ACE Centre Advisory Trust

Making software inclusive and digital publications accessible

Guidelines for software developers and publishers

Andrew Lysley
David Colven

October 2005

acecentre
advisory trust

Acknowledgments
With thanks to Stephen Druce & Pat Walsh (ACE Centre), Mats Lundalv (Dart, Sweden), Simon Judge (ACT, Birmingham), and Tim Edmonds (UKtesters) for their helpful suggestions and comments.

First edition published December 2002
Revised October 2005

David Colven and Andrew Lysley
ACE Centre Advisory Trust

The ACE Centre Advisory Trust
92 Windmill Road, Headington, Oxford OX3 7DR
Tel: 01865 759800
Fax: 01865 759810
email: info@ace-centre.org.uk
www.ace-centre.org.uk
Registered charity No 1040868

The ACE Centre Advisory Trust is a registered charity, partly funded by the DfEE via BECTa, and The Gatsby Charitable Foundation.

Contents

Foreword: Access to educational software for all?

These guidelines focus on *access* to the Internet, multimedia educational resources and other software. The implications of having access to this technology are far-reaching for the improvement in the educational experience for children and others with disabilities. In general, the issues relating to access raised in these guidelines apply to all computer, web-based applications and other electronic publications. The recommendations presented can be applied broadly to all educational software development as well as to other electronic publications and software. The document is an edited and updated version of an earlier publication[1].

Executive summary

Why is accessibility important?

Information and Communication Technology (ICT) or Information and Learning Technology (ILT) can give students with a whole range of disabilities independent access to information and learning, not to mention fun. Furthermore, accommodating users with specific needs will improve the design and user interfacing for everyone.

Different needs

There is a whole range of different disabilities that need to be considered. These include people with:

- Motor impairments
- Communication and Language Difficulties
- Hearing impairment
- Visual impairment

People who have these impairments benefit greatly from often only modest application customisation and good navigation design. These user groups include people who:

- Can only use a keyboard
- Cannot access any kind of external keyboard or pointing device
- Have no sight
- With visual perceptual difficulties
- Have no hearing
- With some hearing loss.

Among these users there will be some who have moderate to severe language difficulties.

What can you do?

Accessible software, web-based materials and other resources should support the user's choice of output methods including any software devices. It should also be consistent with the user's choice of system behaviour, colours, fonts sizes, and keyboard settings along with other alternative output options that different users may require. For example applications

and web sites should allow text to be saved on request by a user so that it can be read later via the end user's specialist utilities.

The overall mantra is **Keep It Simple**. The more complex, diverse and non-standard an application or resource becomes, the more difficult it will be for many users to access. A recent survey from Microsoft has shown that in fact the *large majority* of users will at some time in their lives benefit from what may be termed alternative access.

Legal obligations

All development environments are now addressing the needs of the disabled community particularly following the Americans with Disabilities Act (ADA section 508) and the British Disability Discrimination ACT (DDA) 1995 and its supplementary Special Educational Needs Discrimination ACT (SENDA) 2001. These make it a *legal necessity* for producers of ICT or ILT materials to make reasonable efforts to include people with disabilities.

How this document helps

To help software and resource developers enable access for these users we have outlined some strategies that should be implemented. These include enabling existing alternative technologies to utilise standard system calls, creating alternatives to sounds and pictures, providing simpler language alternatives and allowing customisation of resources.

By following the strategies outlined in this document software and resource developers can improve the educational opportunities for a large number of children. Indeed they can be provided with access to a whole new world of knowledge, a world that has been denied them until this new technology became available.

Introduction

Multimedia material, whether supplied on CD, DVD or via Intra/Internet services is now an essential element of educational technology and widely used in mainstream society. The emergence of the interactive whiteboard as a central classroom tool brings ICT into the heart of the modern classroom but it has also added to the complexity of user interaction with multimedia resources. There is, therefore, a pressing need to consider ways of providing consistent and efficient access to these modern technologies for people with complex physical, sensory and/or learning difficulties. However, there are a number of barriers that prevent users from gaining access to these materials. This document sets out to explain what these barriers are, and to suggest ways that software can be designed to minimise the problems and obstacles that confront people with disabilities.

There is much activity on the Internet involving initiatives to increase the accessibility of the Web to people with disabilities. The World Wide Web Consortium's - W3C (**www.w3.org**) has a commitment to lead the Web to its full potential, including the promotion of a high degree of "usability" (not a term to be confused with "accessibility"- see Appendix 4) for people with disabilities. In particular, The Web Accessibility Initiative - WAI (**www.w3.org/WAI**), in coordination with other organizations around the world, pursues accessibility of the Web through five primary areas of work: technology; guidelines; tools; education and outreach; and research and development.

Of course, it is very difficult if not impossible to create multimedia resources and user interfaces that address all the needs of every user with a complex disability. However, some simple, generic modifications to software and user interfaces will enable most applications to be used by the large majority of end users with complex disabilities. In doing so, their disabilities are not turned into additional handicaps.

Who this document is intended for

These guidelines represent a 'wish list' of ways in which software publishers and developers, including Web site developers, can practically support people with special ICT needs to enjoy improved access to the world of educational technology. The information is intended as a general introduction for any developer wishing to understand and overcome the problem of accessibility and usability. It is not intended to be machine or operating system specific,

although there are important differences in the range of accessibility utilities available from computer and system manufacturers. (See Appendix 5).

Basic principles

In order for the potential of users with disabilities to be realised, it is necessary, wherever possible, for multimedia resources to be compatible with the various enabling technologies used by these disabled users. In some cases this may not be technically feasible. We have included some user case studies in boxes in this document.

By designing with obligations to inclusion in mind (see box below) and implementing the few simple guidelines contained in this document, multimedia publishers can optimise the enabling process for people with disabilities. Multimedia software can *easily* be written to an open format in such a way as to allow assistive devices and AT/AAC systems (see Appendix 4 for a definition of these terms) to work alongside. We are not suggesting that specific code be written but rather that all programs should be designed to allow accessibility and, thereby, enable usability. It is much more a matter of *NOT creating the code that excludes accessibility* and thereby usability. Furthermore, by incorporating inclusive principles in the early stages of design and development, additional cost and complexity can be avoided.

It's a "win, win" formula!

See also

DDA - Disability Discrimination Act, 1995 - **www.direct.gov.uk/disability/** or **www.opsi.gov.uk/acts/acts1995/1995050.htm** and the subsequent SENDA - Special Educational Needs and Disability Act 2001 - **www.ukcle.ac.uk/directions/issue4/senda.html**

Different students' needs

The following sections outline some of the challenges faced by people with different disabilities. The selection of disabilities addressed is not designed to be exhaustive and it should also be remembered that many people with disabilities will fall into more than one of the categories.

Many have restricted access to conventional materials such as books and worksheets. The "accessible computer" with its "accessible software" is, therefore, an instrument that can offer exciting opportunities, broaden horizons, and raise expectations.

It is important that standard applications should enable the Assistive Technologies used by many people with disabilities to function properly (or at least *not disable* them altogether as is sometimes the case!).

Such assistive technologies include:

- Switch access software – allowing computer access through a single switch activated by the user
- Keyboard and mouse modifiers (enabled via accessibility control panels or otherwise), alternative keyboards both large and small, and devices such as joysticks and head or eye controlled systems.
- On-screen keyboards/grids and control provided by supplementary programs running simultaneously with the application (for example SAW, Clicker, Discover Switch, Ezkeys, WiViK, The Grid etc...)
- Synthetic & digitised speech that reads content or input in the form of text, icons, & pictures. This may be in the form of extra hardware speech synthesis or software 'text - to - speech'.
- Screen magnifiers and modifiers
- Word lists, word prediction, abbreviation expansion and other ways of accessing writing wore rapidly.

Alternative web browsers and interfaces providing for example symbol support or embellishment of text for poor readers (WWAAC Web Browser - **www.wwaac.org** & Communicate: Webwide - **www.widgit.com/products/webwide/trial.htm**)

Physical disability

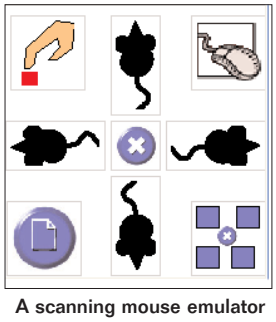
Nearly all programs are now designed to work in windowing environments where controls are selected by moving a mouse pointer and clicking a button or pointing to a whiteboard. The standard interface to a computer is through a keyboard and pointing device - many people with physical disabilities are unable to use these standard devices. They may, for example, have too small a range of movement to operate a mouse, lack the strength to press the mouse button, be unable to press the keys accurately on a keyboard, have poor control of the pointer or be unable to press the button while holding the mouse steady.

People with disabilities, have a wide range of needs. For example, they may have uncontrolled hand movements or tremors that prevent the precise manipulation needed to guide a mouse pointer across the screen. Instead, they depend on keyboards (whether standard or specialised), or special switches for accessing computer technology (see Peter right). By contrast, others with minimal movements (for example, those with muscular dystrophy or motor neurone disease - ALS) often cannot access keyboards easily. They, therefore, rely on alternatives such as on-screen keyboards (they may point and click on these with an optical head-pointing or eye tracking device) or voice recognition systems, which in turn interface and communicate with standard applications.

Some physically disabled students may also have additional cognitive and/or visual perceptual difficulties, perhaps as a consequence of their physical disabilities.

Peter

Peter is a bright 18 year-old who has severe athetoid cerebral palsy, which prevents him from using either a pointing device or a keyboard. His speech, although understandable to most people, is poor and slow. He therefore, tends to be economical with words. His only means of access to his portable PC is by means of a single external switch with which he controls Windows applications such as Microsoft Works and Corel Draw.



A scanning mouse emulator

Peter has benefited enormously from access to a computer since, for the first time, he is able to read e-books without help from an assistant in turning the pages, search the Internet for information, and enjoy the independence of e-banking and e-shopping. However, in order to enable him to do this effectively (without spending most of his time steering a mouse pointer around the screen using simplistic and conventional mouse emulation techniques), the application's functions, keywords and buttons have to be easily accessible to his switches. This can be accomplished (with the minimum of effort and cost) by simple modifications to the application's software code.

Without the mainstream application "making some effort" to accommodate Peter, he will remain where he is now, largely dependent on his helper to do things for him and to try to anticipate many of his more complex thoughts and questions. How much better it would be for his independence, dignity, and quality of life if some of this dependency were removed by implementing just a few, yet very significant improvements to multimedia technology as described in these guidelines.

Visual impairment

Visually impaired students may be either partially sighted or blind and a standard Graphical User Interface (GUI) will be difficult or even impossible to use. Many blind and partially sighted learners are thereby denied effective access to essential print materials. This includes dictionaries, encyclopedias and books in general. Although it may be possible in some cases to use low- vision aids for reading text, it is often the retrieval of digital information that is the greatest barrier to independence. In this context the publication of information in an electronic format is of particular interest. By definition digitally produced materials are designed for use with a computer. As computers can be made accessible to blind and partially sighted learners through specialist "enabling technologies" such as 'text-to-speech', Braille and large-character and high contrast displays, there is potential for making most standard applications accessible.

Ann

Some partially sighted students have a reading speed of about 40 words a minute. (the normal reange can be up to 200 words per minute or more)This can sometimes affect their concentration.

Ann, for example, can read more comfortably from a computer screen that is set to white on black than from large printed text. She is studying English at university and has found the Complete Works of Shakespeare available on CD-ROM to be very helpful. She takes advantage of the CD's search tools and index menus to check cross-references and it is much more convenient than using a magnifier to read all Shakespeare's plays.

Hearing impairment

This can cover everything from profound deafness to slight or fluctuation hearing loss such as 'glue ear'. People who are born profoundly deaf, or who become profoundly deaf before they acquire spoken language, find it more difficult to lip-read for example. Many hearing impaired people use sign language but this has a different grammar from English and American Sign Language is a different language from British Sign Language. There are also other forms of sign language used in the UK. Many children with severe hearing loss are helped to learn English by using another sign system, Sign Supported English, which does have the same syntax and grammar as English.

If a child has a significant hearing loss, particularly from a pre-lingual stage, their language and cognitive development can be adversely affected. They will benefit from having materials available in visual instead of audible form, for example, as sub-titles, alternative texts for spoken output, and descriptors for sounds.

Communication, language or learning difficulties

There is a significant group of people with communication, language or learning difficulties that require simplification or re-purposing (see Appendix 4) of learning resources and web content. They may also require other forms of differentiation in the user interface. For example, some might find the language structure, its form or representation,

and the vocabulary level of text too complex. If so, simplified or re-purposed language, easy access, and alternative user interfacing (for example, web pages embellished with key symbols, pictures or icons) within the basic functionality of an application's or web site's environment might be called for. Such users may also require that the barriers of elaborate menus, options and dialogue boxes be removed. Supporting navigation within an application or web site through tailored systems incorporating multi-modal feedback through sound, voice and graphics may also require enabling technologies.

Benefits to end users and producers

"The power of the web is in its universality. Access by everyone regardless of disability is an essential aspect." Tim Berners-Lee

"Over the next 10 to 15 years, technology has the capacity to virtually eliminate barriers (faced by people with disabilities..." Steve Ballmer, CEO Microsoft Corporation[2]

What are the benefits to the user?

ICT can give students independent access to information and learning environments. This allows them to work in privacy, without all information having to be filtered by scribes in the form of helpers and assistants. Consideration of learning methods and work patterns makes it clear just how important this new freedom is for people with disabilities. Such independence via accessible Information Technologies can give access to users for work, play and education through:

- Reading: providing many disabled people with an extraordinary opportunity to read independently and to 'turn' pages, often for the first time.
- Information retrieval: accessing electronic encyclopedias, dictionaries, databases and websites on all kinds of subjects such as natural science, history, cinema, the arts, photography and huge photo and video libraries.
- Recording and Writing
- Music, film and sound archives.
- Recreation and entertainment.

What are the benefits to publishers and developers?

In adopting a "design and access for all" ethos you:

- Open a door to an important minority for whom this may be the only way to access the information.
- Improve design and user interfacing for non-disabled people.
- Ensure equal opportunities for people with a disability using ICT.

- Gain access to a bigger market for very little effort.
- Invite good publicity.

and thereby create a better product !

This will also enable you to support the revised Code of Practice, implemented in January 2002, which provides a framework for developing a consistent approach to meeting children's Special Educational Needs (SEN) and places the rights of children at the heart of the process. (**www.teachernet.gov.uk/teachinginengland/detail.cfm?id=377**)

Compliance with the Disability Discrimination Act (DDA)

It is now a legal duty for those supplying the education sector to make their products accessible. It has been legal requirement for web sites to be accessible since 1999 for example. Education is subject to DDA following the Special Educational Needs and Disability Act (SENDA) 2001. Educational institutions c purchasing educational software need to ensure that disabled students are not put at a disadvantage by the purchase of inaccessible software for delivery within the curriculum. Institutions are required to make 'reasonable adjustments' to ensure that a disabled student is not placed at a 'substantial disadvantage'. These requirements will obviously influence their choice of software, educational material and media policies. Also, if you follow the guidelines in this document you are unlikely to be sued!

New markets

Multimedia technology has had a dramatic impact on people with special needs and opens new and significant markets for 'inclusive' products and services to those who seize the opportunity.

Provision of a much-needed service

In drawing up a modest list of suggestions (see below) we have borne in mind the impact on costs and resources and feel that the burden of any extra work will be more than compensated for by the improved image and prestige gained by the publisher or developer.

Spin-offs to benefit able-bodied people

It has been clear for many years that software and user interfaces designed for people with special needs are often just as useful to non-disabled users of ICT. Designing for 'All' is often simply a matter of good design as advocated in a large body of human-computer interface literature.

In many cases we can see that consideration of the user interface in terms of people with disabilities has made for a better 'mainstream' product. For example, the provision of simple keyboard shortcuts for 'Play' (say "P") or 'Movie' (say "M") buttons for sound or video clip resources, will be used by many non-disabled users. In many circumstances it is more convenient and faster than the alternative of steering the mouse pointer to the relevant icon.

What are the latest accessibility and usability challenges within current ICT technology?

Technology is changing at an ever-increasing rate. At any time we may be confronted by new challenges caused by the developing technology.

Interactive whiteboards

Interactive whiteboards have their own inherent problems that complicate the way in which resources can be accessed. Children in wheelchairs (or those who are just of short stature or have foreshortened limbs) find interacting with these boards difficult, even if they are mounted low down on the wall.

Interactive whiteboards that use active pens as opposed to touch sensitive surfaces can be particularly difficult for many people with disabilities. They may be able to touch an area of the board (if they can reach it) but holding a pen can be an additional handicap. At the very least, all interactive controls/functions should be displayed at the bottom of the screen.

Many users will only be able to interact with whiteboard displays through their own computers or communication aids. Although this is perfectly feasible, the setting up of communications between whiteboards and an individual's computer is too technically complex and non-standard to make it generally practicable in schools, colleges, or universities.

System and hardware changes

All computer systems are constantly being upgraded and improved. AT producers do not have the resources of companies like Microsoft, Apple or Intel and thus find it difficult to keep pace with changes. For example communication aid users sometimes use the serial (comms) port of computers along with serial keys software to control their computers. The low speed serial port will soon not be supported by systems writers and not available on computers at all!

Development environments

Flash

In the past Flash has been notoriously inaccessible. Macromedia (producers of Flash MX 2004) have been making efforts to improve the accessibility of their tools but this new message has not necessarily reached all those using Flash as a tool to develop Web resources. Macromedia have published excellent guidelines for accessibility within Flash products, for example, "Best Practices for Accessible Flash Design" by Bob Regan – 2004. (**www.macromedia.com/resources/accessibility**).

Shockwave (Macromedia Director)

Shockwave is another multimedia environment that has caused problems with accessibility. Much of the problem now stems from the media creators who use Director to create inaccessible resources. Macromedia have created tools to permit Shockwave to be accessible, but these have not been extensively implemented in the past. (See **www.macromedia.com/macromedia/accessibility/features/director**)

Java

The TRACE Center at the University of Madison, Wisconsin started an extensive project on JAVA accessibility in 1996, in cooperation with SUN microsystems. A report on the project can be found at the following links. Sun has gone a long way in implementing tools for accessibility within the Java development environment. Along with accessibility tools for the developer Sun have made available diagnostic utilities which will point out poor accessibility in Java applets, forms etc.

See **trace.wisc.edu/docs/java_access_rpt/report.htm**
and **www.sun.com/access**
and **www-306.ibm.com/able/guidelines/java/accessjava.html**

'Electronic' books

Electronic books are becoming increasingly popular as a way of delivering text. The electronic format that the text is made available in may make it difficult for some people to access the text. This is particularly so if books contain images. The challenges are similar to those posed by Web sites.

The DAISY standard is worth considering if you are publishing materials that include a large amount of text. The system makes such text available to people with reading problems in an easily navigable form.

A DAISY book can be explained as a set of digital files that includes:

- Digital audio files containing a human narration of part or all of the source text;
- A marked-up file containing some or all of the text (strictly speaking, this marked-up text file is optional);
- A synchronization file to relate markings in the text file with time points in the audio file; and
- A navigation control file, which enables the user to move smoothly between files while synchronization between text and audio is maintained.

The DAISY/NISO standard, formally defines a comprehensive system for Digital Talking Books. A part of this standard is DTBook, an XML vocabulary that provides a core set of elements that are needed to produce most types of books.

The DAISY/NISO standard has been extended so that it can address additional needs for example, mathematics, video support, testing, workbooks, music, dictionaries, and chemistry. Guidelines have been published to make it possible to add modular extensions that are interoperable. See **www.daisy.org/accessibility**

Portable devices

Information from the Web and other sources will in the future be increasingly available on smaller and more portable devices such as PDAs, mobile telephones and MP3/iPod type devices. As these devices shrink in size alternative access becomes more problematic. In general, materials should not be exclusively designed for such devices but should be made available in more standard form as well.

One obvious benefit of mobile phones has been that given to hearing impaired people being able to send and receive SMS (text) messages.

The Web and Web-based content

Much work has been carried out on creating guidelines for web site developers, not all of which has been heeded. The challenge is to allow users with poor language skills to locate navigate and access content more effectively. The Semantic Web offers some hope for the future for providing more effective ways of presenting or defining content and allowing re-purposing of content.

> *"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."* Tim Berners-Lee, James Hendler, Ora Lassila, The Semantic Web, Scientific American, May 2001

Semantic Web technologies are still very much in their infancies, and although the future of the project in general appears to be bright, there seems to be little consensus about the likely direction and characteristics of the early Semantic Web. Data that is generally hidden away in HTML files is often useful in some contexts, but not in others. The problem is that the majority of data on the Web is in this form (HTML). There is no global system for publishing data in such a way as it can be easily processed or re-purposed,, it is difficult to use this data in the ways that one might want to. The Semantic Web should make it easier to publish data in a re-purposable form, so users with sensory or learning disabilities, for example, may be able to access this information.

The Semantic Web is based on RDF (Resource Description Framework) and URIs (Uniform Resource Identifiers). A URI is simply a Web identifier: like the strings starting with "http:" or "ftp:" that you often find on the World Wide Web. URIs form an ideal base technology with which to build a global information system". RDF XML is considered to be the standard interchange format for RDF on the Semantic Web. Once information is in RDF form, it becomes easy to process. For example, Concept Coding (see Appendix 4) is a method that can process this RDF information and re-purpose it for people who use symbolic rather than orthographic methods of reading.

For more information on the Semantic Web go to **www.semanticweb.org** or for information on Web Universal design see **we04.com/resources/accessibility/index.htm**

Games

The ACE Centre is currently running a project looking at the accessibility of 'computer games' - GameOn. One of the main difficulties with PC software is the use of DirectX. This by-passes many of the accessibility systems and in many cases grabs the entire screen, leaving no room for on-screen keyboards etc.. Many people in or associated with the games industry (IGDA - International Games Development Association) have formed a special interest group promoting accessibility. They are producing guidelines for the industry. For more information go to **www.igda.org/wiki/index.php/Game_Accessibility_SIG** They have come up with a wish list for accessibility features for games as a 'top ten':

Brief Top 10:

1 Provide volume controls.
2 User definable controls.
3 Broad difficulty level options.
4 Subtitles for dialogue and sound effects.
5 Trainer/help modes.
6 Easy to navigate menus.
7 Text to Speech.
8 Speed control.
9 Thought for partially sighted and colour blind gamers.
10 Simplified control mode option.

More detailed Top 10:

1 Volume controls. Should enable individual volume controls over music and sound effects from off to full volume. These should take effect over the entire game, menus included. These can benefit deaf and learning disabled gamers.

2 User definable controls. Ideally featuring the ability to map multiple controls to individual buttons, rapid fire (if appropriate) and ability to use digital controls to operate analogue controls.

3 Broad difficulty level options. Perhaps ranging from 1-10: 1 being the easiest; 10 the hardest. Level 1 should seem be extremely easy for the majority of gamers. Games that have lives or time limits could offer a wide range of choices here too.

4 Subtitles for dialogue and sound effects.

5 Trainer/help modes. Perhaps an option you can turn on that 'holds your hand' through a game, and points out various features to make them easier to understand.

6 Easy to navigate menus. Quick start for a game, minimal number of controls necessary to navigate.

7 More spoken text. Option for all text to be read out loud.

8 Speed control over game. Variations from very slow to normal.

9 Thought for partially sighted gamers (e.g. adjustable text size / colour scheme etc).

10 Simplified control modes. Simplified controls for gamers unable to use many controls.

What can mainstream software & Web-based developers do to help – the key question?

"There is no simple solution, no one size fits all. But designing for flexibility helps..... Fixed solutions will invariably fail with some people; flexible solutions at least offer a chance for those with special needs" Norman[17]

General accessibility

Usability in software interfacing must mean much more than straightforward accessibility. The mantra is "Efficiency, not sufficiency". In other words the end user with a disability should never have to put a lot of effort in, in order to get a little out of an application or website. The acid test is indeed: "If I put a little in, will I get a lot out?" If the answer is "Yes", then the application or website will be "efficient" and not merely "sufficient". To achieve this, there are a number of general features for developers to consider that will guarantee improved design and access for all.

Principles of accessible software

These basic principles should be followed when developing software:

Choice of input methods

Support the user's choice of alternative input methods including keyboard, mouse, voice, and assistive devices emulating keyboards via the serial (serial keys) and other ports. The primary requirement is to provide keyboard access (mouseless operation) to all features and functions of the software's application wherever possible. The operating system usually provides support for input via the serial port, keyboard movement of the mouse pointer, and other keyboard enhancements.

Choice of output methods

Support the user's choice of output methods including display, sound, and print. The primary requirement is to provide text labels for icons, graphics, and user interface elements and to support visual indications for sounds. Implementing the accessibility APIs (e.g., Java Accessibility, Microsoft Active Accessibility, etc.) for the target platform will meet this principle.

Consistency and flexibility

Make the application consistent with the user's choice of system behavior, colours, fonts sizes, and keyboard settings. Provide a user interface that can be customized to meet needs and individual preferences. Make actions and controls as consistent as possible to aid automation and ease of use.

Web-based resources

Many resources are now in Web compatible formats such as HTML, XML and can include Flash, Shockwave and Java plug-ins. In such cases the version 1 guidelines produced by the WAI initiative of the W3C (**www.w3.org/WAI**) should be followed and perhaps exceeded for switch users. The second published edition of the web content guidelines is expected in 2005/6 although it is available in draft form already (**www.w3.org/TR/WCAG20**). However, there are a number of other web sites in addition to WAI which provide help for designers, for example the British Universities' TechDis Accessibility and Usability Resource, (**www.techdis.ac.uk**) and the Dublin Core Metadata Iniatiative (DCMI - **dublincore.org**).

Information Display and Screen Layout

Screen displays should be as clear and uncluttered as possible: information should be presented in logical and consistent positions. Similar information should be grouped and visually coded (e.g. colour coded) where appropriate. Similar information and controls should be consistently positioned and presented. Navigation design, the size and position of buttons or links, consistency of position of links, the use of 'breadcrumb' or 'paper' trails can all aid the user in getting to the place they want to be. For example, see **www.webdesignpractices.com/navigation/breadcrumb.html**. This can be in the form of the 'back' action in web browsers or as a 'navigate up the logical tree' from the present position. Either way, the user should have a visual clue as to where they are, and a simple mechanism to get to somewhere else. Efficiently and simplicity are important, particularly to those who have to spend a great deal of effort getting to where they are!

Navigation Design

There are at least 4 ways of navigating a GUI that are perfectly standard in Windows and partially implemented in Mac OS. These are:

- Pointer (mouse point and click – do not replace the system pointer with your own, the user may require a different format)
- Direct keyboard short cut (e.g. control + P for 'Print' or Alt + F4 for 'Close Applicationt')
- Indirect short cuts via menus (Alt + F, followed by A for 'Save As')
- Scanning keyboard control (e.g. using the Tab key to move around a dialogue, Alt + arrow keys to scan menus). Don't forget to make the 'active control' highlight clear, the Windows standard is not very visible.

If all these protocols are implemented then users with different needs and control methods will be able to access an application or resource efficiently.

Pointer users now include whiteboard and touch screen in their arsenal of equipment. Consistent positioning of buttons is important. Ideally they should be configurable as to where they are on the screen, but if only one position is possible then the bottom of the screen is best. The buttons or 'hot spots' should be sufficiently large for those with poorer coordination to find it easy to activate. We would recommend a minimum size of 1/12 of the screen height i.e. 50 pixels on a 800 x 600 screen. All buttons should have direct and if possible scanning keyboard shortcuts.

Appearance

Colours and fonts should be carefully chosen in terms of visibility and consistency. There should be good colour contrast and particular care should be taken to avoid the juxtaposition of elements that are tonally very close to each other. Set a sensible default and the let the user change things as they need. Colour coding of different types of information pages - for example, menu pages in one colour, information pages in another, is an example of good practice and will also improve navigation through the application or web site. Users often use the standard personalisation settings for the display in Windows control panel to set up more readable screen for themselves. It would be preferable if other applications used those colours and fonts in their interfaces.

Windows XP now has a built in magnifier and, although this should not be disabled by an application, it is not a panacea for users with visual problems.

Data access and indexing

Wherever possible, applications should be designed so that data and indexing are free from the constraints of the interface. In this way it should always be possible for the end user to simply and easily replace a graphical interface with a pure text. This principle offers the useful spin-off when it comes to cross-platform development, permitting the data to remain unchanged as new GUIs are developed.

Saving text

Applications and web sites should allow text to be saved on request by a user so that it can be read later via the end user's specialist utilities. For example, navigational assistance should be provided for the standard user interface to enable the user to make primary searches, and then save records or sections to disk for later investigation.

Time dependency

Reduce or abandon the need for a speedy response wherever possible. Most people who

Claire

Claire is a young woman with athetoid cerebral palsy now in a mainstream secondary school.. Over the years she has used a variety of access methods starting with a single switch to the specialised joystick she uses now. She uses a computer as both a writing aid and a way of making herself better understood through voice (text-to-speech) output. By using an on-screen keyboard and word prediction she has recently gained an A* mark for GCSE maths – a year early.

have a disability will be slow at entering keystrokes or moving to a mouse button. Obviously 'shoot em up' games are not going to be accessible to people with serious physical or visual difficulties. However, if programs are 'impatient', (i.e. expecting a response within a certain time frame and moving on to other prompts or options automatically), it can cause serious problems.

Application customisation

A degree of personalisation should be available within all software in order to match the individual needs of the user. Someone with a visual problem may wish to change the font size, typeface and colour contrast in order to be able to read information from a page. Alternatively, simplified menus with fewer options may be required for someone with a cognitive problem. Simplified menus have been taken up in Microsoft Word 2000 and Word itself has been available in simplified form as 'Talking First Word' from Research Machines for some time.

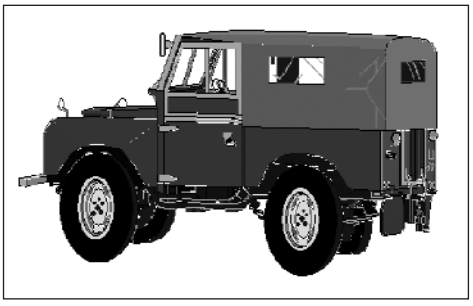
Physical disability access

Specific examples are given below to considerations for users with specific computer-interface problems:

Users who can only use a keyboard

There is a range of alternative input devices regularly used by people with physical difficulties. Some people using alternative inputs are very slow and/or weak and tire easily. It is important that:

- all functions can be accessed through the keyboard either directly or indirectly via keyboard-controlled menus and dialogues.
- where mouse control is absolutely essential to the functioning of an application/web site, then at least the mouse buttons or "hotspots" should be large and highly visible. We would suggest a minimum size of 48 pixels square on an 800 x 600 resolution screen.



A picture drawn by keyboard with a program that has been written with keyboard users in mind (AccessMaths)

Users who benefit from modified pointing devices

There are some disabled users, who could use pointing devices such as mice, trackerballs, or modified joysticks, if the way in which the computer behaves were made more appropriate to their needs. Technology in this area has made great strides in the past few years. Head pointing and eye gaze devices are now much improved and the use of eye gaze in particular has become a practical proposition for disabled users as exemplified in the Cogain project (http://www.cogain.org/). People using pointing devices often need to access Web based/format material by using specialised browsers such as the one developed by the WWAAC consortium (http://www.wwaac.org/).

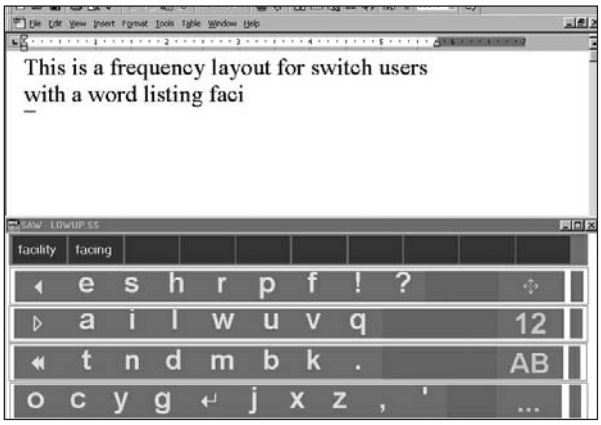
Existing good practice in the form of available software utilities already allows:

- the speed of the mouse to be slowed or variably accelerated (Microsoft's Accessibility Options and Intellimouse)
- an increase in the time between double clicks (various Windows mouse drivers)
- the mouse pointer to be enlarged or assigned a different colour and even animated.
- screen magnification that enlarges specific areas, for example, desktop icons, making it easier to target the pointer accurately.
- Web format resources should be accessible through alternative Web browsers.

N.B. It is essential that these fundamental accessibility functions are not "squeezed out" or overwritten by the application itself.

Users who cannot use keyboards or pointing devices

There are a number of user groups who cannot access any kind of external keyboard or pointing device. They include switch users like Professor Stephen Hawking who uses a specialist switch utility that gives access to an on-screen keyboard and mouse emulator or modifier. Switch utilities like Discover Switch, The Grid, SAW, WiViK and Clicker allow grids of selectable information to be presented on the screen and accessed via as few as one or two external switches. The elements of the 'grids' or 'selection sets' can be highlighted and items chosen with a switch press. The switch utility then feeds the selected information to the application or web site as though it had come directly from the keyboard. The user or helper can design the content of the grids and its keyboard-equivalent information (including macro strings). This means that a switch user can access any program that can be controlled from the keyboard. An interactive demonstration of switch scanning methods can be found at: **www.switchscanning.org.uk/new2.htm**



For access by means of a switch utility, keystroke equivalents for all mouse actions, buttons, 'hotspot' areas, menus, and functions should be provided within the software application or web pages. This allows scanning software to easily interface with the software to provide access directly to the required controls.

At the moment keyboard equivalents may only be practicable within certain operating systems. For example, many Microsoft Windows applications are fully or nearly fully keyboard accessible from the keyboard and accessible even for single finger.

The Macintosh system can be modified to make it more accessible. Some development systems, those mainly used in web design such as 'Flash' and 'Shockwave' have historically had

very poor accessibility for keyboard users. A simple procedure to see if access is achievable is to test the application with the mouse unplugged!

For users who can use only a mouse, mouse equivalents for keystroke commands should also be provided, either directly on the screen as buttons or at least by means of the on-screen keyboard (e.g. The Grid, SmartType, SAW etc.) that they normally use. Can the application be accessed without the keyboard, only using a mouse and mouse driven on-screen keyboards?

Use of the manufacturers' recommended standard system calls to the keyboard will usually enable alternative access to keyboard functions. Where this is not possible, software designers should ensure that buttons or 'hotspot' areas that are repeatedly used on different screens are placed, wherever possible, at consistent points on the screen, for example, always along the bottom of the screen (useful for easy access by children using a classroom white board anyway). This will ensure that the switch utility can enable pointing to be achieved by means of the same x, y pixel co-ordinates.

As we have said before, switch emulation of the mouse within switch utilities is of course possible but is very slow for the user, illustrating the difference between "sufficiency and efficiency" and "accessibility" and "usability".

Finally, it can save time and effort for users if 'hotspots' or mouse buttons are visible at all times rather than just during 'mouse over' or mouse click events.

Visual Impairment

Visual impairment does not mean blind. Many users have a wide range of visual problems which increase with age. Graphical user interfaces are problematic both for navigating and for information retrieval. It is important that interfaces are clean, uncluttered and consistently laid out with keyboard alternatives for 'point and click' controls. Wherever possible the user should be able to define the font, size and colours of any text presented. Pictures, symbols and icons should have alternative text descriptions. Information in text format may need to be presented in alternative formats and should also be available for conversion via text-to-speech. Retrieval of information can be carried out directly from the CD-ROM, DVD or web page by accessing files via the user's own specially modified system. In this case any accessible files should be named descriptively. This means that the main emphasis can be placed on achieving access to navigation provided that, at the point of information retrieval, the relevant data can be downloaded. Text data should for example take the form of unformatted text ready for conversion or re-formatting.

Catherine

Catherine is blind. She is 13 and is included for most of her timetable into the mainstream classes of a large mainstream secondary school. She can find an article relevant to a particular topic on a CD-ROM based encyclopedia or by surfing the Internet. But to keep up with her peer group she has to be able to retrieve and digest information quickly and independently. Again this can only be achieved if the multimedia material she is searching is genuinely accessible.

There are different considerations regarding access to mixtures of text and graphics and access to text only material. Access to graphical material depends on the visual acuity of the

user. For example, a graphical interface is unlikely to be accessible to blind users. Considerations that will facilitate access for partially sighted users include: The RNIB point to the US ADA Section 508 guidelines at **www.section508.gov** and the Trace center guidelines at **trace.wisc.edu/docs/software_guidelines/software.htm**. The latter is now quite old and general but contains many points that are still worth considering. Key points are:

- Icons should be very clear and readily distinguishable from one another.
- Key elements should be reinforced through speech, such as menu box items having a keyword spoken when a pointer is moved over them. These can be 'screen tips' made available to utilities that use speech synthesis.
- Text captions should be associated with pictures (similar to HTML alternate text), so that users, who may not fully appreciate the detail of an illustration, may still be aware of its primary content. This text can then be spoken via speech synthesis.
- Follow best practice guidelines about font size and type, screen and text colours, and contrasts making these configurable if possible.
- Be consistent in the positioning of instructions, options and hotspots.
- Use auditory support (digitised and synthesised speech) throughout the interface as well as for pop-up windows.
- Avoid drag and drop

Many partially sighted users can read text from a screen if:

- Font sizes are not too small and are in a clear font style, preferably set by the user.
- The font letter and background colours are configurable.
- Font typeface and sizes for body text are adjustable so that the user can navigate by recognising word shapes but will need greater access to read detail.
- Text "read out" is made possible everywhere. This can be done either directly through digitised speech or by using a software text reader utility and software text-to-speech. See the guidance under "Screen reading access" (see below).
- Screen text needs to be well spaced - people read from screens about half as well as from books.

Screen reading access can be considerably improved if:

- Windows overlap as little as possible and multiple windows are only used sparingly. This overlapping affects both the clarity for the partially sighted person and also the extent to which screen-reading utilities can be tailored to the particular screen display or web page.
- Window positions are as constant as possible. Where windows do overlap, their position should be as consistent as is practicable so that the screen-reading software knows where to "pick up" the window position.

- There are not too many variables in the layout. A small number of standard layouts make life much easier for screen readers.
- There is an active on-screen cursor (carat). This need not be visible, but should not simply be a graphics image. If included, it will mean that the text can be highlighted and read one line or one section at a time.
- The cursor icon and pointers are clearly visible and not too small.
- General sound reinforcement can give valuable spatial cues.
- Don't just allow larger text in the same fancy font. It's rather like shouting to make yourself understood to an non-English speaker. Simpler or shorter words may be needed to make reading easier.

Hearing Impairment

Many resources now contain extensive audio resources. Those with hearing impairments will need some method of accessing this information. Closed captioning of commentaries and descriptions of sounds being played are required for full accessibility. At least a text equivalent or description of any commentary should be available. Windows Accessibility does include a 'Sound Sentry' which indicates the name of any sound file being played; it would be helpful if such files were named 'dog barking.wav' for example, rather than 'ax1035b.wav'! Because of the restrictions of DOS and earlier versions of Windows a habit of giving files short, cryptic names has developed. Good practice would be to give *all* resource files meaningful names.

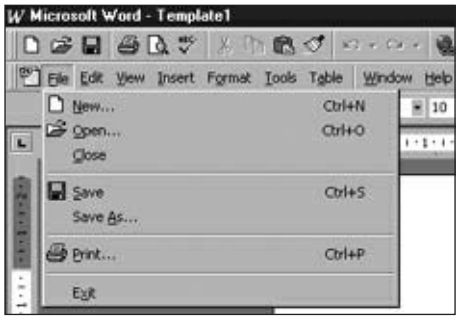
If people are shown speaking on video clips try to ensure that the lips are clear so that lip-readers are helped. Background noises such as the music found in some games can be confusing and mask the spoken or essential sounds, at least make such 'lift music' optional. Again simpler language will be easier for those whose language development has been held back by hearing loss.

Extra help for those with communication, language or learning difficulties

Simplification or the use of "plain" language is an issue that is relevant to *all* users. But it has significant impact on those with communication, language and learning difficulties; those in particular for whom a country's first language is not the end user's first language; and those who have to commit instructions to auditory memory. Indeed, development methods, which use alternative language versions of all commands held centrally, not only aid simplification but also enable other language versions to be created easily and quickly.

It is recommended that you:

- Provide the 'hooks' in the software design to highlight or enlarge/magnify the text as it is read, and/or send it to 'SAPI' (Speech Application Program Interface – see Appendix 4)software speech or an external speech synthesiser. Alternatively, it can be sent to another application that carries out such specialist functions. If text is enlarged or magnified as it is read, this will also help people with a visual impairment.



Simplified File menu in MSWord

- Provide text equivalents to digitised sounds
- Provide alternative, easier text by way of a summary with a limited vocabulary and simpler sentence structure. This could be used by other utilities to translate text into iconic/symbolic representations and is particularly important in the context of web pages.
- Use short sentences avoiding embedded clauses wherever possible!
- Avoid the use of jargon when it is not relevant to the task.
- Use "pop-up" or "screen tip" texts (with auditory support if required) to support icons within the layout.
- Enable the substitution of appropriate icons or pictograms to replace or complement keywords (for example, within menus).

Don't forget in involve users with disabilities in your prototyping and testing

Some users need the option of 'keyword' AAC (augmentative and alternative communication symbols) and other graphic representations to support orthographic instructions. There are a range of AAC symbol systems in use (for example, Picture Communication Symbols, Rebus, Bliss & PICTO). Providing simplified texts will be key to implementing this as an external utility for content re-purposing and symbol-embellished annotations. The Concept Coding Framework (CCF) is a project aiming to provide a linguistic link between these symbol systems and orthographic language (**www.conceptcoding.org**). This bridging approach implies that the CCF is not intended to replace and exclude the use and development of current and future proprietary assistive technologies in the AAC field. Instead the CCF and its bridging technology and tools are designed to provide added value general service and information providers who want to support improved accessibility to their products.

A top 10 accessibility guide

1 All controls should be accessible from the keyboard and should not override the operating system's Accessibility Options.

2 Create consistent page layouts with buttons and controls in fixed positions wherever possible.

3 Enlarge the size of your 'hotspots' to at least 50 pixels square.

4 Make text style, size and colour configurable (allow style switching).

5 Create alternative texts for pictures and where picture maps are used, provide alternative text menus as well.

6 Make all textual material available as text that can be copied.

7 Remove time dependency for controls.

8 Create alternative simplified texts, or at least summaries.

9 Create alternative texts for sounds or at least give sound files descriptive names.

10 Utilise system accessibility tools wherever possible.

*Inclusivivity * rating*

What you should do to make you software inclusive. In line with the 'Bobby' ratings for web sites we would propose an 'inclusivity' rating for accessibility.

***** The program is directly and efficiently accessible by all users

**** The program is efficiently accessible for most users - either directly or by using accessing tools

*** Alternative access devices or software can access most aspects of the program

** The program is easy to use but requires mouse and keyboard to be used

* The program has poor accessibility for everyone

Summary

To summarise, the key points for mainstream software and web page designers interested in meeting reasonable standards of accessibility and usability are, as follows:

Interface

- Do not remove or disable system accessibility tools
- Do not disable individual user preference software for providing accessibility support
- Reduce effort and speed up access.
- Ensure there is keyboard access for all controls with keyboard shortcuts that give absolute results.
- Keep buttons in the same location and, if not customisable, at the bottom of the screen.
- Utilise system accessibility tools wherever possible.

Text

- Allow users to customise fonts and colours
- Reduce language complexity.
- Make all textual material available as text that can be copied.

Sounds

- Label sound resources.
- Provide visual indication when playing sounds
- Have text alternatives for sounds.

Graphics

- Enlarge the size of your 'hotspots' to at least 50 pixels square (on an 800x 600 screen resolution) and maintain consistency in positioning.
- Make sure "picture maps" have alternative text descriptions.
- Make available to alternative access designers the bitmaps and other graphics used throughout the application/web site. Ideally this should be as a single, coherent resource.

Or to put it another way:

Make ALL your materials accessible and navigable to EVERYONE. But focus on usability as well as accessibility, in that way you will open up the world of information to everyone including those with disabilities.

Appendix 1:
Contact centres for information about accessibility

AbilityNet
IBM United Kingdom Ltd
Birmingham Road
Warwick
Warwickshire CV34 5J
Tel 0800 269545
Fax 01926 312847
enquiries@abilitynet.co.uk
www.abilitynet.co.uk

Access Committee of England
12 City Forum
250 City Road
London EC1V 8AF
Tel: 0207 250 0008 Fax: 020 7 250 0212

Access to Communication & Technology
91 Oak Tree Lane
Selly Oak
Birmingham B29 6JA
Tel: 0121 627 8235
Fax: 0121 627 8210
act@southbirminghampct.nhs.uk
www.wmrc.nhs.uk/act

ACE Centre
Aiding Communication in Education
92 Windmill Road
Oxford OX3 7DR
Oxfordshire
Tel: 01865 759800 Fax: 01865 759810
info@ace-centre.org.uk
www.ace-centre.org.uk

ACE Centre North
Units 11 & 12
Gatehead Business Park
Delph
Saddleworth
OL3 5DE
Tel 01457 829444
Fax 01457 829441
enquiries@ace-north.org.uk
www.ace-north.org.uk

The Advisory Unit: Computers in Education
126 Great North Road
Hatfield AL9 5JZ
Hertfordshire
Tel: 01707 266714 Fax: 01707 273684
sales@advisory-unit.org.uk
www.advisory-unit.org.uk

British Dyslexia Association
98 London Road
Reading RG1 5AU
Bershire
Tel: 0118 966 2677/ Helpline 8271
Fax: 0118 935 1927
www.bda-dyslexia.org.uk

BECTa
Milburn Hill Road
Science Park
Coventry CV4 7JJ
UK
Tel 024 7641 6994
Fax 024 7641 1418
e-mail becta@becta.org.uk
Web Site www.becta.org.uk

British Computer Society Disability Group
1 Sanford Street
Swindon
Wiltshire
SN1 1HJ
Tel 01793 417417
mbauer@hq.bcs.org.uk
www.disability.bcs.org.uk

CALL Centre
University of Edinburgh
Paterson's Land
Holyrood Road
Edinburgh EH8 8AQ
Scotland
Tel 0131 651 6235
Fax 0131 651 6234
callcentre@ed.ac.uk
www.callcentrescotland.org.uk

CAST
40 Harvard Mills Square
Suite 3
Wakefield,MA 01880-3233
Tel (781) 245-2212
cast@cast.org
www.cast.org

CENMAC
Charlton Park School
Charlton Park Road
London SE7 8JB
Tel 020 8854 1019
Fax 020 8854 1143
cenmac@lgfl.net
www.cenmac.com

Central Remedial Clinic
Vernon Avenue
Clontarf
Dublin 3
Tel: +353 1 8332206 Fax: +353 1 83354576
www.iol.ie

Chailey Heritage Rehab. Eng. Unit
Chailey Heritage
North Chailey
Near Lewes BN8 4EF
East Sussex
Tel: 0182572 2112 ext 210

Compass
Royal Hospital for Neuro-disability
West Hill
Putney
London SW15 3SW
Tel 0208 780 4500 ext 5237
gderwent@whn.org.uk
www.rhn.org.uk/cat.asp?catid=22

Computer Centre for People with Disabilities
University of Westminster
72 Great Portland Street
London W1N 5AL
Tel: 020 7915 5457 Fax: 020 7911 5162
www.wmin.ac.uk/ccpd

Dart
Kruthusgatan 17
Göteborg 411 04
Sweden
Tel 031- 739 80 80
Fax 031- 739 80 90
e-mail dart.su@vgregion.se
Web Site www.dart-gbg.org

Equal = ability

170 Benton Hill
Wakefield Road
Horbury
West Yorkshire WF4 5HW
Tel 01924 270335 Text: 01924 232436
Fax 01924 232435
mail@equalability.com
www.equalability.com/

Hereward College Access Centre

Bramston Crescent
Tile Hill
Coventry CV4 9SW
Tel: 024 76 461231

ISAAC

International Society for Alternative and Augmentative Communication
49 The Donway West
Suite 308
Toronto M3C 3M9
Ontario
Canada
Tel: +1 416 385 0351 Fax: +1 416 385 0352
isaac_mail@mail.cepp.org
www.ISAAC-online.org

Mary Malborough Centre

Windmill Road
Headington
Oxford OX3 7LD
Tel: 01865 227600 Fax: 01865 227463

NASEN

National Association for Special Educational Needs
4 Amber Business Village
Amber Close
Tamworth B77 4RP
Tel 01827 311500
Fax 01827 313005
e-mail welcome@nasen.org.uk
www.nasen.org.uk

National Autistic Society

393 City Road
London NW2 5RB
Tel 020 7833 2299
Fax 020 7833 9666
nas@nas.org.uk
www.nas.org.uk/

Nielsen Norman Group

48105 Warm Springs Boulevard
Fremont
California 94539-7498
Tel +1 (415) 682-0688
info@nngroup.com
www.nngroup.com/reports/accessibility/

RNIB Education Unit

Jane Lambert
The Sale Service
Garrow House
190 Kemsal Road
London W10 5BT
Tel: 020 7 388 1266 Fax: 020 7 383 7613

RNIB National Education Service

Garron House
190 Kensel Rd
London W10 5BJ
Tel: 020 8 968 8600 Fax: 020 8 960 3593
djohnson@rnib.org.uk

SIH

Specialpedagogiska institutet Läromedel Datapedagogen
Box 1100, 871 29 Härnösand
Tel. +46(0) 0611 887 70
Fax. +46(0) 0611 268 66
sit@sit.se
www.sit.se/net/

SKILL

Chapter House
18-20 Crucifix Lane
London SE1 3JW
Tel 020 74500620
Fax 020 74500650
skill@skill.org.uk
www.skill.org.uk

Swedish Handicap Institute

Hjälpmedelsinstitutet, Sorterargatan 23
Box 510
S-162 15 Vällingby
Tel: +46 8 620 17 00 Fax: +46 8 739 21 52
www.hi.se/English/default.shtm

Special Needs Computing Research Unit

University of Teesside
School of Computing
Middlesbrough TS1 3BA
Tel 01642 34 2656
e.pearson@tees.ac.uk
rime.tees.ac.uk/sncru/welcome.htm

Special Needs Research Centre

School of Education, Communication and Language Sciences
Faculty of Humanities and Social Sciences
University of Newcastle upon Tyne
Newcastle-upon-Tyne
Tel +44 (191) 2225672
snrc@newcastle.ac.uk
www.ncl.ac.uk/ecls/research/education/snrc

The Foundation For People With Learning Disabilities

Conference Unit
20-21 Cornwall Terrace
London NW1 4QL
Tel: 020 7535 7461 Fax: 020 7535 7474
tdyson@mhf.org.uk

TRACE Center

Trace Research & Development Center
University of Wisconsin-Madison
2107 Engineering Centers Bldg.
1550 Engineering Dr.
Madison WI 53706
Winsconsin
Tel: +1 608 262 6966 Fax: : +1 608 262 6848
info@trace.wisc.edu
trace.wisc.edu

VICTAR (was R.C.E.V.H.)

School of Education
University of Birmingham
Birmingham B15 2TT
Tel 0121 414 6733
victar-enquiries@bham.ac.uk
www.education.bham.ac.uk/research/victar/

Web Accessibility Initiative (WAI)

W3C Web Accessibility Initiative
MIT/CSAIL, Building 32-G530
32 Vassar St
Cambridge, MA 02139
USA
+1 617 253 2613
jbrewer@w3.org with "cc" to
w3t-pr@w3.org
www.w3.org/WAI

Appendix 2:
Some software and hardware accessibility tools

Product	For	Type	Description
Accessibility Options *Microsoft* www.microsoft.com/enable	Win	Utility	Built in to Microsoft Windows from Windows 95 on. They permit changes to the keyboard response, altering the key auto-repeat rate and speed at which the system responds as well as zoom facilities and control over the visibility of the interface.
SAW *The ACE Centre* www.ace-centre.org.uk	Win	On-screen Keyboard	Switch Access to Windows. Giving switch and mouse pointer users full keyboard access and mouse control. The program includes a full graphic-based designer for creating new keyboards. . It can be accessed by switch users and pointer users.
Discover Switch *Madentec* www.madentec.com	Win Mac	On-screen Keyboard	Originally Ke:Nx for the Mac OS. A number of keyboards are supplied along with template keyboards that can be modified to individual requirements. It monitors the windows that are open and can automatically load the appropriate keyboard.
The Grid *Sensory Software* www.sensorysoftware.com	Win	On-screen Keyboard	Rectilinear grids of controls can be created and used by switch and mouse pointer users to control applications. . It can be accessed by switch users and pointer users.
Wivik *Liberator / Hugh Macmillan Centre* www.wivik	Win	On-screen Keyboard	A number of different keyboards may be accessed from WiViK to match a number of languages and layouts. It has a series of fixed keyboards and built in word prediction. . It can be accessed by switch users and pointer users.
EZKeys *Words+* www.words-plus.com	Win	On-screen Keyboard	A fixed keyboard and mouse emulator that has been versioned from the original DOS program. It has a series of fixed keyboards and built-in word prediction.

Product	For	Type	Description
Clicker *Crick Software* www.cricksoft.com	Win	On-screen Keyboard	Clicker is mainly used as a stand-alone program that uses text-to-speech and symbols to support writing. It has its own writing window. It can be accessed by switch users and pointer users. It can be used to output from its keyboard (grid) to any windows application. Grids may be created and edited.
Windows Switch *AU Enterprises* www.advisory-unit.org.uk	Win	On-screen Keyboard	A simple grid on screen keyboard utility for single switch users. Keyboard and mouse controls can be accessed to operate applications.
Joystick / Trackerball *Traxsys* www.traxsys.com	Win Mac	Mouse alternative	Large joysticks and trackerballs with supplementary switches for drag and double click etc. They can also have hardware speed controls,
HeadMouse *Origin Instruments* www.orin.com	Win Mac	Mouse alternative	An external mouse emulator that uses reflections of infra-red light from a spot worn by the user to send mouse emulated movements to a mouse port.
SmartNav AT *Naturalpoint* www.naturalpoint.com	Win Mac	Mouse alternative	A USB camera that tracks head movements by tracking a reflective spot worn by the user and translated into mouse movements.
Intellikeys *Inclusive Technology* www.intellitools.com	Win Mac	Keyboard	A membrane keyboard that can be internally programmed with keyboard and mouse macros. The keyboard itself connects to either a PS/2 or USB port. It can also be used as a switch input interface.
Concept Keyboard *Traxsys* www.traxsys.com	Win	Keyboard	A membrane keyboard that can be programmed with keyboard macros. It requires a utility program running in the computer to contain the macros. The keyboard itself connects to a serial or USB port.
King and Mini Keyboards *Tash Inc* www.tashinc.com	Win Mac	Keyboard	Enlarged or miniaturised keyboards based on membrane technology. They contain their own time filtering systems to eliminate tremor and unwanted keyboard repeats.
Joystick to Mouse *Sensory Software* www.sensorysoftware.com	Win	Utility	There are a number of utilities available that will convert games joystick control input into mouse movements.
Jaws *Freedom Scientific* www.freedomscientific.com	Win	Utility	A screen reader that will take text from the screen and speak it using text-to-speech software.
Intellimouse *Microsoft* www.microsoft.com	Win	Utility	A utility for Microsoft Intellimouse or Trackballs that gives a much wider control to mouse speeds and click functions.

Appendix 3:
IBM Software Accessibility Checklist

Version 3.5.1 (28 May 2003)

http://www-306.ibm.com/able/guidelines/software/accesssoftware.html

1: Keyboard access

1.1 Provide keyboard equivalents for all actions.

1.2 Do not interfere with keyboard accessibility features built into the operating system.

2: Object information

2.1 Provide a visual focus indicator that moves among interactive objects as the input focus changes. This focus indicator must be programmatically exposed to assistive technology.

2.2 Provide semantic information about user interface objects. When an image represents a program element, the information conveyed by the image must also be available in text.

2.3 Associate labels with controls, objects, icons and images. If an image is used to identify programmatic elements, the meaning of the image must be consistent throughout the application.

2.4 When electronic forms are used, the form shall allow people using assistive technology to access the information, field elements and functionality required for completion and submission of the form, including all directions and cues.

3: Sounds and multimedia

3.1 Provide an option to display a visual cue for all audio alerts.

3.2 Provide accessible alternatives to significant audio and video.

3.3 Provide an option to adjust the volume.

4: Display

4.1 Provide text through standard system function calls or through an API (application programming interface) which supports interaction with assistive technology.

4.2 Use color as an enhancement, not as the only way to convey information or indicate an action.

4.3 Support system settings for high contrast for all user interface controls and client area content.

4.4 When color customization is supported, provide a variety of color selections capable of producing a range of contrast levels.

4.5 Inherit system settings for font, size, and color for all user interface controls.

4.6 Provide an option to display animation in a non-animated presentation mode.

5: Timing

5.1 Provide an option to adjust the response times on timed instructions or allow the instructions to persist.

5.2 Do not use flashing or blinking text, objects, or other elements having a flash or blink frequency greater than 2 Hz and lower than 55 Hz.

6: Verify accessibility

6.1 Test for accessibility using available tools.

Appendix 4:
Glossary and Definitions

AAC	Augmentative and Alternative Communication. Using systems other than voice or orthographic writing to communicate.
Accessibility	The measure of how possible it is for people with different needs to control or access information from a system.
API	Application Program Interface – A set of routines, protocols and tools for building software applications.
AT	Assistive Technology
COGAIN	The COGAIN (Communication by Gaze Interaction) project is a pan-European project that aims to integrate eye gaze expertise for the benefit of users with disabilities, focussing on motor-control disorders. www.cogain.org
Concept Coding / Concept Coding Framework (CCF)	A method for rationalising symbolic representation of communications and information. There are a number of different communication symbol sets in use and the Concept Coding database acts as a way of referencing these systems to each other and to orthographic text in any language. A sort of multi-lingual dictionary for symbols and words.
DDA	Disability Discrimination Act, 1995 (UK)
DECO	This ACE Centre project offers the potential to help many people with severe physical difficulties to write with both an ease and speed which, until recently, was not considered possible - many times quicker, in fact, than the commonly used specialised hardware and software. the most popular eye control and head movement systems currently available.
Dwell Selection / Clicking	A method whereby a mouse click is generated or an item selected when the pointer remains stationary (or within a small range) for a set time.
Eye Tracking / Pointing	A system of controlling the mouse pointer by just looking at the point you want it to go.

GUI	Graphical User Interface
Head Pointing	A system of controlling the mouse pointer by moving your head.
HTML	Hypertext Markup Language – The authoring language used to create documents on the World Wide Web
Inclusion	Inclusive design means designing products which as many people as possible can use.
Joystick	These can be switch (four switches – one for each direction- like games pads) or analogue (like games joysticks). These are used to steer the mouse pointer around the screen
On-screen Keyboards	Alternative access software, which replaces the physical keyboard with a virtual one on a computer display. These are usually accessed either by pointing and clicking (including dwell clicking) or scanning with switches.
Re-purposing	Taking information and presenting it to the user in a different format to the user so that they can access the information easier or navigate or search the information in an alternative way that is better suited to their needs.
SAPI	Speech Application Program Interface, system tools available in two versions (4 and 5) from Microsoft. These APIs allows an application to use any compatible voice engine to produce speech from text. Voice engines are available both free (from Microsoft) and purchasable (from RealSpeech, Loquendo and others) in a variety of languages and accents. SAPI 5 is built in to Windows XP.
Scanning	The process of enabling a user to select keys, commands, or controls by presenting them sequentially. Any of the items being scanned can then be selected by pressing a switch for example.
Section 508	Section 508 is US legislation (ADA) which was enacted to eliminate barriers in information technology, to make available new opportunities for people with disabilities, and to encourage development of technologies that will help achieve these goals.
SENDA	Special Educational Needs and Disability Act (UK)
Sticky Keys	A system of allowing full access to key combinations with a single finger – you don't have to hold any keys down (e.g. control + P becomes control followed by P)
Switch (user)	A device used do control scanning to access communication or other activity with a simple on/off state. (A switch user is a person who controls and communicates with usually one or two switches)

Trackerball	The earliest form of graphic pointer control after the keyboard. An 'upside-down' mouse used by people with disabilities and mainstream users to control the mouse pointer. The ball is rolled by hand, foot, chin or other part of the body and the mouse pointer moves accordingly.
TTS	Text-to-Speech, the process of artificially generating spoken, audible language from text.
Usability	The measure of how much effort is needed to access a given system. A system may be accessible i.e. you can do everything, but not simple or easy to use.
Visual perception	The world we see is inside us, an illusion. Most of us share the same internal perception of a real and physical universe outside us. However, some people see things differently and have Visual perception difficulties.
WAI	Web Accessibility Initiative of W3C
XML	Extensible Markup Language – A specification developed by W3C for web documents

Appendix 5:
Key accessibility resources for operating systems

Windows

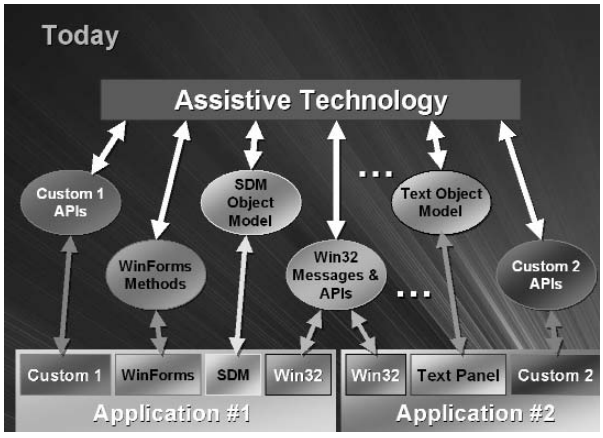
The portal for accessibility at Microsoft is **www.microsoft.com/enable**. An article outlining Microsoft's view of the future of accessibility is available at **www.microsoft.com/enable/business/future.aspx**.

The MSDN support for accessibility can be found at:
msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/accessibility.asp

There is also some general guidance and support available from IBM at :
www-306.ibm.com/able/access_ibm/index.html

Windows 95 ,98 and ME

Although these are now obsolete systems a large number of computers in schools still operate under them.



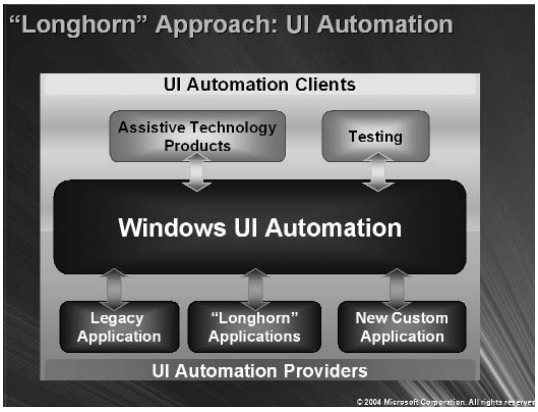
Windows 2000 and XP

This is the system most frequently encountered system, and the one that new developments will currently be aiming at until Vista comes nearer to being available. To see a general user guide to accessibility in XP see **www.microsoft.com/windowsxp/using/accessibility/default.mspx**

Vista (Longhorn)

This is the renamed Longhorn, the next generation system from Microsoft. This will contain UI-Automation.

Windows UI Automation Defined



From: 'Microsoft Confidential' Providing Access: Innovation FOR Society - Gary M. Moulton, Product Manager Accessible Technology Group

- A key part of the new accessibility model for Windows
- Gathers information about the user interface (UI)
  - Dynamically discovers UI structure
  - Extracts property information
  - Receives event notifications when UI changes
  - Queries an element for its behavior
- Interacts with UI elements
  - Clicks a button, scrolls a list, moves a window, etc.
  - Injects keystrokes and mouse input

Key Innovations in UI Automation:

- Logical tree – structure of the UI
  - Stitches all UI trees into one coherent structure
  - Resembles the structure perceived by an end user
- Properties – important UI information
  - Name, Bounding Rectangle, Persistent ID, etc.
- Control patterns – control behavior
  - Scroll, Selection, Window, ExpandCollapse, etc.
- Events – notification of UI changes
- Window creation, change in focus or selection, etc.

Benefits of Windows UI Automation:

- Automation framework built into "Longhorn"
  - Platform-level support for automating all UI elements
  - Exposes a consistent object model for all controls
  - Independent of locale, machine and resolution
- Creates new opportunities for innovation in:
  - Assistive technology products
  - Automated UI testing
  - Command-and-control utilities

Macintosh

The home page for Macintosh accessibility is found at **www.apple.com/accessibility**. This gives a guide to the accessibility features within Mac systems. For developers there is further

support to be found at: **developer.apple.com/documentation/Accessibility**. The document "Introduction to Apple Human Interface Guidelines" is a good starting point: (**developer.apple.com/documentation/UserExperience/Conceptual/OSXHIGuidelines/index.html**). The document is available in PDF format and as a Web document.

Mac OS 8/9

Again these are no longer supported, but Macintosh computers running these systems (and earlier!) are still in use. Support can still be obtained by searching for "OS 8" or "OS 9" (note space) in the accessibility area of this web site.

Mac OS X

This is the current system that new developments will be aimed at and is the one best covered by the available support. OSX has many universal accessibility features and an overview of these can be found at **www.apple.com/accessibility** and **www.apple.com/macosx/features/universalaccess**. The developer pages for accessibility can be found at: **developer.apple.com/referencelibrary/GettingStarted/GS_Accessibility/index.html**

Unix

KDE Accessibility Project

(KDEAP) is the main forum for accessibility for Unix (**accessibility.kde.org**) The KDEAP is a small on-line community of developers and other volunteers dedicated to ensure that our favorite desktop is accessible to all users, including those of us with physical handicaps.

Linux

Because of the open source development of this system, a number of different 'flavours' of Linux including RedHat Linux, Turbo Linux, GNU Linux and GNOME have developed.

Disability access to GNOME

Accessibility is enabling people with disabilities to participate in substantial life activities that include work and the use of services, products, and information. We define GNOME Accessibility as the suite of software services and support in GNOME that allows people with disabilities to utilize all of the functionality of the GNOME user environment.

Linux Accessibility Resource Site

lars.atrc.utoronto.ca

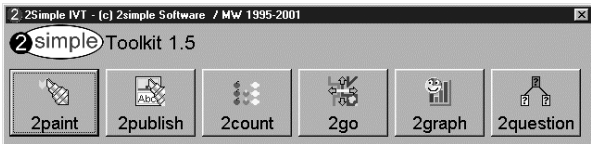
A summary of the work of the Linux Accessibility community - Currently edited by David Bolter and hosted by the Adaptive Technology Resource Centre (ATRC)

The Linux Accessibility HOWTO

This covers the use of adaptive technologies that are available for the Linux operating system, as well as the software applications and hardware devices that can be installed to make Linux accessible to users with disabilities. The information provided targets specific groups of individuals with similar disabilities. See **www.tldp.org/HOWTO/Accessibility-HOWTO**

Appendix 6:
Software case study

Making the 2Simple video toolbox accessible to switch users



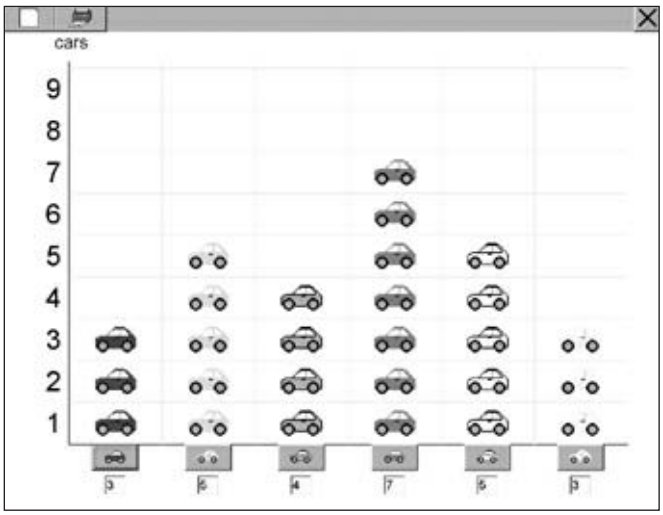
The 2Simple video toolkit was:

> *"Developed to meet the specific needs of Nursery, Reception and KS1 pupils. Created for the UK curriculum, to make developing ICT skills simpler. Six easy to use programs and over 50 example video lessons to create books, borders, envelopes, letters, pictures, pictograms, graphs, teach directions, make branching databases."*

The creators of 2Simple approached the ACE Centre for advice as to how they might make this suit of programs accessible and more inclusive. The programs were at a fairly late stage in their development and radical changes to the design were not possible without undue expense.

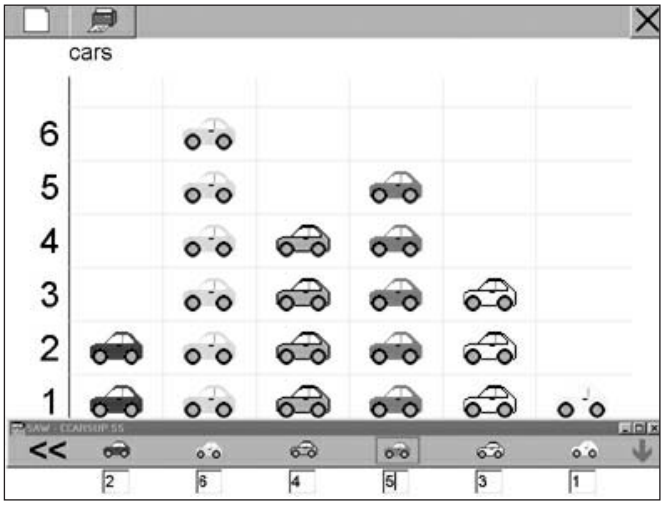
Modifying the design

The original design was very much based on 'point and click' controls, most of which were accessed through buttons. The simplest way to make the program accessible to switch users was to make it keyboard controlled and use a switch utility (in this case the ACE Centre's SAW switch accessing software) to act as the keyboard interface.



Counting in the original 2*Count* element of the program (shown left) is done by clicking on the button beneath each column. In this example we are counting the number of cars of each colour.

The result



SAW was able to produce a scanning system that closely matched the original interface in its appearance and function (see screen below). The result was a full implementation which could be trimmed for simpler use. In the case of '2Count' within the same software suite, SAW replaced the buttons with scanned items that carried out the same function.

The simple expedient of adding keyboard controls for all buttons and controls made the product totally accessible and quite efficient for switch users. Earlier collaboration with the company would have ensured higher levels of cost effective efficiency.

Problems encountered

- The access system was in parts more cognitively complex than the tasks themselves.
- Switch users were left with restrictions in drawing and other mouse driven environments.
- Making the switch users interface match the program visually was time-consuming, in particular, giving the switch users controls the same graphics as the original program.
- Matching the screen resolutions meant two versions of the switch interface and time restricted the final version to 800 x 600 resolution only.
- Some of the keyboard controls did not have an absolute result, for example, only moving up and down the list of available colours and not allowing any colour to be selected immediately by the switch user. This meant that some of the scanning selection was more awkward and slower that it should have been.

Lessons for the future

It would be simpler and quicker if mainstream interfaces were developed that could be scanned either directly or indirectly from the keyboard. This should be by highlighting control buttons in sequence. This could be by arrow or tab – shift+tab keys. Such scanning is common in Windows where the tab key is often used. However the standard highlighting is very poor. The alternative interface could then simply be used to scan the application directly rather than having to program in the entire keyboard shortcuts.

This would reduce the need for complex groups of similar scanning selection sets (grids) with different icons. Having to accommodate different numbers of buttons on similar screens is particularly difficult and time consuming to construct.

In the future the use of UI Automation combined with 'intelligent' switch interfaces could enable users to access many programs without intervention by a designer of switch access systems.

Taking these ideas further

The ideas contained in these guidelines are only first principles. If you are interested in taking these ideas further, then please contact one of the many national and international centres that have expertise and experience in this field (Appendix 1). Their advice and consultation is often free and independent. And if they don't know the answers to your specific questions, they will surely know someone who does! There is enormous benefit to be gained from consultation during the early stages of software or web site design. Don't leave it until the product development phase when it may be too late to change even the simplest things. Each software application has features that may require considered treatment, and centres such as the ACE Centres in England will be more than willing to provide you with help and advice.

So, if you are a publisher, software developer, or web site designer working in the multimedia field, please initially contact:

David Colven – The ACE Centre, Oxford, UK
colven@ace-centre.org.uk

Andrew Lysley – The ACE Centre, Oxford, UK
lysley@ace-centre.org.uk

References and reading

1 David Colven and Tina Detheridge 1993, CDROM - a matter of access: NCET (now BECTa), ISBN 1 85379 267 5.

2 "The Future of Accessible Technology, Microsoft" www.microsoft.com/enable/business/future.aspx (no date)

3 Simeon Keates and John Clarkson 2003, Countering design exclusion,: Springer, ISBN 1 85233 769 9

4 Simeon Keates, John Clarkson, Patrick Langdon and Peters Robinson (eds) 2004, Designing a more inclusive world: Springer, ISBN 1852338199

5 John Gill, Access-Ability, Making technology more useable by people with disabilities,: RNIB, ISBN 1 85048 030 6

6 "A Search for meaning", Duncan Graham-Rowe, New Scientist p21, 29th January 2005

7 ACE Centre Advisory Trust, 1999, Computers and Inclusion: Factors for Success, Ace Centre Advisory Trust, Oxford

8 Best, A, 1992, Teaching Children with Visual Impairments, Milton Keynes, Open University Press

9 Blamires et al., 1999, Enabling Technology for Inclusion, Paul Chapman, London

10 Vincent, T, Wilkinson, A, Sapsed, D, ed. 1999, Guidelines for Accessible Courseware, TTLP Advisory Group, Milton Keynes, The Open University

11 IDEA ,1990, Individuals with Disabilites Act, Public Law, pp.101–476

12 Cooper, D and staff of The Open University Office for Students with Disabilities 1991, Students with Disabilities: Some Guidelines for Tutorial and Counselling Staff, Milton Keynes, The Open University

13 Pester, K, 1993, Hearing IT, Coventry, NCET.

14 Webster, A, 1989, 'Deafness and Learning to read 1:Literacy Across the Curriculum', Journal British Association of Teachers of the Deaf, (13) 3 1989 pp82-92

15 Wells, G, 1987 The Meaning Makers; Children Learning Language and Using Language to Learn, London, Hodder and Stoughton

16 Mick Thomas, 2000, An introduction to inclusive software design – draft paper, BECTa www.becta.org.uk/page_documents/teaching/discussiondocument.pdf

17 Donald Norman, The Design of Everyday Things, The MIT Press 1990l, ISBN 0 385 267746

Document ref: A0010