5

Accessibility issues for web-based information systems

Paul Catherall

The importance of web standards, usability and accessibility

Introduction

Recent years have witnessed the rapid growth of the World Wide Web as the most popular form of technology used on the Internet – in the areas of education, leisure and commercial activity. While it is generally accepted that the popular web browser application has broken down the barriers of technical complexity seen in early Internet technologies, the popularisation of the World Wide Web has itself given rise to further debate on its usability and accessibility as a medium of information and communications.

While there are many facets to the debate on the growing prevalence of the World Wide Web, including issues of ethics, authority and ownership, this chapter will attempt to define and explore web-based usability and accessibility issues from a technical and practical perspective – with particular emphasis on the accessibility of web browser software as an interface to information for disabled users. It is hoped that the chapter will provide an insight into universal access problems posed for the broad range of complex web-based systems, including knowledge management systems described elsewhere in this text.

The emergence of web-based systems

The availability of network technology and emergence of the Internet from the late 1970s, saw the early adoption of networked information

systems, particularly within higher education and the information sector. These early systems were often text-based and driven by textual command input (usually via a terminal or telnet-type interface), requiring significant training in use of syntax that differed from system to system. In recent years, the web browser application has provided a more accessible alternative to these older, less intuitive Internet technologies. Furthermore, the growth of the World Wide Web as a medium for commerce and leisure has resulted in the prevalence of web browsing skills across society, allowing organisations to provide networked services with significantly fewer training considerations than in the past.

It is in this context that web-based information/knowledge management systems have recently been deployed across the public, educational and commercial sectors, including a wide range of 'enterprise level' (i.e. organisation-wide) systems, including systems used for library records, student and course records, staff and human resources data, information-querying and user-support systems, finance systems and learning and teaching systems. Many of these systems still rely on traditional relational database management systems (RDBMS) such as Oracle and SQL. However, the user interface for these systems is increasingly web-based and may usually be accessed via any computer connected to the Internet.

Practical examples illustrating the use of a web front-end for information or knowledge management systems could include the following:

- Use of a human resources management system by personnel staff and other administrative staff across an organisation or within a partner organisation located elsewhere in the country.
- Use of a university e-learning system providing course material and communication functions. Course material and collaboration features could be accessed by students within the university or from home. Lecturers could similarly upload new course notes or participate in student discussions from any computer connected to the Internet.
- Use of a content management system to provide a secure staff intranet – including features such as e-mail, discussion tools, online document publishing, knowledge querying or data repository functions from any networked computer.

Definition of web technologies

To understand the World Wide Web and the importance of accessibility for information/knowledge management systems, it is perhaps useful to

have a basic knowledge of the Internet (upon which the Web runs). The Internet has been around a long time, developed during the 1960s by the US military's ARPA or Advanced Research Project Agency (FOLDOC, 2005a). This technology quickly spread to the US academic community and saw the development of the civilian Internet we know today. The early Internet was largely limited to academic and research institutions, with communications software such as e-mail running on complex UNIX-based terminals, which required significant technical skills to operate.

Two major advances allowed for the development of today's World Wide Web. These included the widespread adoption of graphical user interfaces (GUI), such as Microsoft Windows, which provided mouse-based interaction, rather than relying on typed input. In addition, the emergence of the web browser software application (and related technical specifications), developed by Tim Berners-Lee around 1993, revolutionised the way an ordinary computer user could interact with the Internet.

It may be worth mentioning that typical web pages are just simple text files (along with associated files such as images) located on a powerful computer linked to the Internet (called a server, and in this case a web server). In the process of browsing to a web page, the user simply downloads the web page via a web browser application onto their own computer (the client). It may be worth considering certain aspects of the World Wide Web in more detail.

Web pages are more properly known as HTML (Hyper Text Mark-up Language) documents. An HTML file usually ends in '.htm' or '.html' (e.g. mypage.htm) and can be opened in Notepad or any other text editing software. In addition to describing the file type, HTML also describes the language or script comprising the document. This script determines how the document content is displayed and is used to 'mark-up' ordinary text, for example, the HTML 'tag' for a heading is shown below:

```
<h1>Welcome to my web page!</h1>
```

As can be seen above, most HTML tags (or elements) require an 'opening' and 'closing' tag. HTML script can be viewed using Notepad or a similar text editor. However, a web browser application is needed to actually interpret the HTML and display the page content properly; the above example would display as follows in a web browser:

Welcome to my web page!

You can view the HTML for any web page by running your web browser, such as Internet Explorer – browse to a website, then using the pull-down toolbar at the top of the screen to select View/Source.

There are several structural features of HTML essential to the web page, these include the following:

- <html> … </html>: This tag is used to enclose the entire HTML code.
- <head> … </head>: All content in this tag is invisible to the user and occurs at the top of the document, containing the document title tag, descriptive information about the document and document style information.
- <title> … </title>: This is actually inserted inside the <head> area and defines a title for the web page which is not displayed by the web browser, but is used by search engines and for other indexing functions, for example:

  <title>John's Web Page</title>
- <body> … </body>: This tag is used to contain all the content the user will actually see in the web browser, it could include tables, paragraphs, headings and hyperlinks to other resources.

A very basic web page could look as follows in HTML (note the <h1> and <p> tags used for a heading and paragraph respectively):

```
<html>
<head>
<title>John's Web Page</title>
</head>
<body>
<h1>Welcome to John's web page</h1>
<p>This page is all about John…</p>
</body>
</html>
```

Another aspect of the web browser is the use of the URL (uniform resource locator) address to request and retrieve HTML documents. This is shown in the address bar found at the top of the web browser screen, e.g. *http://www.draigweb.co.uk*.

Recent web browsers include Internet Explorer, Netscape Navigator and Opera; they are all based on the original browser developed by Tim Berners-Lee.

Recent years have seen the development of HTML into a more strictly defined mark-up language (XHTML or Extensible Hyper Text Mark-up Language), separating document display properties (such as colour and font face) from document structure (such as headings, paragraphs and other descriptive content). Some of the most widely used versions of HTML (W3C, 2005a) include:

- *HTML 4.01*: The original format for web pages, still a recognised standard.
- *XHTML 1.0*: An early version of Extensible Hyper Text Mark-up Language, permitting some features of HTML while enforcing a stricter document structure.
- *XHTML 1.1*: The most recent HTML standard providing the most highly structured form of HTML.

The main difference between HTML and XHTML concerns 'well-formedness' – i.e. providing appropriate 'closing' tags to maintain integral mark-up and 'nesting' these correctly within the document structure. For example, with Hello the 'bold' tag is closed and 'nested' within a paragraph tag.

XHTML emphasises the removal of display-focused tags, such as <font> and relies on an external style sheet (a file containing style information) to separate display aspects from document structure. The result is a 'vanilla' HTML document composed of purely structural and descriptive information, without the clutter of style instructions embedded in the page itself. The importance of this factor for accessibility will be discussed at later stages in the chapter.

What are web standards?

Central to the discussion of web accessibility and usability is the issue of technical standards – i.e. how to ensure that web pages display as the author intended and in a consistent manner within any standard web browser.

In the early days of the World Wide Web, a number of popular web browsers emerged, each one adhering to varying degrees with the official specifications for HTML. However, many of these early browsers also included their own proprietary HTML features or interpretations.

If the web author used browser-specific (i.e. non-standard HTML) code in their web page (e.g. special code for Netscape Navigator), they could rely on the page displaying correctly in the corresponding web browser. However, when the web page was viewed in a different browser, unintended behaviour could result.

The answer to this lack of consistency in HTML code was basically twofold: to improve the conformance of web browser software to the standard HTML model – the 'Document Object Model' (W3C, 2005b) and to ensure the corresponding HTML (the document code or script used to create web pages) was also regulated by standards. The organisation responsible for these standards is the World Wide Web Consortium or W3C (*http://www.w3c.org*), founded by the inventor of the web browser, Tim Berners-Lee.

There are several key standards integral to web resources, these include:

- *Mark-up languages*: Including the specifications for HTML, XHTML and related versions.
- *Cascading style sheets (CSS)*: Also called 'style sheets' these allow the web author to create a text file containing instructions for the web browser on how to display HTML elements; CSS files end in *.css and are referenced in the <head> area of the HTML document, for example:

  <link rel=stylesheet href="stylesheet.css" type="text/css">

There are currently two CSS specifications – Levels 1 and 2 (W3C, 2005c), either may be used by modern web browsers; version 2 – the most recent version, provides greater scope for stylising document display.

CSS allows the web author to control the visual style of an entire website using a single CSS file, allowing style consistency across all pages. Importantly, CSS allows for separation of HTML display instructions from textual content and structure. Effectively this means the HTML document can be read in a range of agents or web browsing software without the clutter of traditional HTML style tags, such as <font face="arial">. An example CSS instruction to format paragraph text using the colour black, in Arial font at size 10 could appear as follows in the CSS file:

```
p {
font-family: Arial;
font-size: 10pt;
color: #000000;
}
```

The issue of developing standard web browsers is still in process today, but has come very far indeed, with levels of conformance much improved across browsers such as Opera, Netscape Navigator, Mozilla and Internet Explorer. The issue of regulating or standardising HTML document creation is more problematic and relies mostly on the skill, awareness and conscientiousness of IT professionals responsible for HTML authoring. Additionally, web documents can be created in a number of ways, either coding the page 'by hand' to 'mark up' content using HTML tags, or using an automated method such as web editor software (e.g. Dreamweaver) which provides a 'word-processor' style interface for authoring HTML.

While it is impossible to force individual developers to adhere to HTML standards, it is perhaps more achievable to improve the standards compliance of web editor software such as FrontPage and Dreamweaver and also the automated web 'output' of recent web-based information or knowledge management systems. In recent years, web editors such as Dreamweaver have conformed more strictly to HTML specifications, including features to allow the author to audit and correct non-standard HTML or XHTML.

However, it should also be noted that organisations are increasingly moving away from traditional web page development using web editing software (such as Dreamweaver) and are instead adopting systems managed via a web-based interface. These systems, utilising technology such as RDBMS are allowing non-technical staff themselves to input and manage digital resources via the web interface, without the need to work through skilled web developers. However, the proliferation of web-based systems for a wide range of functions has itself prompted a greater focus on the capacity of these systems to provide an accessible and usable interface.

Conformance to web standards among modern information or knowledge management systems delivered via a web-based interface has been mixed and there is still much reliance on the web author or systems developer either to purchase a standards-compliant system or modify system templates to produce a more compliant HTML interface.

Examples of information systems that allow for development of web standards-compliant HTML include the open source (non-commercial) content management systems Plone (*http://www.plone.org*) and Zope (*http://www.zope.org*), which include a templates-based system, effectively allowing these systems to conform easily to web standards.

What is usability?

Usability and accessibility are related but distinct requirements for web-based resources. While usability is of general importance for the entire audience viewing web content, the latter is of particular importance for users with disabilities.

There have been many definitions of web usability, inevitably related to aspects of wider information technology and computer software. The development of GUIs through the Microsoft Windows and Apple Macintosh operating systems has increased the general usability of computing, an area which had previously been the reserve of those familiar with command line interfaces such as UNIX and DOS. The graphical interface used by Windows and similar systems is sometimes described as WIMP (windows, icons, menus and pointers).

While the graphical interface has widened the usability of a wide range of computer applications, including Internet software, such as the web browser, the transformation of computing from textual input to graphical interfaces has itself posed challenges for usability. A good example illustrating this is the change from command line programming (to develop computer software) to visual programming, which involves use of forms, icons and other graphical representations to construct computer software. Visual programming languages such as Visual C++ have posed difficulties for blind programmers familiar with older text-based programming interfaces. The FOLDOC Online Dictionary of Computing comments:

> A visually transformed language is a non-visual language with a superimposed visual representation. Naturally visual languages have an inherent visual expression for which there is no obvious textual equivalent. (FOLDOC, 2005b).

The shift from text to graphical interfaces may appear to suggest an obvious improvement in usability, but the above example illustrates the kind of difficulty presented by GUI systems for some visually impaired users.

Other aspects of usability include basic issues of information technology literacy and assumptions concerning users' technical skills. While the number of households possessing home computers and having private Internet access has increased in recent years – at 37 million Internet users in the UK (Internet World Stats, 2005), there is still debate and concern for a 'digital divide' between this IT literate section of

society and certain groups who possess little or no IT skills. This group typically includes the elderly and less affluent members of society.

Usability and training are also inseparable in this regard, as an awareness of a user audience can enable anticipation of training needs while also ensuring appropriate IT resources are provided.

Basic issues of usability for the Web include aspects of page layout and design, legibility of text, appropriate contrast between textual information and background colours and conformance with web standards to enable user customisation within the web browser (e.g. to increase text size or set a preferred colour scheme).

The terms *usability* and *accessibility* are frequently used interchangeably and there is some debate on how far the two strands should be distinguished at all when implementing standards-based systems (i.e. the concept of web standards as a baseline encompassing usability for all users, regardless of particular needs or disabilities). Certainly, an integrated perspective has been adopted by the World Wide Web Consortium – particularly emphasised through the Web Accessibility Initiative or WAI (*http://www.w3c.org/WAI/*), and the Web Content Accessibility Guidelines 1.0 or WCAG (W3C, 1999a), which incorporate aspects of HTML/XHTML compliance, general usability considerations and support for specific forms of disability.

While later sections will consider general usability issues, it should be noted that the current ethos of accessibility is based firmly on core web standards, which provide a basis for delivering usable information for the entire spectrum of society.

What is accessibility?

Accessibility in a general sense is defined by the *Pocket Oxford Dictionary* (1994) as: 'reachable or obtainable; readily available … easy to understand'. Web accessibility concerns the delivery of online information and services in a context that is universally usable for society, regardless of age, disability or culture. This definition is echoed in the TechDis introduction to accessibility: 'Accessibility is about removing barriers to participation and engagement' (TechDis, 2005).

It may be useful to consider some of the general implications of disability for web browsing and how accessibility can offset access difficulties.

Users with low or no usable vision will rely either on a facility to increase or otherwise modify the textual display, or on an alternative form

of output, such as a 'screen reader' to read content via computer speakers. Display customisation (e.g. to increase text size) is possible in most web browsers and via accessibility features in the Windows operating system (e.g. use of 'Magnifier'). However, some visually impaired users may rely on additional 'assistive technology' in the form of software or equipment to fulfil their requirements. Examples of this kind of software include the Dolphin suite of applications – including a screen reader and Braille output software for use with a Braille reader (an additional peripheral connected to the PC which can output a Braille surface).

Users with cognitive disabilities may require alternative textual information within the web document itself, or use of assistive technology to improve text clarity or size. Individuals with motor-related disabilities, such as cerebral palsy, arthritis or repetitive strain injury (RSI) may rely on alternative input devices such as tracker-balls, soft keyboards or specially developed input devices. Of course, some users may require a combination of these solutions; a more detailed description of issues associated with disabilities is provided in the second part of this chapter.

In terms of the Web and HTML documents, a wide range of techniques and approaches exist to ensure web pages are constructed in a standards-compliant, logical and generally considerate manner that is usable for both general users and those with disabilities, most notably using the W3C's Web Content Accessibility Guidelines 1.0 (W3C, 1999a) for implementing accessible web resources. While it is impossible to develop web output that is optimal for every disability or circumstance, it is indeed possible via web standards to ensure content is usable within a broad range of user 'agents', thus ensuring some degree of accessibility.

Problems associated with web documents

The web browser application is a very forgiving and flexible tool – aspects of the web page such as layout, style, navigation menus, decorative images and the 'hidden' code comprising the web page itself are design factors limited only by the imagination of the web author.

In recent years, it has almost become the norm for web developers to furnish their web pages with special effects, breaking out of the limitations of the basic HTML document to include Flash animations, JavaScript menu systems or other features designed to add a sense of interaction or originality. Furthermore, web designers have traditionally used HTML in a purely pragmatic manner, often using inaccurate or

non-standard HTML on the basis that their page 'works' in whatever web browser they designed it. The possibility that their inaccurate HTML will cause the page to function improperly in a different web browser has only recently become a focus among web authors.

Web browser software and the pages they are used to 'read' (i.e. HTML based documents) present a number of inherent problems particularly relevant to issues of usability and accessibility, these include:

- A wide range of web browsers exist produced by many different companies and not-for profit consortiums (e.g. Netscape, Internet Explorer, Mozilla).
- A wide range of HTML formats exists and are used across the World Wide web (e.g. HTML 4.01, XHTML 1.0, XHTML 1.1).
- A wide range of approaches exist to developing web content (e.g. development via hand coding using a text editor such as Notepad, development using a web editor application such as FrontPage or production of web pages via an automated process within a content management or similar system).
- While web and accessibility standards have been in existence for several years, it has historically been difficult to impose these on any of the above.
- Web browsers are permissive – even a very poorly coded page with HTML errors may 'work' on any particular web browser.
- There is no inherent structure or rule-set to ensure web pages provide logical, consistent or standard forms of site navigation (aside from W3C recommendations). Almost every web page is different and requires the user learn the unique navigation method (if present) and other conventions within each site; the same principle applies to layout, style, hyperlink conventions etc.
- Web browsers interpret web standards differently, i.e. while greater standardisation across web browsers now exists, discrepancies still exist in the interpretation of HTML, CSS, JavaScript etc., forcing vigilant developers to test pages in a wide range of browsers.
- Web browsers use different usability/accessibility conventions, e.g. access key functions are provided to operate the web browser via the keyboard. However, different key-combinations are used for the same functions across different browsers (e.g. Opera, Mozilla), forcing disabled users to learn dozens of new access keys within each browser

used (the same problem applies to software applications such as Adobe Acrobat, Word etc.); other discrepancies include methods for using a local, user-defined style sheet and for setting local text colour, font size and browser magnification.

Mapping W3C standards

Table 5.1 illustrates some of the web and accessibility standards introduced previously, divided into three columns: aspect, specific standards and how assessed. The first aspect, 'web standards', lists the technical and structural specifications used to create web resources, including the various forms of

Table 5.1 **Overview of web standards, usability and accessibility**

Aspect	Specific standards	How assessed?
Web standards	Mark-up languages (e.g. HTML 4.01, XHTML 1.0), cascading style sheets (CSS 1 and 2).	Manual auditing possible (i.e. intensive review of code), but applications exist to audit entirely automatically, generate reports etc.
Usability	Usability and accessibility are reflected in both web and accessibility standards, however, subjective issues also include: ■ clarity of style, layout, font-size, use of colour and contrast; ■ use of consistent, logical and meaningful site navigation options; ■ use of written style, language and conten appropriate for the audience; ■ working hyperlinks (i.e. links not broken).	Mostly manual/subjective auditing.
Accessibility	Web Content Accessibility Guidelines (WCAG 1.0); Section 508 of the US Rehabilitation Act.	Automatic auditing possible for WCAG/US 508, but manual/ subjective auditing is still necessary for some specific aspects of these guidelines.

HTML and CSS (style sheets); the second aspect considers issues of usability and subjective issues for web auditing, the last aspect considers methods used to assess formal accessibility standards.

Web-based accessibility and the law

Disability in the UK – facts and figures

There are currently around ten million registered people disabled in the UK (around one in seven of the population), including around seven million disabled people in the workplace:

> There are approximately 10 million disabled adults in the UK covered by the Disability Discrimination Act, which represents around 18 per cent of the population … Over 6.8 million disabled people are of working age, which represents 19 per cent of the working population. (Employers' Forum on Disability, 2005).

The forms of disability vary from visual and cognitive to motor and mobility disabilities, for a brief breakdown of disability in the UK see the figures below from The Economic and Social Research Council (2005):

- 750,000 wheelchair users registered in the UK;
- 19 per cent of men and 13 per cent of women reported having hearing difficulties;
- 55,000 people registered as deaf;
- 157,000 people registered as blind;
- 1.8 million diabetics in the UK;
- 350,000 people with epilepsy.

Implications of IT and web browsing for disabled users

As the World Wide Web becomes an increasingly prevalent information medium, it is necessary to consider the implications of web browsing for various forms of disability (Table 5.2). In this context it may be worth

Table 5.2 **Overview of disabilities and conditions affecting web browsing**

Disability/condition	General issues for web browsing
Speech and language impairments	Language difficulties may arise due to cerebral palsy, autism or as a result of injury; language difficulties could pose a problem for speech recognition software working via a web interface; similarly, users with difficulties comprehending language may experience problems when listening to multimedia content containing speech. Alternative textual content or use of screen readers (i.e. reading text aloud) may offset some problems faced by users with speech or language impairments.
Visual impairments	Individuals with low vision as opposed to total blindness or unusable vision may require some method to improve the clarity, size or contrast of textual or graphical information. It is important to build web-based resources with some means of resizing HTML output relative to the user's screen resolution or browser settings. This can often be achieved by using relative values for tables, text and other elements (e.g. 50 per cent width as opposed to a fixed size such as 10 pixels). Colour blindness is another issue to consider when designing resources; sufficient contrast between fonts and background colours should be used, although most web browsers allow the user to customise their own display preferences. Blind users should also be considered who may rely on screen-reader software, Braille displays or other assistive technology; these users include individuals legally blind (i.e. having less than 20/200 vision) and those with total blindness – e.g. caused by conditions such as glaucoma or cataracts; depending on the level of blindness, a combination of customised style (e.g. within the browser) or provision of alternative text (i.e. in the place of images) can offset some of the limitations for these users. Additionally, CSS now provides the ability to interact with speech readers (aural CSS), adding stress, changing the speed at which text is read and other means of suggesting an auditory equivalent to the visual features of text, such as bold or italic (see *http://www.w3.org/TR/REC-CSS2/aural.html*).

Table 5.2 **Overview of disabilities and conditions affecting web browsing (*Cont'd*)**

Disability/condition	General issues for web browsing
Mobility/motor limitations	Mobility or motor impairment concerns difficulties involving interaction between the user and input devices such as the mouse or keyboard. Arthritis, muscular dystrophy or repetitive strain injury (RSI) are just a few conditions where the user may not be able to use the mouse or keyboard in a typical manner. A range of methods exists to improve access for these users. For those with severe motor impairments, head or mouth sticks are used as pointing devices, while keyboard shortcuts can assist users with RSI or other conditions affecting the ability to type or use the mouse easily.
Hearing impairments	Any user who is deaf to some degree may encounter difficulties when using multimedia content (e.g. where music or speech is present). Alternative textual content is usually possible for most multimedia formats such as Flash.
Cognitive disabilities	Cognitive disabilities are neurological conditions such as dyslexia (affecting the underlying skills that are needed for learning to read, write and spell), dyspraxia (an impairment or immaturity of the organisation of movement in the way that the brain processes information) and Irlen's syndrome (a problem with how the nervous system encodes and decodes visual information).* Users with cognitive or neurological disabilities such as dyslexia or dyspraxia may have difficulties internalising or comprehending textual content in a typical manner and may require either customisation tools to improve text size, clarity and contrast, or may require alternative textual content. Epilepsy should also be considered when creating web resources, as this condition can involve seizures which can be triggered by flashing or blinking images, these type of animations should generally be avoided (see *http://www.w3.org/TR/WCAG10-CORE-TECHS/#flicker*).

*For further information on these conditions see British Dyslexia Association (2005), Dyspraxia Foundation (2005), Irlen Institute (2005).

defining these disabilities with some of the issues encountered by disabled users (technical solutions for the issues mentioned in the table are listed in the section on the WCAG Guidelines by priority level).

UK legislation

Disability Discrimination Act 1995 and amendments (DDA)

The Act describes the legal status of disability and the responsibilities of organisations to provide equality of provision for users with access difficulties (Home Office, 1995; OPSI, 1995). Core aspects of the legislation are as follows:

- A person is categorised as disabled 'if (they have) a physical or mental impairment which has a substantial and long term adverse effect on (their) ability to carry out normal day-to-day activities' (part 1.1).
- Since 1999, organisations are required to use 'reasonable adjustment' for disabled users in the provision of goods, facilities or services (including non-charging services). Examples include 'access to and use of any place which members of the public are permitted to enter', 'access to and use of means of communication' and 'access to and use of information services'.
- Employers are required to make 'reasonable' adjustment for employees having disabilities. This could include provision of assistive technology or special training.
- Educational organisations (higher and further education) were required to provide a 'disability statement', outlining 'the provision of facilities for education and research made by the institution in respect of persons who are disabled persons'.

Further information on the Act may be obtained at Disability.gov (*http://www.disability.gov.uk/*).

An amendment to the Act was introduced on 1 October 2004 – Rights of Access Goods, Facilities, Services and Premises (OPSI, 2002). This amendment requires that services and facilities provide additional 'reasonable adjustment' for disabled users, such as more accessible building layout. The Disability Rights Commission (*http://www.drc-gb.org*) has issued guidelines for the amendment, citing web-based services as examples (Disability Rights Commission, 2004a).

Special Educational Needs and Disabilities Act (SENDA) 2001

The Act requires educational organisations to employ 'reasonable adjustment' to deliver educational services for disabled users (Home Office, 2001; OPSI, 2001). Core aspects include the following:

- To make provision for disabled users without 'substantial disadvantage in comparison with persons who are not disabled'.
- Institutions must provide 'anticipatory' adjustment for disabled students, i.e. to prepare policies, procedures and resources to support learning and teaching for disabled users as a standard feature of course delivery, i.e. not simply as a response to disabled applicants.
- A range of alternative formats are required for the provision of information, such as alternative colour schemes for colour-blind users.
- 'Reasonable adjustment' is exempt where this undermines academic standards, causes excessive financial difficulty, conflicts with health and safety legislation or adversely affects the education of other students.
- Assistive technology should be made available where appropriate (e.g. screen readers, Braille readers).
- Facilities/staff resources should exist to support disabled users in the use of assistive technology (e.g. disability support team).
- Electronic resources such as web pages and other digital information should be accessible for disabled users.

It can be seen that the above legislation does cite electronic resources and that there is a corresponding legal case to ensure web-based resources are generally accessible. Furthermore, SENDA has required the education sector to apply many aspects of the DDA previously exempt for education. The specific means of implementing accessibility for the Web may at times appear arbitrary to actual W3C standards. However, web standards are mentioned in the recommendations of the Disability Rights Commission for implementing DDA and related laws, particularly in a recent report, *The Web Access and Inclusion for Disabled People* (2004):

> It is important that those who train website developers include standard training modules on disability awareness and the techniques required to translate that awareness into practice. Corresponding modules should be incorporated into any

continuing professional development prescribed. (Disability Rights Commission. 2004b: 39)

Web standards and web accessibility standards

Overview of core standards

Traditionally, a range of proprietary technologies were seen on the World Wide Web, with leading web browsers developing their own proprietary HTML standards. However, in recent years, the W3C has improved cooperation with a wide range of information technology organisations to implement web standards. Core standards developed by the W3C include:

- *CSS (cascading style sheets)*: Traditional HTML provided tags to display textual content (e.g. hello), to control colour and layout (e.g. <font color="000000">). CSS may be used to separate HTML structure and content from appearance. The use of a CSS file means that the web browser or assistive technology agent may parse or understand HTML without the clutter of embedded style. CSS files have a .css extension and the CSS code refers to HTML elements, e.g. the following CSS rule is used to display paragraph text using the colour blue:

  P: {color: #0000ff};

  There are two types of CSS: level 1 and level 2. The latter offers more complex display options. CSS files are linked to HTML-based documents in the HEAD region of the mark-up file, for example:

  <link rel="style sheet" href="mystyle.css" type="text/css"/>

  The specifications defining CSS are available at the W3C site (*http://www.w3.org/Style/CSS/*).
- *HTML*: The basic standard for web pages is HTML. Several versions of HTML have been defined, XHTML being the latest version. The main differences between traditional HTML and XHTML includes the requirement for 'well formed' document structure (the rules defining how HTML elements can be used within the document hierarchy), use of lower-case tags and the need to properly 'nest' all elements (e.g. a correctly 'nested' paragraph which is also bold:

example). HTML standards defined by the W3C include HTML 4.01, XHTML 1.0. and XHTML 1.1. Compliance with standards such as XHTML 1.1 should ensure that web resources may be viewed or accessed by any standard web browser, such as Internet Explorer, or assistive technology. The full HTML and related specifications may be obtained at: *http://www.w3.org/MarkUp/*.

Web Content Accessibility Guidelines 1.0

The W3C's Web Content Accessibility Guidelines (WCAG) 1.0 have become the industry standard for developing accessible web-based resources. Three levels of compliance exist: A (Priority 1 compliance) is the minimal level of compliance required; AA (Priority 1 and 2 compliance) provides increased accessibility; and AAA (Priorities 1, 2 and 3) indicates the highest level of compliance.

The guidelines in s. 508 of the US Rehabilitation Act 1998 (*http://www.section508.gov*) provide another major standard for web resources. Although this standard is US-specific, the 508 guidelines are actually based on the WCAG aspects of standards compliance:

- *Use of TITLE*: The title tag is one of the most important tags in the HTML or XHTML document. This is used by a large number of web browsers to store a 'bookmark' about the web page for later viewing. The title tag is also used by search engines such as Google and other systems for user searching and listing results, e.g. <title>University of Somewhere</title>.
- *ALT tags*: Where images occur in web documents, an alternative textual description could be provided for non-visual users. This alternative text may be 'spoken' by a screen reader or displayed in any text-only context, for example:

  A similar descriptive attribute 'longdesc' can also be used to provide a hyperlink to a web page containing further details of the image, e.g.

  In addition to the above, the 'title' attribute can provide alt-style text for almost any HTML element, e.g. a hyperlink:

  John's Page

- *Tables*: Tables in HTML should be defined to indicate the presence of tabular data; table headers (i.e. the row across the top) should contain a 'TH' tag to define each column; tables used for layout should be avoided where possible. Additionally, a SUMMARY tag may be used to define the purpose of tables, e.g. <table summary="This table contains useful information">.
- *Captions*: Similar to ALT tags, these consist of a textual alternative for more complex multimedia presentations, such as Macromedia Flash MX. Typically, a caption will provide a short text description for audio or video files, which may be displayed onscreen for users with hearing problems.
- *Alternative style sheets*: Several alternative CSS (style sheets) may be available within a web resource, allowing the user to select from a range of available styles to display the document. Not all current browsers support selection of alternative style sheets.
- *Interactive features*: Web forms, check boxes, drop-down menus and other interactive features may present difficulties for disabled users and assistive technology, such as screen readers. Interactive features (e.g. a selection menu) may sometimes be replaced by simple hyperlinks. However, a range of methods exist to ensure interactive features are accessible – e.g. defining a 'tabindex' (to cycle through pull-down options using the tab key), or 'access keys' to use web features without the need for a mouse.

Validation and auditing techniques for web-based systems

Automatic mark-up validation

Systems may be tested using either software installed on a computer or some web-based services. Auditing web resources should be carried out using official W3C tools or software recommended by the W3C:

- *W3C HTML Validator*: The W3C site provides a range of tools for checking the core standards compliance of HTML/XHTML web resources. Web pages containing HTML or XHTML errors may present problems for either standard web browsers or third-party equipment, such as screen readers. There are several validator tools

available at the W3C site, including the W3C MarkUp Validation Service (*http://validator.w3.org*) and the Web Development Group Validator (*http://www.htmlhelp.com/tools/validator*); see also 'A Real Validator': *http://arealvalidator.com/*).

- *W3C CSS Validator* (*http://jigsaw.w3.org/css-validator*): CSS used to control colours and layout may be checked using this tool.

Automatic WCAG validation

- *Bobby*: The Bobby accessibility validator is the most popular tool for auditing WCAG compliance, and also allows for checking US 508 support. Bobby is available as a commercial software application (*http://www.watchfire.com*) and a similar free online tool called WebXact is also available (*http://webxact.watchfire.com*). The Bobby/WebXact tool provides a detailed report of WCAG or US 508 compliance for a given web page.
- *LIFT for Dreamweaver or FrontPage* (*http://www.useablenet.com*): This plug-in for the FrontPage or Dreamweaver web editors provides detailed reports, interactive prompts and other features for developing accessible HTML/XHTML.

Web Content Accessibility Guidelines in detail

WCAG 1.0, defines standards for delivering accessible web resources (see the full WCAG document at *http://www.w3.org/TR/WAI-WEBCONTENT/*).

There are three 'priority' levels of compliance (levels 1, 2 and 3) described on the WCAG site (1999a):

- Conformance Level 'A': all priority 1 checkpoints are satisfied.
- Conformance Level 'Double-A': all priority 1 and 2 checkpoints are satisfied.
- Conformance Level 'Triple-A': all priority 1, 2 and 3 checkpoints are satisfied.

The WCAG site (1999a) defines the priorities as follows:

- *Priority 1*: A web content developer must satisfy this checkpoint. Otherwise, one or more groups will find it impossible to access

information in the document. Satisfying this checkpoint is a basic requirement for some groups to be able to use web documents.

- *Priority 2*: A web content developer should satisfy this checkpoint. Otherwise, one or more groups will find it difficult to access information in the document. Satisfying this checkpoint will remove significant barriers to accessing web documents.
- *Priority 3*: A web content developer may address this checkpoint. Otherwise, one or more groups will find it somewhat difficult to access information in the document. Satisfying this checkpoint will improve access to web documents.

There are 14 WCAG guidelines, each guideline contains several 'checkpoints'; these are also defined according to priority level – e.g. checkpoint '3.3 Use style sheets to control layout and presentation' is also defined as 'priority 2'. Some checkpoints also apply to several priorities under certain circumstances (for a breakdown of checkpoints where these circumstances are cited, see W3C, 1996b).

The WCAG guidelines are grouped under general headings, each one contains a number of detailed subsections; however, the following list provides a useful précis of the guidelines (W3C, 1999a):

1. Provide equivalent alternatives to auditory and visual content.
2. Do not rely on colour alone.
3. Use mark-up and style sheets and do so properly.
4. Clarify natural language usage.
5. Create tables that transform gracefully.
6. Ensure that pages featuring new technologies transform gracefully.
7. Ensure user control of time-sensitive content changes.
8. Ensure direct accessibility of embedded user interfaces.
9. Design for device-independence.
10. Use interim solutions.
11. Use W3C technologies and guidelines.
12. Provide context and orientation information.
13. Provide clear navigation mechanisms.
14. Ensure that documents are clear and simple.

While it is possible to review HTML in detail for WCAG compliance, there are a range of options available for auditing web resources,

such as using Bobby or similar automated methods (as previously described).

In addition, W3C (2000) provides detailed examples illustrating the checkpoints. To discover further examples of how the guidelines are implemented, search for examples on the W3C site (*http://www.w3c.org*) using the site search box to look up the required mark-up element name (e.g. 'alt').

Table 5.3 illustrates the WCAG in some detail. Further examples may be found by following the URLs provided in the second column.

Table 5.3 **WCAG guidelines by priority level**

Checklist item	Checkpoint notes
Priority 1	
1.1 Provide a text equivalent for every non-text element (e.g. via 'alt', 'longdesc', or in element content). This includes images, graphical representations of text (including symbols), image map regions, animations (e.g. animated GIFs), applets and programmatic objects, ASCII art, frames, scripts, images used as list bullets, spacers, graphical buttons, sounds (played with or without user interaction), stand-alone audio files, audio tracks of video, and video.	For example, in HTML: Use 'alt' for the IMG, INPUT, and APPLET elements, or provide a text equivalent in the content of the OBJECT and APPLET elements. For complex content (e.g. a chart) where the 'alt' text does not provide a complete text equivalent, provide an additional description using, for example, 'longdesc' with IMG or FRAME, a link inside an OBJECT element, or a description link. For image maps, either use the 'alt' attribute with AREA, or use the MAP element with A elements (and other text) as content. Refer also to checkpoint 9.1 and checkpoint 13.10.
1.2 Provide redundant text links for each active region of a server-side image map.	Refer also to checkpoint 1.5 and checkpoint 9.1. * See Section: '7.4.4 Server-side image maps' in the WCAG techniques at: *http://www.w3-.org/TR/WCAG10-HTML-TECHS/*
1.3 Until user agents can automatically read aloud the text equivalent of a visual track, provide an auditory description of the important information of the visual track of a multimedia presentation.	Synchronize the auditory description with the audio track as per checkpoint 1.4. Refer to checkpoint 1.1 for information about textual equivalents for visual information.

Table 5.3 **WCAG guidelines by priority level (*Cont'd*)**

Checklist item	Checkpoint notes
1.4 For any time-based multimedia presentation (e.g. a movie or animation), synchronize equivalent alternatives (e.g. captions or auditory descriptions of the visual track) with the presentation.	* Multimedia files such as Flash allow for 'captions' to display text alternatives when displaying audio or video, these captions may be useful for deaf/hard of hearing users.
2.1 Ensure that all information conveyed with color is also available without color, for example from context or mark-up.	* It is unwise to rely on colour to convey information. For example, an important notice displayed in red will not convey importance for blind users, this should be indicated in <B> (bold) or , or should be indicated in plain language in the context of the text.
4.1 Clearly identify changes in the natural language of a document's text and any text equivalents (e.g. captions).	For example, in HTML use the 'lang' attribute. In XML, use 'xml:lang'.
5.1 For data tables, identify row and column headers.	For example, in HTML, use TD to identify data cells and TH to identify headers.
5.2 For data tables that have two or more logical levels of row or column headers, use mark-up to associate data cells and header cells.	For example, in HTML, use THEAD, TFOOT, and TBODY to group rows, COL and COLGROUP to group columns, and the 'axis', 'scope', and 'headers' attributes, to describe more complex relationships among data.
6.1 Organize documents so they may be read without style sheets. For example, when an HTML document is rendered without associated style sheets, it must still be possible to read the document.	When content is organized logically, it will be rendered in a meaningful order when style sheets are turned off or not supported.
6.2 Ensure that equivalents for dynamic content are updated when the dynamic content changes.	* See Section: '8.1' in the WCAG techniques at: *http://www.w3-.org/TR/WCAG10-HTML-TECHS/*

Table 5.3 **WCAG guidelines by priority level (*Cont'd*)**

Checklist item	Checkpoint notes
6.3 Ensure that pages are usable when scripts, applets, or other programmatic objects are turned off or not supported. If this is not possible, provide equivalent information on an alternative accessible page.	For example, ensure that links that trigger scripts work when scripts are turned off or not supported (e.g. do not use 'javascript:' as the link target). If it is not possible to make the page usable without scripts, provide a text equivalent with the NOSCRIPT element, or use a server-side script instead of a client-side script, or provide an alternative accessible page as per checkpoint 11.4. Refer also to guideline 1.
7.1 Until user agents allow users to control flickering, avoid causing the screen to flicker.	Note. People with photosensitive epilepsy can have seizures triggered by flickering or flashing in the 4–59 flashes per second (Hertz) range with a peak sensitivity at 20 flashes per second as well as quick changes from dark to light (like strobe lights).
8.1 Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies	Refer also to guideline 6. * See Section: '8.2' in the WCAG techniques at: *http://www.w3-.org/TR/WCAG10-HTML-TECHS/*
9.1 Provide client-side image maps instead of server-side image maps except where the regions cannot be defined with an available geometric shape.	Refer also to checkpoint 1.1, checkpoint 1.2, and checkpoint 1.5. * See Section: '7.4.3' in the WCAG techniques at: *http://www.w3-.org/TR/WCAG10-HTML-TECHS/*
11.4 If, after best efforts, you cannot create an accessible page, provide a link to an alternative page that uses W3C technologies, is accessible, has equivalent information (or functionality), and is updated as often as the inaccessible (original) page.	* For example, provide a static HTML alternative if you cannot make a script generated page accessible. It may be necessary to update the alternative page manually to ensure content is complementary to the script generated page.
12.1 Title each frame to facilitate frame identification and navigation.	For example, in HTML use the 'title' attribute on FRAME elements.
14.1 Use the clearest and simplest language appropriate for a site's content.	* Use concise language for navigation/ menus, with hyperlink labels describing the content of actual resources as closely as possible.

Table 5.3 **WCAG guidelines by priority level (*Cont'd*)**

Checklist item	Checkpoint notes
Priority 2	
2.2 Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen.	* See Section: '7.5' in the WCAG techniques at: *http://www.w3-.org/TR/WCAG10-HTML-TECHS/*
3.1 When an appropriate mark-up language exists, use mark-up rather than images to convey information.	For example, use MathML to mark up mathematical equations, and style sheets to format text and control layout. Also, avoid using images to represent text – use text and style sheets instead. Refer also to guideline 6 and guideline 11.
3.2 Create documents that validate to published formal grammars.	For example, include a document type declaration at the beginning of a document that refers to a published DTD (e.g. the strict HTML 4.0 DTD).
3.3 Use style sheets to control layout and presentation.	For example, use the CSS 'font' property instead of the HTML FONT element to control font styles.
3.4 Use relative rather than absolute units in mark-up language attribute values and style sheet property values.	For example, in CSS, use 'em' or percentage lengths rather than 'pt' or 'cm', which are absolute units. If absolute units are used, validate that the rendered content is usable (refer to the section on validation).
3.5 Use header elements to convey document structure and use them according to specification.	For example, in HTML, use H2 to indicate a subsection of H1. Do not use headers for font effects.
3.6 Mark up lists and list items properly.	For example, in HTML, nest OL, UL, and DL lists properly.
3.7 Mark up quotations. Do not use quotation markup for formatting effects such as indentation.	For example, in HTML, use the Q and BLOCKQUOTE elements to markup short and longer quotations, respectively.

Table 5.3 **WCAG guidelines by priority level (*Cont'd*)**

Checklist item	Checkpoint notes
5.3 Do not use tables for layout unless the table makes sense when linearized. Otherwise, if the table does not make sense, provide an alternative equivalent (which may be a linearized version).	Note. Once user agents support style sheet positioning, tables should not be used for layout. Refer also to checkpoint 3.3.
5.4 If a table is used for layout, do not use any structural markup for the purpose of visual formatting.	For example, in HTML do not use the TH element to cause the content of a (non-table header) cell to be displayed centred and in bold.
6.4 For scripts and applets, ensure that event handlers are input device-independent.	* See Section: '8.2' in the WCAG techniques at: *http://www.w3-.org/TR/WCAG10-HTML-TECHS/*
6.5 Ensure that dynamic content is accessible or provide an alternative presentation or page.	For example, in HTML, use NOFRAMES at the end of each frameset. For some applications, server-side scripts may be more accessible than client-side scripts.
7.2 Until user agents allow users to control blinking, avoid causing content to blink (i.e. change presentation at a regular rate, such as turning on and off).	* See Section: '8.2' in the WCAG techniques at: *http://www.w3-.org/TR/WCAG10-HTML-TECHS/*
7.3 Until user agents allow users to freeze moving content, avoid movement in pages.	When a page includes moving content, provide a mechanism within a script or applet to allow users to freeze motion or updates. Using style sheets with scripting to create movement allows users to turn off or override the effect more easily. Refer also to guideline 8.
7.4 Until user agents provide the ability to stop the refresh, do not create periodically auto-refreshing pages.	For example, in HTML, do not cause pages to auto-refresh with 'HTTP-EQUIV=refresh' until user agents allow users to turn off the feature.
7.5 Until user agents provide the ability to stop auto-redirect, do not use markup to redirect pages automatically. Instead, configure the server to perform redirects.	* See Section: '12.6' in the WCAG techniques at: *http://www.w3-.org/TR/WCAG10-HTML-TECHS/*

Table 5.3 **WCAG guidelines by priority level (*Cont'd*)**

Checklist item	Checkpoint notes
8.1 Make programmatic elements such as scripts and applets directly accessible or compatible with assistive technologies	Refer also to guideline 6. * See Section: '12.4' in the WCAG techniques at: *http://www.w3-.org/TR/WCAG10-HTML-TECHS/*
9.2 Ensure that any element that has its own interface can be operated in a device-independent manner.	Refer also to guideline 8. * See Section: '8.2' in the WCAG techniques at: *http://www.w3-.org/TR/WCAG10-HTML-TECHS/*
9.3 For scripts, specify logical event handlers rather than device-dependent event handlers.	* See Section: '12.4' in the WCAG techniques at: *http://www.w3-.org/TR/WCAG10-HTML-TECHS/*
10.1 Until user agents allow users to turn off spawned windows, do not cause pop-ups or other windows to appear and do not change the current window without informing the user.	For example, in HTML, avoid using a frame whose target is a new window.
10.2 Until user agents support explicit associations between labels and form controls, for all form controls with implicitly associated labels, ensure that the label is properly positioned.	The label must immediately precede its control on the same line (allowing more than one control/label per line) or be in the line preceding the control (with only one label and one control per line). Refer also to checkpoint 12.4.
11.1 Use W3C technologies when they are available and appropriate for a task and use the latest versions when supported.	* For general accessibility information and related standards, see the Web Accessibility Initiative (WAI): *http://www.w3c.org/WAI/*
11.2 Avoid deprecated features of W3C technologies.	For example, in HTML, don't use the deprecated FONT element; use style sheets instead (e.g. the 'font' property in CSS).
12.2 Describe the purpose of frames and how frames relate to each other if it is not obvious by frame titles alone.	For example, in HTML, use 'longdesc,' or a description link.
12.3 Divide large blocks of information into more manageable groups where natural and appropriate.	For example, in HTML, use OPTGROUP to group OPTION elements inside a SELECT; group form controls with FIELDSET and LEGEND; use nested lists where appropriate; use headings to structure documents, etc. Refer also to guideline 3.

Table 5.3 **WCAG guidelines by priority level (*Cont'd*)**

Checklist item	Checkpoint notes
12.4 Associate labels explicitly with their controls.	For example, in HTML use LABEL and its 'for' attribute.
13.1 Clearly identify the target of each link.	Link text should be meaningful enough to make sense when read out of context – either on its own or as part of a sequence of links. Link text should also be terse. For example, in HTML, write 'Information about version 4.3' instead of 'click here'. In addition to clear link text, content developers may further clarify the target of a link with an informative link title (e.g. in HTML, the 'title' attribute).
13.2 Provide metadata to add semantic information to pages and sites.	For example, use RDF to indicate the document's author, the type of content, etc. Note. Some HTML user agents can build navigation tools from document relations described by the HTML LINK element and 'rel' or 'rev' attributes (e.g. rel="next", rel="previous", rel="index", etc.). Refer also to checkpoint 13.5.
13.3 Provide information about the general layout of a site (e.g. a site map or table of contents).	In describing site layout, highlight and explain available accessibility features.
13.4 Use navigation mechanisms in a consistent manner.	* Top-level navigation options (i.e. main categories as opposed to sub-categories) should ideally remain the same across all documents, providing a consistent method to navigate.
Priority 3	
1.5 Until user agents render text equivalents for client-side image map links, provide redundant text links for each active region of a client-side image map.	Refer also to checkpoint 1.2 and checkpoint 9.1. * See Section: '7.4.2' in the WCAG techniques at: *http://www.w3.org/-TR/WCAG10-HTML-TECHS/*
2.2 Ensure that foreground and background color combinations provide sufficient contrast when viewed by someone having color deficits or when viewed on a black and white screen.	Refer also to checkpoint 1.2 and checkpoint 9.1. * See Section: '7.5' in the WCAG techniques at: *http://www.w3-.org/TR/WCAG10-HTML-TECHS/*

Table 5.3 **WCAG guidelines by priority level (*Cont'd*)**

Checklist item	Checkpoint notes
4.2 Specify the expansion of each abbreviation or acronym in a document where it first occurs.	For example, in HTML, use the 'title' attribute of the ABBR and ACRONYM elements. Providing the expansion in the main body of the document also helps document usability.
4.3 Identify the primary natural language of a document.	For example, in HTML set the 'lang' attribute on the HTML element. In XML, use 'xml:lang'. Server operators should configure servers to take advantage of HTTP content negotiation mechanisms.
5.5 Provide summaries for tables.	For example, in HTML, use the 'summary' attribute of the TABLE element.
5.6 Provide abbreviations for header labels.	For example, in HTML, use the 'abbr' attribute on the TH element.
9.4 Create a logical tab order through links, form controls, and objects.	For example, in HTML, specify tab order via the 'tabindex' attribute or ensure a logical page design.
9.5 Provide keyboard shortcuts to important links (including those in client-side image maps), form controls, and groups of form controls.	For example, in HTML, specify shortcuts via the 'accesskey' attribute.
10.3 Until user agents (including assistive technologies) render side-by-side text correctly, provide a linear text alternative (on the current page or some other) for all tables that lay out text in parallel, word-wrapped columns.	Note. Please consult the definition of linearized table. This checkpoint benefits people with user agents (such as some screen readers) that are unable to handle blocks of text presented side-by-side; the checkpoint should not discourage content developers from using tables to represent tabular information.
10.4 Until user agents handle empty controls correctly, include default, place-holding characters in edit boxes and text areas.	For example, in HTML, do this for TEXTAREA and INPUT.
10.5 Until user agents (including assistive technologies) render adjacent links distinctly, include non-link, printable characters (surrounded by spaces) between adjacent links.	* See Section: '6.2' in the WCAG techniques at: *http://www.w3-.org/TR/WCAG10-HTML-TECHS/*

Table 5.3 **WCAG guidelines by priority level (*Cont'd*)**

Checklist item	Checkpoint notes
11.3 Provide information so that users may receive documents according to their preferences (e.g. language, content type, etc.)	* Provide alternative formats, e.g. where a Word file is used, also provide an alternative HTML version.
13.5 Provide navigation bars to highlight and give access to the navigation mechanism.	Provide information so that users may receive documents * Use a menu of links to provide consistent navigation across all pages.
13.6 Group related links, identify the group (for user agents), and, until user agents do so, provide a way to bypass the group.	* See Section: '6.2' in the WCAG techniques at: *http://www.w3-.org/TR/WCAG10-HTML-TECHS/*
13.7 If search functions are provided, enable different types of searches for different skill levels and preferences.	* For example, provide an 'advanced' option for website search engines.
13.8 Place distinguishing information at the beginning of headings, paragraphs, lists, etc.	Note. This is commonly referred to as 'front-loading' and is especially helpful for people accessing information with serial devices such as speech synthesizers.
13.9 Provide information about document collections (i.e. documents comprising multiple pages.).	For example, in HTML specify document collections with the LINK element and the 'rel' and 'rev' attributes. Another way to create a collection is by building an archive (e.g. with zip, tar and gzip, stuffit, etc.) of the multiple pages. Note. The performance improvement gained by offline processing can make browsing much less expensive for people with disabilities who may be browsing slowly.
13.10 Provide a means to skip over multi-line ASCII art.	* See Section: '7.3' in the WCAG techniques at: *http://www.w3-.org/TR/WCAG10-HTML-TECHS/*

Table 5.3 WCAG guidelines by priority level (*Cont'd*)

Checklist item	Checkpoint notes
14.2 Supplement text with graphic or auditory presentations where they will facilitate comprehension of the page.	Refer also to guideline 1.
14.3 Create a style of presentation that is consistent across pages.	* Do not change the font size or layout style across a website but ensure style is consistent across all pages (i.e. using CSS.).

Source: The table is derived from the Guidelines on the W3C site (*http://www.w3.org/TR/WCAG10/full-checklist.html)*, with additional notes from the author indicated with an asterisk.
Note: this table also appeared in Catherall (2004).

Further possibilities for auditing web resources

A range of additional methods exist to audit pages for accessibility – ideally, web documents should function using a wide variety of web browsers and using alternative input devices (e.g. keyboard only); suggestions provided by the W3C (1999a) include:

- Use a text-only browser or emulator (e.g. Lynx: *http://lynx.browser.org/*). It is important to note that most screen-reader software relies heavily on the actual text comprising the HTML document; if your page functions in a text-only browser such as Lynx, there is a good chance it will function successfully using assistive technology.
- Use multiple graphic browsers, with:
  - sounds and graphics loaded,
  - graphics not loaded,
  - sounds not loaded,
  - no mouse,
  - frames, scripts, style sheets and applets not loaded.
- Use several browsers, old and new.
- Use a self-voicing browser, a screen reader, magnification software, a small display (including a range of screen resolution settings, including 800 × 600 pixels per inch and higher resolutions), etc.

- Use spell and grammar checkers. A person reading a page with a speech synthesiser may not be able to decipher the synthesiser's best guess for a word with a spelling error. Eliminating grammar problems increases comprehension.
- Review the document for clarity and simplicity. Readability statistics, such as those generated by some word processors, may be useful indicators of clarity and simplicity. Better still, ask an experienced (human) editor to review written content for clarity. Editors can also improve the usability of documents by identifying potentially sensitive cultural issues that might arise due to language or icon usage.
- Invite people with disabilities to review documents. Expert and novice users with disabilities will provide valuable feedback about accessibility or usability problems and their severity.

Further accessibility tools are listed at the W3C accessibility resources site (*http://www.w3.org/WAI/ER/existingtools.html*).

Usability and the Web

Basic usability issues

Usability is a term used widely and interchangeably with 'accessibility', but perhaps the best definition of usability for web-based systems concerns how intuitive these systems are in terms of navigation and the user interface.

The interface used to access web resources should be designed for ease of access, with a clearly defined navigation menu that provides a persistent route to areas within the resource (i.e. links to core pages or functions persistently onscreen). Links within the interface should be defined using concise and relevant language (i.e. the name of menu items should relate to the content that will be accessed).

The computer 'platform' or system specifications are also an important consideration for delivering web resources. These include the operating system used to run software, such as Microsoft Windows 2000 or Windows XP, the web browser used, e.g. Netscape Navigator 7, and the display resolution available to view web resources. A controlled and standardised platform environment is possible on computers within a defined organisation – e.g. delivering a web-based staff intranet on computers with Windows XP using Internet Explorer and a screen

resolution of 1024 × 768 pixels. However, where home Internet access is required, it may be difficult to ensure a standard platform, because not all home users will possess computers capable of running the latest operating system or web browser.

While it is possible to issue guidelines for users accessing a particular web resource, care should be taken to ensure the resource is functional across a range of environments (e.g. non-Windows operating systems such as Apple Macintosh and a range of web browsers).

The W3C site provides detailed statistics on the kind of platforms used to access web resources. Currently, the most widely used web browser is Internet Explorer version 6 (with around 70 per cent using this browser), with other leading browsers including Internet Explorer 5, Opera 7 and Netscape variants. According to W3C (2005d), 'Internet Explorer 6 is the dominating browser, XP is the most popular operating system, and most users are using a display with 800 × 600 pixels or more, with a color depth of at least 65K colors'.

Navigation and content structure

The navigation menu or links provided by web resources should consist of simple hyperlinks rather than complex script-generated buttons or roll-over images. Text rather than images should be used for interface navigation. Images or icons used to access information should be clearly defined, as the meaning of icons can be ambiguous.

Bandwidth and file size considerations

Bandwidth is the capacity of a network to transfer data at particular speeds. All web pages and supporting files such as image files must be downloaded onto the user's computer for viewing. The larger the file size (e.g. in kilobytes), the longer it takes to download. For organisations with high-speed Internet access, file size is not usually a problem, but Internet connection speed may be a problem for home users connecting to the Internet via a 56k modem. Use of high-resolution images or content-rich documents such as PowerPoint presentations should be carefully considered.

Scripts and client-side issues

It is important to consider the impact of extensive use of JavaScript or other 'inline' code within the normal HTML or XHTML document. Use

of JavaScript listed within the actual HTML document can clutter or seriously impair the interpretation of HTML by screen readers and other assistive technology. Alternatives to the traditional method of listing scripts within the HEAD of the HTML document include use of a JavaScript 'include' to refer to a script contained in an external file or by providing an alternative <noscript> tag to provide alternative content for agents unable to render the script, for example:

> <noscript>Follow this link for an alternative page</noscript>

While there has been a proliferation of scripts such as use of 'roll-over' buttons for navigation, it is important to consider the impact of graphical or script-based elements for users of assistive technology or text-based browsers unable to use JavaScript.

Navigation links should consist of basic HTML hyperlinks with clearly defined labels (i.e. the text the user clicks on, e.g. Link).

Use of JavaScript actions such as 'onclick' to activate hyperlinks should be avoided or alternative simple hyperlinks should be made available. However, it is possible to construct highly accessible navigation menus using relatively simple hyperlinks, while incorporating graphical effects via CSS, thus avoiding the use of JavaScript or other complex methods. (*On a slightly technical note:* the use of 'display: block' in an external CSS file can produce a roll-over effect when applied to a div containing a hyperlink, this method can be elaborated to include the border colour of the hyperlink to produce a virtual button effect – the link itself remains a simple hyperlink.)

Application formats and alternative file types

It is often useful to consider providing web resources via a range of different file formats, e.g. in both HTML and Word. Free viewers are available for the Microsoft Office Suite (containing Word, Excel, PowerPoint etc.) and for Adobe Acrobat (the PDF or Portable Document Format reader). However, the file size of these viewers are fairly large for downloading over a slow modem connection, with the result that some users may have difficulty viewing application-specific documents such as PDF or Word. Provision of any document in an alternative HTML format should overcome accessibility difficulties associated with application specific document formats.

The role of assistive technology

General concepts and system features

A range of software exists to facilitate access to web resources for disabled users; in many cases, ordinary web browsers may be used as an accessibility aid, where standard browser features may enhance the user experience. Additionally, specialist web browsers exist to provide added accessibility features (e.g. speech functions to 'read' the web page).

Where a conventional web browser is not sufficient, third-party equipment may be required, such as a Braille display machine for blind users. In addition to the above, the core operating system (such as Microsoft Windows XP) also provides useful tools for accessing digital or online resources such as the screen reader Narrator and Magnifier tools.

Overview of assistive technology software, operating systems and web browsers

Microsoft Windows XP

The XP operating system provides a range of accessibility tools, including Magnifier, an onscreen keyboard to access keys via an alternative input control (such as a joystick or mouse) and the Narrator tool to 'read' documents. Other access features include 'sticky keys' to access Windows features using key combinations pressed incrementally. Additionally, Windows colours may be customised to provide high contrast or custom colour schemes.

Microsoft Internet Explorer 6

Internet Explorer allows the user to disable style sheets provided with remotely accessed web pages and apply their own custom style sheet to the browser (e.g. to set font colour, typeface, background colour). A range of 'access keys' are also provided for users with motor/mobility problems, e.g. pressing the 'backspace' key takes the user back to the previous location (URL), whereas 'tab' moves to the next element onscreen.

Netscape Navigator 7

Netscape provides a 'text zoom' feature to increase text size; fonts and colours may also be set within browser preferences. Netscape supports

selection of multiple style sheets if available within the web page (e.g. a 'text only' style sheet disabling colour or images) – these styles may be selected using a basic pull-down menu. A range of keyboard shortcuts is also provided.

Screen readers

Screen reader software such as JAWS and HAL provides additional access for users with vision problems to 'read' web resources. The latest version of JAWS provides support for a range of languages and integrates with Internet Explorer.

Screen magnifiers

Screen magnifiers provide screen or text magnification software to increase the appearance of the display, including images and text; examples include Supernova and Lunar.

Refreshable Braille readers

These systems are third-party components that work alongside a typical computer, providing a dynamically generated Braille document (e.g. using raised pins beneath a flexible surface). Both HAL and Supernova support refreshable Braille displays.

Conclusion

This chapter has sought to offer an insight into the problems posed by web-based systems for users with access difficulties and for the general usability of web resources. While the World Wide Web has contributed to the massive growth in computer usage in the home, education and industry, it can be seen that this new communications medium poses new challenges for accessibility and usability.

The guidelines established by the W3C have provided an effective approach to auditing and developing accessible web content, facilitated by automatic and manual validation methods. In recent years, these standards have been supported by leading disability organisations such as the Disability Rights Commission (*http://www.drc-gb.org*) and the Royal National Institute for the Blind (*http://www.rnib*).

It is hoped that this chapter has illustrated some of the broad issues in addressing general web usability and accessibility for disabled users. However, it should be noted that the primary goal of web development should be to develop standards compliant HTML, as valid HTML or XHTML is essential for web agents such as web browsers or assistive technology to correctly interpret and render web content. The ultimate goal, increasingly required under UK law and the statutes of other countries requires compliance with the Web Content Accessibility Guidelines at the minimum of level 1, but ideally at higher levels such as 2 or 3.

At the time of writing this document, responsibility for delivery of standards-compliant web resources (or selection of pre-developed web-based systems meeting these criteria) rests largely with individual web developers or other IT professionals. However, the onus is increasingly on companies or organisations concerned with the development of information/knowledge management systems to embrace web and accessibility standards, as organisations move increasingly away from incidental website development to use of enterprise-level systems delivered via the web browser.

To conclude, consider the words of Tim Berners-Lee, the inventor of the World Wide Web, speaking at Cambridge, Massachusetts in 1997:

> Worldwide, there are more than 750 million people with disabilities. As we move towards a highly connected world, it is critical that the Web be usable by anyone, regardless of individual capabilities and disabilities... (W3C, 1997)