title>
  </head>
  <body>
    <h1>
      Hello, this is a simple web page.
    </h1>
    <p>
      It has very little to it.
    </p>
    <hr/>
  </body>
</html>
```

Figure 2.1 A simple HTML page

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 an 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 with angle brackets, i.e. the "less than" (<) and "greater than" (>) symbols. Most tags are paired up, as on and off switches for a formatting instruction.

For instance, in Figure 2.1 there are <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. There is a shorthand for a pair of tags with no content: in the example, <hr/> appears instead of <hr></hr> (used to insert a horizontal rule).

For a full listing of HTML tags and their usage, consult one of the resources notes in Appendix B Section 2.4. 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;
- The <head> ... </head> pair of tags contains the header information for the file;
- The <title> ... </title> tags enclose 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, which should be used once and only once in the page, enclose the main heading displayed on the page.

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 B Section 2.4.

2.4.2 XHTML

HTML is not quite an XML format: a new variant of HTML, XHTML, was developed to be compliant to XML whilst staying close enough to HTML to be understood by browsers. The examples of HTML in this guide are also valid XHTML, except that, for brevity, we omit from the start of the listings some declarations that are required for validity. For reference, Figure 2.2 shows a simple XHTML document with these declarations included.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-GB">
<head>
  <title>XHTML Document</title>
</head>
<body>
  <h1>XHTML Document</h1>
  <p>This document is valid XHTML 1.1.</p>
</body>
</html>
```

Figure 2.2 A valid XHTML document

The first line is an "XML declaration": it provides an XML version number and indicates the "character encoding" of the document. The DOCTYPE declaration provides a reference to the version of XHTML (1.1 in this case) and a URL that can be used by programs to validate the document against the XHTML 1.1 "Document Type Definition" (DTD) document published by W3C. Finally, the <html> tag specifies the XML "namespace" that qualifies the names of XHTML elements and declares the language of the document to be British English. A full description of these directives is beyond the scope of this guide.

2.4.3 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 another object, such as an image, that is tagged in the HTML code as an index to some other piece of information, called the "target", somewhere on the Web page or site (short links) or to another Web site (long links). 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.

To enable the browser to locate the target resource, the hyperlink contains a Web address or "URL" (Uniform Resource Locator). The general format of a URL is as follows:

```
scheme://host.domain.domain/directory/directory/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 "host" 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).

A URL can be abbreviated: if the scheme and host are missing, the target is reached using the scheme and host that were used for the referring page. Additionally, all or part of the directory path of the target can be omitted. In this case, paths starting with a slash ("/") are referenced relative to the host's top-level directory, whereas paths without the leading slash are referenced relative to the referring document's "base address". The base address of a Web page is usually the URL from which it was retrieved, but this can be overridden using the <base> HTML element in the <head> of the document:

```
<base href="Base address of document"/>
```

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.png">
  <img src="images/image1thumbnail.png" alt="Link to image1"/>
</a>
```

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

```
<a href="http://www.name.domain/images/image1.png">
  <img src="http://www.name.domain/images/image1thumbnail.png"
    alt="Link to image1"/>
</a>
```

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 contents should not be embedded on a Web page if it is to be downloaded separately.

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 PNG files, but otherwise it will prompt the user to decide if they want to open the file using some other (possibly platform-specific) application, or save it to disk.

2.5 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:

```
<head>
  <link rel="stylesheet href="path/MyStyleSheet.css"
    type="text/css"/>
</head>
```

- 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>
  <style type="text/css">
    <--
    h1
    {
      font-size: 15pt;
      font-weight: bold;
      color: maroon;
    }
    -->
  </style>
</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.

```
<body>
  <p style="font-size: 150%; color: red; background-color: yellow">
    Paragraph of text.
  </p>
</body>
```

- 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.

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 overrides the browser default styles, an embedded style sheet overrides a linked style sheet and an inline style overrides an embedded style sheet.

An example of the use of a linked CSS is shown in Figure 2.3 on page 9. The CSS file consists of a set of declarations. The first one sets the defaults for the headings h1 to h3 as centred bold Arial font. The next two declarations modify the styles of h2 and h3 headings. The alignment of h3 headings is now specified twice; in this case the second definition prevails, but only because it comes second. Next, two "classes" are defined, "highlight" and "copyright", which may be applied as a property of any type of element. The final declaration specifies that highlighted parts of copyright sections should be silver, not yellow. This rule overrides the rule for "ordinary" sections of class "highlight" because it has a more specific selector.

The HTML in Figure 2.4 on page 10 demonstrates how to use this stylesheet. The HTML file links to example.css using a "link" element in the header section of the document. The two screenshots (Figure 2.5 and Figure 2.6) 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.

CSS is capable of far more advanced styling. It allows 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 (Document Object Model, see Section 2.17), CSS properties are accessible to scripting languages, and it is therefore possible to dynamically change almost anything about the way a page looks.

```
h1, h2, h3
{
  text-align: center;
  font-family: Arial, sans-serif;
  font-weight: bold;
}

h2
{
  font-style: italic;
}

h3
{
  text-align: left;
}

.highlight
{
  font-weight: bold;
  background: yellow;
}

.copyright
{
  text-align: left;
  font-size: 70%;
  font-style: italic;
}

.copyright .highlight
{
  background: silver;
}
```

Figure 2.3 CSS document "example.css"

```
<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 <span class="highlight">copyright</span> notice.
    </p>
  </body>
</html>
```

Figure 2.4 Example Web page using a linked stylesheet

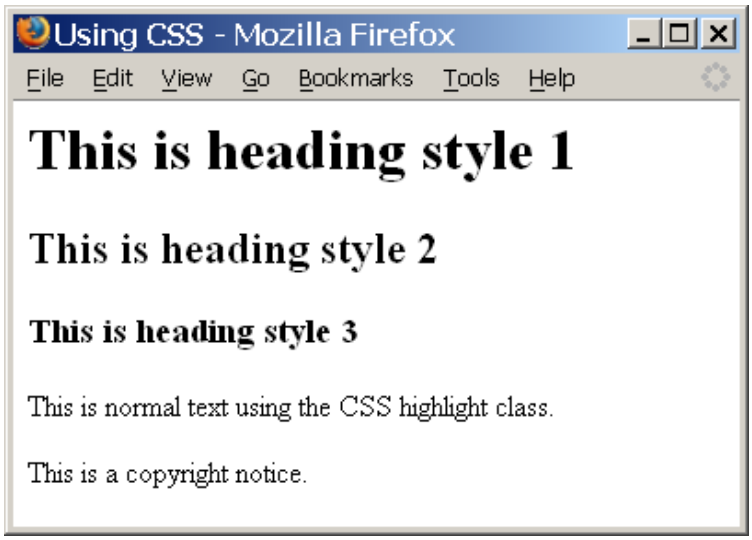

Figure 2.5 The appearance without the CSS

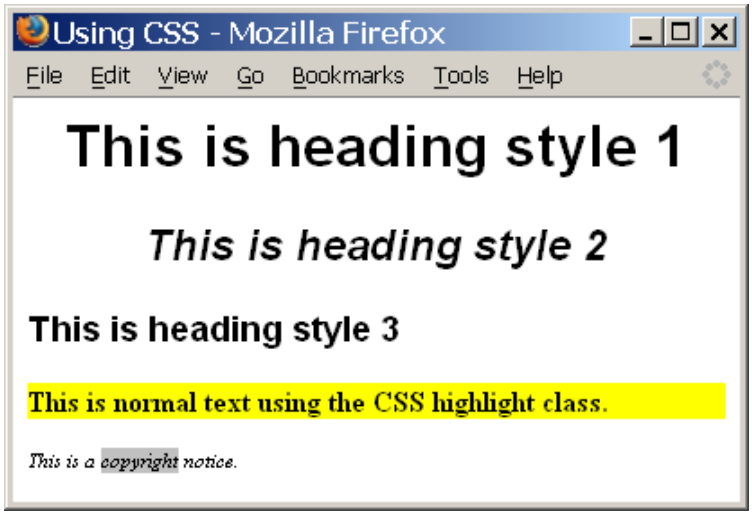

Figure 2.6 The appearance with the CSS

2.6 Multimedia

2.6.1 Images

Images on the Internet are almost entirely of one of three types: GIFs (file name: *.gif), PNGs (*.png) or JPEGs (*.jpg). Each is produced by applying a data compression algorithm to an image file, such as a bitmap, with the goal of making it 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 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 B Section 2.6.

- GIF (Graphics Interchange Format - pronounced either "gif" as in "gift" or "jif"): GIF is a format suitable for images with only a few distinct colours, such as line drawings, lettering 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 (progressive display); and the newer GIF89a, which additionally offers background transparency and animation. An animated GIF comprises a series of GIF sub-images which are played in succession like a movie. All browsers now support GIF89a.

  Use of GIFs has, however, been overshadowed by concerns over patents that cover its compression algorithm. These concerns led to the development of a superior replacement, PNG.

- PNG (Portable Network Graphics - pronounced "ping"). PNG format was devised 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 colour (not including the alpha channel) or 16-bit greyscale; so saving, restoring and re-saving an image will not degrade its quality, unlike standard JPEG (even at its highest quality setting). However, its compression ratio is not as good as JPEG at its best.

PNG has one drawback compared to GIF: it does not support animation. An animation format derived from PNG, called MNG (Multiple-image Network Graphics - pronounced "ming"), exists but is not widely supported. The main problem with MNG appears to be its complexity. Two new alternatives, called "APNG" and "MNG-VLC", have been proposed, both of which add animation to PNG without the complexity of MNG. However, it is not yet clear that either will be widely supported.

PNG is an open format: it is not encumbered by patents. PNG is supported by most browsers although Microsoft Internet Explorer does not, at time of writing (October 2004), support alpha channels.

- JPEG (Joint Photographic Experts Group - pronounced "jay-peg"): 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 but not so well on lettering, simple cartoons or line drawings. JPEG is the predominant storage format used by digital cameras.

  JPEG compression is lossy, meaning that some information is lost when storing images in this format. The payoff for this is smaller file size and hence faster downloads. To enable lossy compression without significant loss of quality, 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.

  As a result of its compression method, JPEG offers superior compression ratios to GIF and PNG for scanned photographs and similar material.

2.6.2 Vector Graphics

"Vector graphics" images are built from geometric elements rather than pixels. One advantage of this is that the image can be rendered on devices offering widely varying graphical resolutions.

SVG (Scalable Vector Graphics) is an XML-based two-dimensional vector graphics format. Specifications of SVG are published by the World Wide Web Consortium (W3C). SVG provides native support for animations and can also be animated by JavaScript using the DOM (see Section 2.17). DOM and JavaScript can also be used to make SVG documents interactive.

Currently, the major browsers do not support SVG natively but instead require plugins to render it, which means that it is not accessible to JavaScript running in the browser. There are plans for Firefox to support SVG. SVG is also supported by the mobile phone industry, which has assisted W3C in the creation of simpler formats SVG Tiny and SVG Basic.

2.6.3 Animation

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

- GIF 89a: As described above, the GIF 89a data format supports animation. This produces movement by playing a number of sub-images, and can play through an image sequence either once or repeating indefinitely. To animate a large image smoothly would be impractical in GIF due to the large size of file required.
- 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.

  Java provides precise control over the animation both at design and run time. Parameters such as play speed, pause duration 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 libraries can read JPEG, GIF and PNG files, which are by far the most extensively used. It provides a number of tools for manipulating images, such as scaling and grabbing regions of an image. The Java 2D API provides more sophisticated tools including alpha blending (image composition with partial transparency), bezier paths, coordinate transformations, textures and gradients.
- Client side scripting (see Section 2.7): 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. The Document Object Model (DOM) provides access for the scripts to manipulate every aspect of the web page except those implemented by plugins (see Section 2.17 below).
- VRML - Virtual Reality Mark-up Language (see Section 2.14): 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 imposed by the world viewer software.
- ActiveX Components (see Section 2.19): 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 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 plugin is required to view Flash animations.

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

2.6.4 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 multimedia presentation where the different media support one another. It is also required to provide direct access to resources that are intrinsically audio such as music and speech.

A naive encoding for sound data would lead to impractically large files. As an example, one second of CD-quality stereo sound would require 44,100 samples (the sample rate of CDs is 44.1 kHz) on 2 channels (stereo) with 16 bits per sample, taking 1,411,200 bits or approximately 172 KB. Consequently, 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.

As with images, sound can be encoded in lossless or lossy ways. Lossy compression will generally provide better compression ratios by taking advantage of transformations that make little or no subjective difference to the listener. There are many coding schemes in use, but the following are the most widely used:

- MP3 (*.mp3) (ISO-MPEG-1/2 Audio Layer-3): MP3 is a popular lossy encoding for music and portable MP3 players are commonplace. The MP3 format typically provides a compression ratio of 1:10 for stereo CD-quality sound. A drawback of MP3 is that the MP3 algorithms are patented and the patent owners demand licensing fees for commercial uses of MP3 in Internet streaming applications.
- WAV (*.wav) (WAVeform audio format): WAV is the original standard audio format for Windows PCs. It can contain audio information in various different encodings but most usually contains data in a simple, inefficient lossless format. It is not suitable for Web use due to the large size of WAV files.
- Ogg Vorbis (*.ogg): Ogg Vorbis is a patent-free alternative to MP3. It is not as widely supported as MP3 but it is becoming more popular: there are many software encoders and players, and several portable music players support the format.
- Windows Media Audio (*.wma) (WMA): WMA is Microsoft's lossy audio format. It is patented and licences are required to create services using WMA files. WMA provides support for digital rights management (DRM), allowing content providers to restrict the usage of copyrighted works.

2.6.5 Movies and Video

Using appropriate plugins, various movie formats may be viewed through a Web browser. All the movie formats use compression schemes to minimise download times, but even so a couple of minutes of high quality video is likely to be over a megabyte in size.

To reduce waiting times, some formats support streaming, where the video is buffered at the client end and playback from the buffer begins before the entire video has been transferred.

Despite these measures, a fast Internet connection is required for video playback of an acceptable resolution and quality, and this should be taken into account when offering movies to a target audience. In addition to the bandwidth issue there is the issue of storage: as video files are large, both the server and the user are going to have to use a significant amount of hard disk space to store them. Recently hard disk sizes have increased rapidly, making this less of an issue.

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 adds many technologies and is particularly suitable for low bit-rate audio-visual coding scheme using audio-visual objects.

  MPEG 1 is the standard generally used to show movies across the 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 plugin 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.

  QuickTime 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 multimedia formats, such as AVI and MPEG, MP3, WAV and MIDI.

  In addition, Apple has developed a QuickTime API for Java, 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 0.03 MB/sec. However if you use maximum settings, it can run as high as 0.3 MB/sec.

- WMV (Windows Media Video, *.wmv): Microsoft's container format for audio and video. WMV includes support for Digital Rights Management (DRM). WMV data can appear in *.wmv files or can be included in an AVI file.

For web links to movie and video information see Appendix B Section 2.6.

2.6.6 Good Practice

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

As with all Web pages, any multimedia 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, PNG and GIF are widely supported, so only the choice of format based on image type, quality etc. is at issue; bear in mind, however, that not all browsers have support for alpha channels in PNG images;
- 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 1 or 2 KB 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 plugins 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.

With video, to get the highest quality results, the video should be captured at a high resolution (640x480 or higher). Even if it is intended that smaller final movies are to be delivered, a high-resolution 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.

2.7 Client-Side Scripting

Client-side scripts are programs written in a scripting language which are embedded into the web page HTML and are run on the user's machine by the browser. Some browsers are unable to run particular scripting languages. In this case a plugin or a change of browser will be required.

2.7.1 JavaScript

JavaScript is the most widely supported client-side scripting language for the Web. JavaScript is an object-oriented scripting language, most useful in embedded contexts from which it can access objects of its host environment.

Microsoft has defined a similar language to JavaScript, called JScript. The IT standards body Ecma International has published a standard for a third language, called ECMAScript, which contains essentially the features that JavaScript and JScript have in common.

```
<body>
  <p>This is HTML.</p>
  <script type="text/javascript">
    <!--
    var message = "This is JavaScript!";
    document.write("<h2>" + message + "</h2>");
    // -->
  </script>
  <p>This is HTML again.</p>
</body>
```

Figure 2.7 JavaScript embedded in HTML

Figure 2.7 shows how JavaScript can be written inside an XHTML <script> element.

The JavaScript is surrounded by XML comment delimiters <!-- and --> to maintain well-formedness of the XHTML: without them, the "<h2>" and "</h2>" string literals would be parsed as tags. Unfortunately, XML comments may not contain two adjacent hyphens and "--" is an operator in JavaScript.

```
<body>
  <p>This is HTML.</p>
  <script src="test.js" type="text/javascript" />
  <p>This is HTML again.</p>
</body>
```

Figure 2.8 JavaScript referenced from HTML

Figure 2.8 shows an alternative to embedded JavaScript: here, the JavaScript is stored in a separate file and referenced by URL. This method has several advantages:

- The JavaScript does not need commenting for XML well-formedness.
- There are no character sequences that will be forbidden e.g. "--".
- Browsers can cache the script so download times will be lower whenever a script is re-used.

Some key features of JavaScript are:

- JavaScript is object-oriented, like Java or C++, and so deals with objects, methods and properties. The JavaScript above invokes the "write" method of the "document" object and passes it the variable "message". This will change the properties of the document object such that the contents of the variable "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 an <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".

- Although almost all browsers support JavaScript, some do not, 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 --> (XHTML 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.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN"
  "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en-GB">
  <head>
    <title>Simple JavaScript</title>
    <script type="text/javascript">
      <!--
function duplicate(element)
{
  element.parentNode.appendChild(element.cloneNode(true));
}
      // -->
    </script>
  </head>
  <body>
    <p onclick="duplicate(this)">Click me!</p>
    <p onclick="duplicate(this)">Click me instead!</p>
  </body>
</html>
```

Figure 2.9 Handling events with JavaScript

The JavaScript in Figure 2.9 illustrates this ability to respond to events. When the text in one of the paragraphs is clicked, a copy of that paragraph is appended to the document.

The "duplicate" function is defined in the <head> of the document. It can now be called from within another <script> or an "active" element, i.e. one that can generate an event. In this case the elements that call the function are the two <p> paragraph elements. When one of the paragraphs is clicked, the onclick event is triggered, and the "duplicate" function is called. The "parentNode", "appendChild" and "cloneNode" identifiers are defined by the Document Object Model (DOM) (see Section 2.17).

When using JavaScript it is very important to test the page on a number of different browsers as they each behave slightly differently. This is particularly true of DOM functionality as there are significant differences between, in particular, Microsoft Internet Explorer and Firefox. 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. However, such scripts require extensive maintenance to support new browsers and new versions of existing browsers. Instead, follow the JavaScript, XML and DOM specifications and restrict yourself to the functionality that works well in all popular browsers.

2.7.2 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. VBScript is very similar in functionality to JavaScript and is a subset of the widely used Microsoft Visual Basic programming language.

VBScript is not commonly used due to its proprietary nature and platform dependence. The far more common JavaScript has been integrated into all of the popular graphical browsers and can be considered to be platform-independent.

For links to VBScript information see Appendix B Section 2.7.

2.7.3 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.

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. For more information, see the "Software development, testing and validation" SSfM theme page at http://www.npl.co.uk/ssfm/ssfm3/theme3/.

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 2.8. Note how it is no longer necessary to use HTML comments to hide the JavaScript from browsers not supporting JavaScript/VBScript.

2.8 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 Section 1.4 Servers and Clients.

2.8.1 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.

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:

- Google (http://www.google.com)
- Ask Jeeves (http://www.ask.com)
- MSN Search (http://search.msn.com)

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. Examples of subject guides are:

- Open Directory Project (http://dmoz.org)
- Yahoo (http://www.yahoo.co.uk)

For links to resources on CGI see Appendix B Section 2.8.

2.8.2 PHP

PHP is a scripting language that is widely used in Web servers. Instead of being separate CGI scripts, PHP code is usually embedded in HTML pages. A Web Server plugin executes the code and replaces the PHP code with the output generated by the PHP. So, for example, the snippet

```
<p>Hello, <?php echo 'World'; ?>!</p>
```

in file hello.php will arrive at the client as

```
<p>Hello, World!</p>
```

PHP includes bindings to many libraries including database clients, XML-handlers, PDF-generators and image-manipulators.

2.8.3 Server-Side JavaScript

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".

2.8.4 Good Practice

See the notes on client-side good practice on page 19.

2.9 Java

Java is a language widely 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, if not 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 is 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. There is a speed penalty for the VM model, but this has decreased with the release of more sophisticated VMs, 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 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, developers can create new packages of their own.

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

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.mscg.DataTable is found in "/npl/mscg/" and is a class developed by the National Physical Laboratory (NPL), Mathematics and Scientific Computing Group (MSCG).

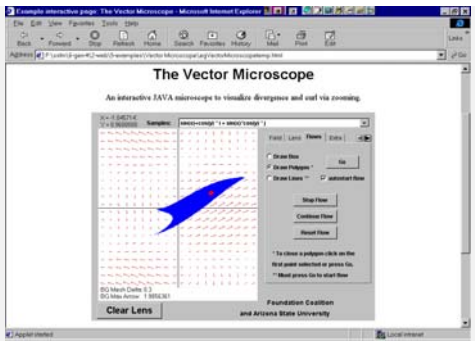

Figure 2.11 Screenshot of an example applet

The main use for Java is in dowloadable, 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 subclasses 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 in Figure 2.12 below, 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>
```

Figure 2.12 Embedding an applet in an HTML document

<applet> elements should have at least the attributes shown. The code attribute defines the path and applet class file from the document base URL. The alt attribute defines the alternative text to be shown while the applet is being retrieved. The name attribute 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.

The <param> element is an applet parameter, where the applet can retrieve the value attribute by dereferencing the name given in the name attribute. 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 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, affecting 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 B Section 2.9.

2.10 XUL

XUL is an XML-based markup language for user interfaces. "XUL" stands for "XML User-interface Language" and is pronounced "zool".

XUL is a by-product of the development of the Mozilla browser. It has two application areas: interactive web pages and standalone applications. The former require Mozilla-based browsers. The latter require "Gecko", which is the component of Mozilla that runs XUL. The user interface of Mozilla is in fact written in XUL.

XUL web pages can by styled with CSS and can contain HTML markup. They can also be assigned JavaScript event-handlers in a similar way to HTML. The event handlers can manipulate the XUL page using the DOM. For example, the XUL script in Figure 2.13 appends text to an initially empty paragraph when the "Execute" button is clicked:

```
<?xml version="1.0"?>
<?xml-stylesheet href="chrome://global/skin/" type="text/css"?>
<window
  id="simple-page"
  title="Simple XUL Example Page"
xmlns="http://www.mozilla.org/keymaster/gatekeeper/there.is.only.xul"
  xmlns:html="http://www.w3.org/1999/xhtml">
  <script type="application/x-javascript">
    function execute()
    {
      document.getElementById('results').appendChild(
        document.createTextNode('Hello!'));
    }
  </script>
  <html:p>This is a test XUL page</html:p>
  <button id="execute-button" label="Execute"
    oncommand="execute(); return true;"/>
  <html:p id="results"/>
</window>
```

Figure 2.13 Example XUL script

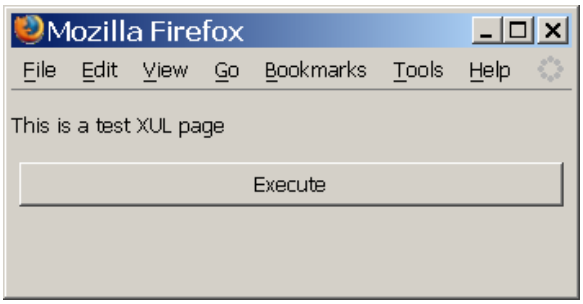

Figure 2.14 The XUL page as it first appears

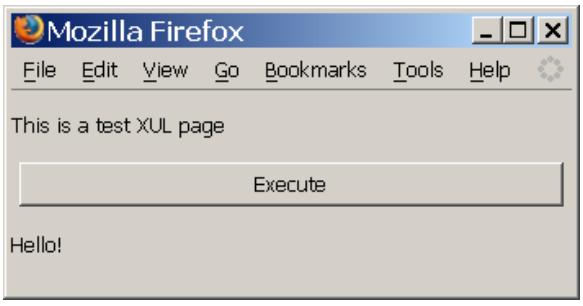

Figure 2.15 The XUL page after the button is clicked

2.11 XBL

XBL ("eXtensible Bindings Language") is an XML-based language for fine-tuning the user interface elements used by XUL. With XBL you can define properties and methods for elements that can subsequently be accessed from the XUL using JavaScript and the DOM. Builds of the Firefox browser with SVG enabled allow graphical user interface elements to be defined using XBL with SVG.

2.12 XAML

XAML ("eXtensible Application Markup Language") is, like XUL, an XML-based language for user interfaces. It was developed by Microsoft and is expected to be supported by "Longhorn", the successor to Windows XP operating system. Whereas XUL uses JavaScript, XAML will work with any of the programming languages supported by Microsoft's ".NET" development platform.

2.13 Mathematical Symbols on Web Pages

The W3C has recommended a markup language for use of mathematics 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, typesetting and voice synthesis.

There are two dialects to MathML: "presentation markup" and "content markup". Presentation markup provides control over the visual display of the mathematics in terms of notation and layout instructions. By contrast, content markup provides a way to specify the meaning of mathematical expressions in terms of mathematical functions and operators. Thus MathML can be used to encode both the presentation for high-quality visual rendering and the content for applications where the semantics plays more of a key role such as scientific software or voice synthesis.

Software support for MathML is expanding. The Mozilla and Firefox browsers can natively display presentational MathML. For Internet Explorer, a plugin is required. MathML can be written by hand or generated from a number of user-friendly equation editor packages. For example, the OpenOffice.org office suite contains an equation editor that can export expressions to MathML format.

```
<?xml version="1.0"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.1 plus MathML 2.0//EN"
    "http://www.w3.org/Math/DTD/mathml2/xhtml-math11-f.dtd"
>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>MathML Example</title>
  </head>
  <body>
    <p>The formula is
      <math xmlns="http://www.w3.org/1998/Math/MathML">
        <msqrt>
          <mn>1</mn>
          <mo>-</mo>
          <msup>
            <mi>x</mi>
            <mn>2</mn>
          </msup>
        </msqrt>
      </math>
    </p>
  </body>
</html>
```

Figure 2.16 An example of MathML

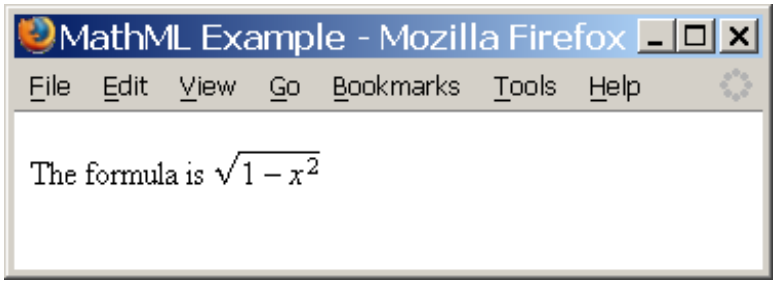

Figure 2.17 MathML displayed by Firefox

MathML can be styled with CSS. For example, the font and colour of parts of formulae can be adjusted. Because MathML is an XML-based format, it can be manipulated by scripts using the DOM (see Section 2.17). For example, JavaScript could be written to allow subexpression expansion to be toggled by mouse clicks, e.g. between "(*x*+1)(*y*+1)" and "*xy*+*x*+*y*+1".

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 PNG image of the desired symbols/equation to be made and displayed, though maintenance and updating of such an image would be relatively difficult and the image may not render well on some viewing systems (e.g. those with very high or low resolutions).

See Appendix B Section 2.13 for links on MathML and displaying mathematical symbols.

2.14 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 2.18 A simple VRML document defining a red cube

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 in Figure 2.18. 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" parentheses are regarded as a single object by VRML. This is called a Group "Node". The group node consists 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.

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".

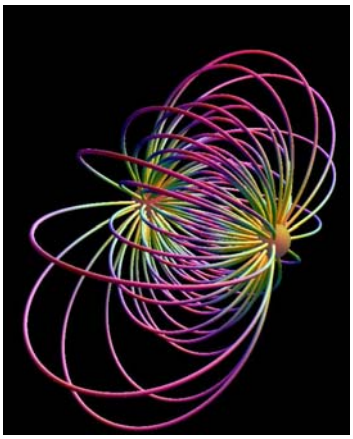

Figure 2.19 A screenshot of a VRML world of the magnetic field around a magnetic dipole

Far more complex objects can then be build up out of a set of simpler objects. Figure 2.19 shows a screenshot of a VRML world modelling the magnetic field around a magnetic dipole.

To view a "world" a VRML viewer (a browser plugin 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 B Section 2.14.

2.15 X3D

X3D ("eXtensible 3D") is a 3D file format derived from VRML. It is being developed by the Web3D Consortium and is in the process of standardisation with ISO.

X3D adds support for multiple file encodings to VRML: there are now three syntaxes, although there is still only one underlying abstract model. One file encoding, "Classic VRML encoding", is very similar to VRML. The others are an XML encoding and a compressed binary encoding.

Unlike VRML, X3D has a modular ("componentised") specification. This allows X3D to be extended in many directions without requiring all X3D tools to support them all.

See Appendix B Section 2.15 for links to Web resources on X3D.

2.16 U3D

U3D ("Universal 3D) is a 3D file format being developed by the 3D Industry Forum ("3DIF"), whose members include Intel and Adobe. 3DIF is working with IT standards body Ecma International and expects to standardise U3D with ISO in 2005.

Promotional material on U3D emphasises its application to CAD (computer-aided design) documents. Specifically, it is promised to be efficient enough to be distributed over the

internet and viewed on affordable computers, and it will be viewable using nonproprietary software.

2.17 The Document Object Model (DOM)

DOM describes a standard API for browsers (and other software libraries) to provide for programmers to use to manipulate XML and HTML documents. Scripts running in DOM-compliant browsers can manipulate Web pages dynamically. Since DOM is applicable to any XML technology, it can, for example, construct MathML and SVG on the fly, in response to user actions.

The DOM specification documents are published by the W3C. These specifications provide three different "levels" of DOM. Level 1 was the first, standardised in 1998, and is the most widely supported DOM standard among browsers. Levels 2 and 3 have followed, both adding functionality to their predecessors.

The W3C DOM specifications specify ECMAScript and Java implementations, but libraries providing DOM interfaces exist for several other languages too, including PHP, Perl and C. In the context of Web browsers, it is the ECMAScript API that is most relevant.

Each level of DOM is divided into sections: for example, DOM Level 2 has a "core", for general XML manipulation, and "HTML", "Style" and "Events" parts among others.

The "Style" part allows DOM applications to dynamically alter the stylesheet used by the browser when rendering the Web page.

The part of the DOM that specifies which elements can trigger changes is called the "Event Model". Events are generated by actions such as 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 B Section 2.17

2.18 XSLT

XSLT (Extensible Stylesheet Language Transformations) scripts specify rules to transform one XML document into another. These scripts are sometimes called stylesheet because they provide a way of applying presentational styles to unstyled XML. However, they are more general in scope than CSS, due to the range of available XML-based languages. For example, an XSLT stylesheet could transform an XML dataset into an SVG image of a graph representing that dataset.

Here is an example of an unstyled XML document of an ad-hoc schema referring to an XSLT script that renders it as XHTML:

```
<?xml version="1.0"?>
<?xml-stylesheet href="simple.xslt" type="text/xsl"?>
<document title="XSLT test document">
  <warning>XML should always be well-formed!</warning>
  <warning>XHTML should always be well-formed and valid!</warning>
</document>
```

The "simple.xslt" file, referred to above, has the following contents:

```
<?xml version="1.0"?>
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns="http://www.w3.org/1999/xhtml">
  <xsl:output method="xml"/>
  <xsl:template match="/document">
    <html>
      <head>
        <title>
          <xsl:value-of select="@title"/>
        </title>
      </head>
      <body>
        <h1>
          <xsl:value-of select="@title"/>
        </h1>
        <xsl:apply-templates/>
      </body>
    </html>
  </xsl:template>
  <xsl:template match="warning">
    <p>
      <strong>
        WARNING:
        <xsl:value-of select="text()"/>
      </strong>
    </p>
  </xsl:template>
</xsl:stylesheet>
```

Firefox 1.0 and Internet Explorer 6.0 correctly render the XML file by applying the XSLT stylesheet and interpreting the result as XHTML. To see the generated XHTML, a standalone XSLT processor such as Saxon can be used. The output (after some whitespace modifications for readability) is:

```
<?xml version="1.0" encoding="UTF-8"?>
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <title>XSLT test document</title>
  </head>
  <body>
    <h1>XSLT test document</h1>
    <p>
      <strong>
        WARNING:
        XML should always be well-formed!
      </strong>
    </p>
    <p>
      <strong>
        WARNING:
        XHTML should always be well-formed and valid!
      </strong>
    </p>
  </body>
</html>
```

XSLT supports invocation of named templates that take parameters, and evaluation of XPath expressions, which include arithmetic operators. XSLT is therefore powerful enough to be called a programming language, although its syntax makes it less attractive than JavaScript for complex algorithms.

2.19 ActiveX

ActiveX is a Microsoft Windows technology to allow software components called ActiveX "controls" to be embedded in Web pages. ActiveX controls have access to the Windows API (Application Programming Interface) and can therefore use operating system services to draw graphics, use the file system and more, as if they were installed software packages.

This unrestricted access to the operating system means that ActiveX controls present a potential security risk. A malicious control could easily delete files or spread viruses. It is up to the user to install ActiveX controls from trusted sources only. The Java Runtime Environment, however, restricts the operating system access of Java applets so that they pose less of a risk.

Another serious drawback of ActiveX compared to Java is that ActiveX controls are tied to a particular architecture, whereas Java applets are platform-independent.

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 National Measurement System (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 , such as the "cost" of 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.npl.co.uk/ssfm/webservices/ssfm_interactive_examples/

For links to examples of data transformation applications see Appendix B 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.

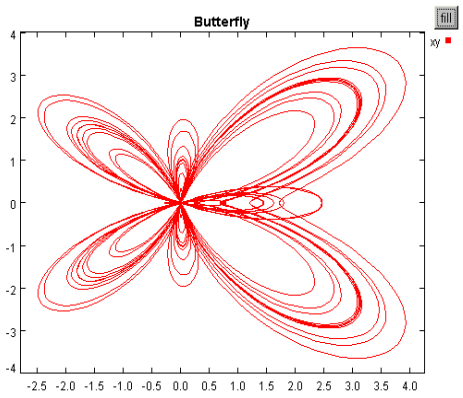

Figure 3.1 A butterfly graph produced online using a Java applet

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.

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.

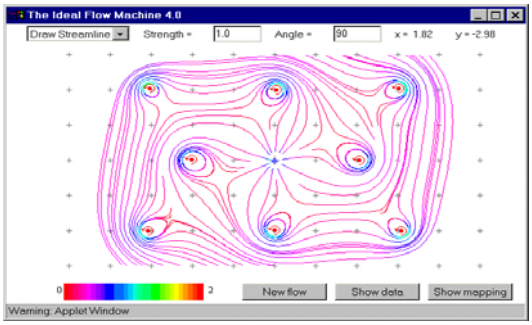

Figure 3.2 An interactive Ideal Flow model implemented as a Java applet

- 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 SS*f*M web site:
  http://www.npl.co.uk/ssfm/webservices/ssfm_interactive_examples/

- 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 "practise" 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.

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 laboratory", 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 B Section 3.4.

3.5 Database Queries

Web pages are increasingly used to access databases. The database is typically run on the Web server or on a machine on the same Local Area Network as the Web server. Server-side scripts access the database, construct Web pages dynamically and handle the users' responses. This avoids the need for installation of dedicated software on the client machines, and provides potentially worldwide access.

For web links to examples see Appendix B Section 3.5.

3.6 Miscellaneous Web Page Enhancements

For links to miscellaneous interactive Web page enhancements see Appendix B Section 3.6.

3.6.1 Site Navigation

Navigation bars can be given dropdown lists with Java, JavaScript or CSS. Dropdown menus can be achieved without scripts or Java applets: for an example of a dropdown menu achieved with CSS alone see http://www.meyerweb.com/eric/css/edge/menus/demo.html. Note that this example requires a browser supporting CSS version 2, such as Firefox: it will not work with Internet Explorer 6.

3.6.2 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.

3.6.3 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.

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;
- WYSIWYG (What You See Is What You Get) editors. These do not require much knowledge of HTML – they use a visual development interface;

For links to HTML editors, see Appendix B Section 4.1.

4.1.1 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.

4.1.2 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 plugins.

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-get 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.

4.1.3 Hybrid Editors

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 an 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.

4.1.4 Document Conversion

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, PNG 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. A free alternative to Photoshop is The GIMP (GNU Image Manipulation Program).

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 3rd 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 plugin 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 *.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

Browsers require a plugin to be able to play movies embedded in Web pages. On Windows, Apple QuickTime, Real Networks RealPlayer and Microsoft Windows Media Player are all available as free plugins on Windows and Mac OS X. RealPlayer is also available on Linux. "gxine" is another browser plugin available on Linux.

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 also export audio-only files in a number of formats.

See Appendix B 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.

The Mozilla and Firefox browsers provide a JavaScript debugger allowing JavaScript code to be executed line-by-line. The debugger provides interactive access to query and modify variables at runtime.

See Appendix B Section 4.3 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 B Section 4.4 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 Borland JBuilder, NetBeans 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 B 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 plugins 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 B Section 4.6 for links to VRML tools.

4.7 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.

- 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++ may be used to write ActiveX controls, and Microsoft Visual C++ is probably the best environment to do this with, as it supports the ActiveX API directly, and provides such tools as the ActiveX test container for viewing the component..
- Java can also be used to develop ActiveX components. 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 B Section 4.7 for links to ActiveX tools.

5. THE FUTURE

Development of the technologies supporting the World Wide Web and interactivity continues. Presented here are some emerging technologies. Some are already close to implementation.

5.1 Synchronised Multimedia - SMIL

To enable simple authoring of TV-like multimedia presentations such as training courses, W3C has designed the Synchronized Multimedia Integration Language, pronounced "smile".

The SMIL standard defines an XML-based language that allows control over the positioning and timing of events 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 edit, and can be assembled on-the-fly by server 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.

Applications supporting SMIL include Microsoft Internet Explorer. See Appendix B Section 5.1 for links to SMIL information.

5.2 XLink

XLink provides an extension to the linking mechanism of HTML. Whereas <a> elements always denote outbound links, XLink supports inbound and third-party links. Third-party links provide a number of "arcs", each of which links any number of resources. XLink provides a mechanism to support "linkbases": directories of links that can be published independently of the resources from which links are traversed.

5.3 The Semantic Web and RDF

The Semantic Web is a concept promoted by the W3C in which relationships between web resources, and also abstract entities, can be expressed in a way that can encapsulate meaning in extensible vocabularies. RDF (Resource Description Framework) is an XML-based notation used for expressing relationships in the Semantic Web framework.

The abstraction of vocabularies makes RDF essentially a generalisation of XLink, as the latter is restricted to specifying hyperlinks. RDF vocabularies exist for a wide variety of applications. For example, the Dublin Core Metadata Initiative publishes an RDF schema (vocabulary) to empower RDF documents to make statements about document metadata such as "document X references/supersedes/is part of document Y" and "Resource R has version number V".

A significant application of Dublin Core metadata is DSpace (http://www.dspace.org/). Developed by MIT and Hewlett-Packard, DSpace is a web-accessible database aimed at indexing the intellectual output of a research organisation. There are many organisations indexing their output with DSpace, including MIT, Cambridge University, Cornell University, the University of Maryland and Glasgow University.

5.4 E4X

E4X (ECMAScript for XML) is an extension of ECMAScript (and therefore JavaScript) to accommodate XML as a native data type. It promises to provide a simpler, far less verbose way of manipulating XML than the DOM for JavaScript. For example, XML fragments are interpreted directly as literals of type "xml". A drawback is that DOM has bindings in many languages whereas E4X represents an extension to the syntax of JavaScript, so equivalents in other languages will require significant work. E4X has been standardised by Ecma International, the same body that has standardised ECMAScript.

The following snippet of E4X-enhanced JavaScript demonstrates the key features:

```
var x = <p>This is XHTML as a <em>native</em> JavaScript literal</p>;
// x has the new "xml" data type
var y = "The emphasised text is: " + x.em;
// y now contains the string "The emphasised text is: native"
```

E4X support for the Firefox browser is under construction.

5.5 XForms

XForms is a re-engineering of HTML forms. It provides finer control over the user interface. In particular, it aids implementation of forms in which some values are calculated from others, and improves the user experience by enabling more thorough validation before sending a response to the web server.

5.6 XSL-FO

XSL-FO (Extensible Stylesheet Language Formatting Objects) is an XML document containing information together with instructions on how to lay out its contents in pages. Tools exist to convert XSL-FO to PDF. This makes XSL-FO a good choice of target format for XSLT stylesheets. In fact, this was the original purpose of XSL-FO. For example, the DocBook markup language for technical documents defines an XML schema and provides an XSLT stylesheet mapping the DocBook XML source to XSL-FO, from which it can be converted to PDF.

5.7 Compound Document Formats

In October 2004 the W3C established a "Compound Document Format Working Group", with the aim of establishing specifications for document formats combining selected document formats such as XHTML, SMIL, SVG and MathML. The group will address not only web browsers on desktop computers but also devices with restricted capabilities, such as mobile phones and interactive televisions.

This will accelerate the adoption of compound formats by authoring tool providers and browser developers.

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 set of functions provided by a software library. Operating systems provide APIs that programs can use to interact with the user and the file system.
APNG – Animated Portable Network Graphics	A proposed extension to the PNG image format that adds support for animation.
AVI – Audio Video Interleave (file *.avi)	A multimedia file format defined by Microsoft.
Applet, Java	A program written in 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 “plugin” component to construct a larger application.
Bean Box	An application allowing a user to plug Java Beans together to produce an application or applet.
Bit – Binary digIT	A switch with two states called "0" and "1"; a unit used to specify the size of a range of values such as the number of different colours a monitor can display. A range is doubled by adding one extra bit. Thus there are two 1-bit numbers (0 and 1), four 2-bit numbers (00, 01, 10 and 11), eight 3-bit numbers etc.
Browser	An application that can navigate the Internet and interpret HTML files to display web pages.
Byte	The basic unit of file size, comprising eight bits.
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.
CAD – Computer-Aided Design	The use of computers to assist the development of 3D structures such as architectural or mechanical designs.
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 it 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 (file *.css)	A language for customising the presentation of Web pages.
DHTML – Dynamic HTML	The name for a collection of technologies that may be used to make a Web page respond to user actions and change its appearance after downloading. The key technologies are HTML, JavaScript, CSS and DOM.
DLL – Dynamic Link Library (file *.dll)	A file format used on Windows systems to store software libraries that can be used by one or more applications.
DOM – Document Object Model	The technology that describes a Web document in terms of object-oriented data structures that may be accessed via a scripting language allowing the document to be manipulated after downloading.
Ecma International	An information technology and consumer electronics standardisation organisation.
ECMAScript	A scripting language standard published by Ecma International to set a baseline for both JavaScript and JScript.
Fortran – FORmula TRANslation	A programming language dating from the 1950s and widely used for numerical and scientific computing.
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 (file *.gif)	A very popular image file format used on the Internet. May be used to produce animations.
GIMP – GNU Image Manipulation Program	A free software package for image editing and enhancement.
GNU – "GNU's Not Unix"	An organisation formed to create a free operating system and to encourage cooperative software development.
GUI – Graphical User Interface	Software functionality that provides windows, icons, buttons etc. for interaction with the user.
HDTV – High Definition Television	Television technology offering higher-resolution images than traditional formats.
HTML – Hypertext Markup Language (file *.html or *.htm)	The language used to specify the structure and 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 annotated 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" describes interfaces to that world that 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.
ISO – International Organisation for Standardization	An eminent international standards body
ISP – Internet Service Provider	An ISP provides a link into the Internet for a client machine.
Java – Applet: see Applet, Java – Bean: see Bean, 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

	code and runs the instructions locally. Each specific platform must have its own Java VM so that the code is translated correctly for it.
JDK – Java Development Kit	The Java development environment
JPEG – Joint Photographic Experts Group (file *.jpg)	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)
MathML	An XML-based markup language for mathematical notation on the Web.
MIDI – Musical Instrument Digital Interface	A music file format.
MNG – Multiple-image Network Graphics	An animated image format that uses PNG format internally and provides more advanced functionality than GIF.
MNG-LC – Multiple-image Network Graphics, Low Complexity	A subset of MNG, trading some features for reduced complexity and thus potentially wider support.
MNG-VLC – Multiple-image Network Graphics, Very Low Complexity	A subset of MNG, more reduced than MNG-LC.
*.mov – see QuickTime	
MP3 (file *.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.
NPLFit	A software package developed by the National Physical Laboratory for modelling experimental data using polynomial and polynomial spline curves.
OOP – Object-Oriented Programming	Computer programming that ties functionality to data structures, calling their combination either a "class" (referring to a given type of data structure) or an "object" (referring to a particular instance of a class).
OpenOffice.org	A free, cross-platform suite of office applications including a word processor, spreadsheet and database.
Pascal	A programming language dating from about 1970.
PDF – Portable Document Format (file *.pdf)	A document format invented by Adobe Systems. It provides precise control over the layout of text and images and is therefore suitable for desktop publishing.
Perl	A scripting language, very powerful at text handling. The most common language to

	handling. The most common language to write CGI applications for the web.
PHP (file *.php)	A scripting language that can be embedded in HTML and executed by Web servers.
PNG – Portable Network Graphics (file *.png)	A relatively new file format for graphics being promoted and gaining support for internet use.
QuickTime, Apple (file *.mov, *.qt)	A multimedia file format for movies and sound.
Remote	Activity that takes place on a computer other than the users one. See Local.
RIFF – Resource Interchange File Format	A general-purpose multimedia file format defined by Microsoft and IBM. The AVI format is based on RIFF.
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 Markup Language	The generic markup language upon which HTML was defined. SGML is a superset of XML.
Shareware	Software that may be distributed and evaluated freely, but for which 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.
U3D – Universal 3D	A 3D file format geared towards CAD applications.
URL – Uniform 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.

	development interface.
Vorbis (or Ogg Vorbis)	An audio compression format
VRML – Virtual Reality Modelling Language (file *.wrl)	A standard and language used to define 3D Virtual Reality "worlds" for display in VRML viewers.
W3C – the World Wide Web Consortium	An international consortium founded in October 1994 to develop Web standards and guidelines.
WAV (or WAVE) – WAVeform Audio (file *.wav)	An audio file format defined by Microsoft and IBM. It is a restricted form of RIFF.
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.
Web3D Consortium	An organisation created to develop and promote technologies for communicating 3D visualisations on the Web.
WMA – Windows Media Audio (file *.wma)	An audio encoding format defined by Microsoft.
WMV – Windows Media Video (file *.wmv, *.avi)	A video encoding format defined by Microsoft.
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. In practice, however, "WWW" is often used synonymously with "Internet".
WYSIWYG – What You See Is What You Get	A descriptive term for authoring applications whose display of the document when editing is an accurate impression of its appearance in use (e.g. when printed out or viewed as a Web page).
X3D – eXtensible 3D	An XML-based 3D file format derived from VRML.
XAML – eXtensible Application Markup Language	An XML-based language defined by Microsoft to build user interfaces (and other programming structures) in the successor to Windows XP.
XBL – eXtensible Bindings Language	An XML-based language for fine-tuning the user interface elements used by XUL.
XForms	An XML-based language for Web forms that provides more interactivity features than HTML forms.
XHTML – eXtensible Hypertext Markup Language	An XML-compliant successor to HTML.

Language	
XLink – XML Linking Language	An XML-based language for embedding in XML documents to specify links between them.
XML – eXtensible Markup Language (file *.xml)	A generic markup language, simpler and more widely supported than SGML.
XPath – XML Path Language	A W3C standard syntax for specifying locations in XML documents.
XSL-FO – eXtensible Stylesheet Language Formatting Objects	An XML-based language for formatted documents. Like PDF, it provides precise control over the formatting of text and images.
XSLT – eXtensible Stylesheet Language Transformations	An XML-based language to specify rules for transforming one XML document into another.
XUL – XML User interface Language	An XML-based language for user interfaces, a by-product of the development of the Mozilla browsers.

Appendix B – Useful Resources and Bibliography

Links to useful resources on the Web are categorised by the section in the main part of the Guide that 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 Programme

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

1.4 Browsers

Internet Explorer	http://www.microsoft.com/windows/ie/default.mspx	
Firefox	http://www.mozilla.org/products/firefox/	
Opera	http://www.opera.com/	

2. INTERACTIVE WEB PAGES – THE TECHNOLOGY

2.3. XML

XML at W3C	http://www.w3.org/XML/	
Latest W3C Recommendation	http://www.w3.org/TR/REC-xml/	

2.4. Hypertext – HTML and XHTML

HTML at W3C	http://www.w3.org/MarkUp/	
HTML validation service by W3C	http://validator.w3.org/	
HTML 4.01 Specification	http://www.w3.org/TR/html401/	
HTML Reference by the Web Design Group	http://www.htmlhelp.com/reference/html40/	
Reference material on the use of frames	http://www.htmlhelp.com/design/frames/	
W3C accessibility guidelines (latest version)	http://www.w3.org/TR/WAI-WEBCONTENT/	

W3C's recommnedation for a Math Mark-up Language for use on the Web	http://www.w3.org/Math/	

2.5. Cascading Style Sheets (CSS)

CSS at W3C	http://www.w3.org/Style/CSS/	
CSS validation at W3C	http://jigsaw.w3.org/css-validator/	Accepts a URL or an uploaded file
CSS guide by the Web Design Group	http://www.htmlhelp.com/reference/css/	

2.6. Multimedia

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://www.webstyleguide.com/graphics/	
Guidance on the use of video and audio	http://www.webstyleguide.com/multimedia/	
JavaScript and Sound - a Tutorial	http://www.hotwired.com/webmonkey/97/50/index3a.html	
SVG at W3C	http://www.w3.org/Graphics/SVG/	
Inkscape	http://www.inkscape.org/	A free, powerful and easy-to-use SVG editor available for Windows and Linux
Batik	http://xml.apache.org/batik/	An SVG toolkit in Java
Adobe "SVG Zone"	http://www.adobe.com/svg/	SVG resources including the free "Adobe SVG Viewer"

2.7. Client-Side Scripting

ECMAScript Language Specification	http://www.ecma-international.org/publications/standards/Ecma-262.htm	
Mozilla Organisation JavaScript Language Resources page	http://www.mozilla.org/js/language/	Relates ECMAScript and JavaScript versions
JavaScript for the Total Non-Programmer - a Tutorial	http://www.webteacher.com/javascript/	

Microsoft Windows scripting documentation	http://msdn.microsoft.com/scripting/default.asp	Includes documentation of JScript and VBScript

2.8. Server-Side Scripting

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 (Visual Basic)	http://www.users.dircon.co.uk/~tbrown/vb/cgi.htm	
PHP	http://www.php.net/	

2.9. Java

Sun Microsystems Java Technology page	http://java.sun.com/	
Sun Microsystems Java programming tutorial pages	http://java.sun.com/docs/books/tutorial/index.html	
Sun Microsystems developer home page	http://developers.sun.com/	
Sun Microsystems Java Beans home page	http://java.sun.com/products/javabeans/index.jsp	
Extensive set of Java resource links	http://www.apl.jhu.edu/~hall/java/	

2.10. XUL

XUL page of the Mozilla Organisation catalogue	http://www.mozilla.org/catalog/architecture/xul/	Many links to XUL information
XULPlanet	http://www.xulplanet.com/	Tutorials and reference material

2.11. XBL

XBL page of the Mozilla Organisation catalogue	http://www.mozilla.org/catalog/architecture/xbl/	Links to XBL specification and tutorials
XULPlanet introduction to XBL	http://www.xulplanet.com/tutorials/xultu/introxbl.html	

2.12. XAML

Microsoft "Avalon" articles	http://msdn.microsoft.com/Longhorn/understanding/pillars/avalon/default.aspx	"Avalon" is the Longhorn user interface API

2.13. MathML

W3c "Math Home"	http://www.w3.org/Math/	

2.14. VRML

VRML archives of the Web3D Consortium	http://www.web3d.org/x3d/vrml/	
NPL Centre for Materials Metrology example web site.	http://sulu.npl.co.uk/netshare/guest/welcome1.html	

2.15. X3D

Web3D Consortium	http://www.web3d.org/	
ISO (International Organization for Standardization)	http://www.iso.org/	

2.16. U3D

U3D file format specification	http://www.ecma-international.org/publications/standards/Ecma-363.htm	
3DIF home page	http://www.3dif.org/	
U3D White Paper	http://www.3dif.org/downloads/Universal3D%20Whitepaper.pdf	

2.17. The Document Object Model (DOM)

DOM at W3C	http://www.w3.org/DOM/	
"libxml2" home page	http://xmlsoft.org/python.html	XML-manipulation software library providing DOM bindings for several languages

2.18. XSLT

XSL at W3C	http://www.w3.org/Style/XSL/	Includes links to specifications and online tutorials
Introduction to XSLT using Java	http://java.sun.com/xml/jaxp/dist/1.1/docs/tutorial/xslt/index.html	

2.19. ActiveX

Microsoft ActiveX control documentation	http://msdn.microsoft.com/library/default.asp?url=/workshop/components/activex/activex_node_entry.asp	
Security Tradeoffs: Java vs. ActiveX	http://www.cs.princeton.edu/sip/java-vs-activex.html	"An unofficial view from the Princeton Secure Internet Programming Team"

3. INTERACTIVE WEB PAGES - APPLICATIONS

3.1. Data Transformations

SSfM Examples of Interactive Web pages	http://www.npl.co.uk/ssfm/ssfm1/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/ssfm1/tt/index.html	

NPL Materials interactive site	http://sulu.npl.co.uk/netshare/guest/	
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

EUROMETROS	http://www.eurometros.org/	A metrology software database developed as part of the SS*f*M programme.
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	
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

4.1. Hypertext - HTML

HTML code editors:

Arachnophillia	http://www.arachnoid.com/arachnophilia/	freeware
Misc. freeware selection	http://www.webmastertoolscentral.com/directory/utilities/editors/html-editors.html	
Any text editor	n/a	

Hybrid Editors

Microsoft FrontPage	http://www.microsoft.com/frontpage/	
Macromedia Dreamweaver	http://www.macromedia.com/software/dreamweaver/	
Nvu	http://www.nvu.com/	Free
LiquidFX	http://www.liquidfx.info/	
Misc. Selection	http://www.webmastertoolscentral.com/directory/utilities/editors/html-editors.html	

4.2. Multimedia

Macromedia - Flash, Director, Shockwave, Dreamweaver	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.
The GIMP	http://www.gimp.org/	A free and cross-platform image manipulation tool
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.microsoft.com/windows/windowsmedia/default.aspx	mp3, wav, qt, mov,
WinAmp player for Windows	http://www.winamp.com/	
Various graphics tutorials on Animation, Photoshop, and others	http://www.freegraphics.com/10_Tutorials/	
Adobe Premiere	http://www.adobe.com/products/premiere/main.html	Video encoder, with support for import/export to/from many formats.

4.3 Client-Side Scripting

Firefox browser	http://www.mozilla.org/products/firefox/	Includes a JavaScript debugger

4.4. Server Side Scripting

The CGI Resource Index	http://www.cgi-resources.com/	Very comprehensive
Perl.com	http://www.perl.com/	Contains a great deal of information about Perl
PHP home page	http://www.php.net/	Contains excellent documentation

4.5. Java

Sun Microsystems Java home page	http://java.sun.com/	Provides free downloads of the Java Development Kit (JDK)
Sun Microsystems Bean development tools link page	http://java.sun.com/products/javabeans/downloads/index.html	
Microsoft Visual J# home page	http://msdn.microsoft.com/vjsharp/	
NetBeans	http://www.netbeans.org/	
Borland Jbuilder	http://www.borland.com/jbuilder/	

4.6. VRML

VRML archives of the Web3D Consortium	http://www.web3d.org/x3d/vrml/	
Cortona VRML Client	http://www.parallelgraphics.com/products/cortona/	Browser plugin, free for personal, non-commercial use.
FreeWRL	http://freewrl.sourceforge.net/	Free VRML and X3D viewer for Mac OS X and Linux with Windows support planned.
OpenVRML	http://www.openvrml.org/	Free cross-platform VRML library and plugin in development.
Iris Explorer licensed by NAG.	http://www.nag.co.uk/visual/IE/iecbb/Product.html	

4.8. ActiveX

Microsoft ActiveX control documentation	http://msdn.microsoft.com/library/default.asp?url=/workshop/components/activex/activex_node_entry.asp	
Microsoft Visual Studio Developer Centre	http://msdn.microsoft.com/vstudio/	

5. THE FUTURE

5.1. Synchronised Multimedia – SMIL

W3C Synchronised Multimedia page	http://www.w3.org/AudioVideo/	SMIL specifications, players and authoring tools

5.2. XLink

W3C XML linking page	http://www.w3.org/XML/Linking	Documentation of XLink and associated standards

5.3. The Semantic Web and RDF

W3C Semantic Web home page	http://www.w3.org/2001/sw/	Links to RDF tools and references
Dublin Core Metadata Initiative	http://dublincore.org/	Metadata standards group
DSpace	http://www.dspace.org/	Research material archiver that uses Dublin Core.

5.4. E4X

E4X standard at Ecma International	http://www.ecma-international.org/publications/standards/Ecma-357.htm	
Release notes of the Mozilla browser, version 1.8 Beta 1	http://www.mozilla.org/releases/mozilla1.8b1/README.html	In the "Under the Hood" section, partial E4X support is mentioned

5.5. XForms

XForms at W3C	http://www.w3.org/MarkUp/Forms/	
Mozilla XForms page	http://www.mozilla.org/projects/xforms/	

5.6. XSL-FO

XSL family at W3C	http://www.w3.org/Style/XSL/	
DocBook guide	http://docbook.org/	XML documentation format, converted to XSL-FO using XSLT
Apache FOP	http://xml.apache.org/fop/	Software to convert XSL-FO to several other formats including PDF

5.7. Compound Document Formats

Compound Documents home at W3C	http://www.w3.org/2004/CDF/	

APPENDICES

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	
Wikipedia	http://en.wikipedia.org/wiki/Main_Page	A vast resource