Design Guidance

Accessibility Principles

Thursday, 5 July 2007

Version 1.0.0.0

Prepared by

Microsoft

TABLE OF CONTENTS

1 Executive Summary 1

2 Related Documents 2

3 Purpose and Scope of Work 3

4 Accessibility Goal for Clinical Applications 4

4.1 Accessibility Goal 4

4.1.1 Basis of the Definition 4

4.1.2 A More Constrained Scope for Clinical Applications 5

5 High Level Requirements for Accessibility of Clinical Applications 7

5.1 Basis of the Requirements 7

5.2 Justification and Explanation of the Requirements 8

6 Document Information 17

6.1 Terms and Abbreviations 17

6.2 Definitions 17

6.3 Nomenclature 17

6.3.1 Body Text 17

6.3.2 Cross References 17

6.4 References 18

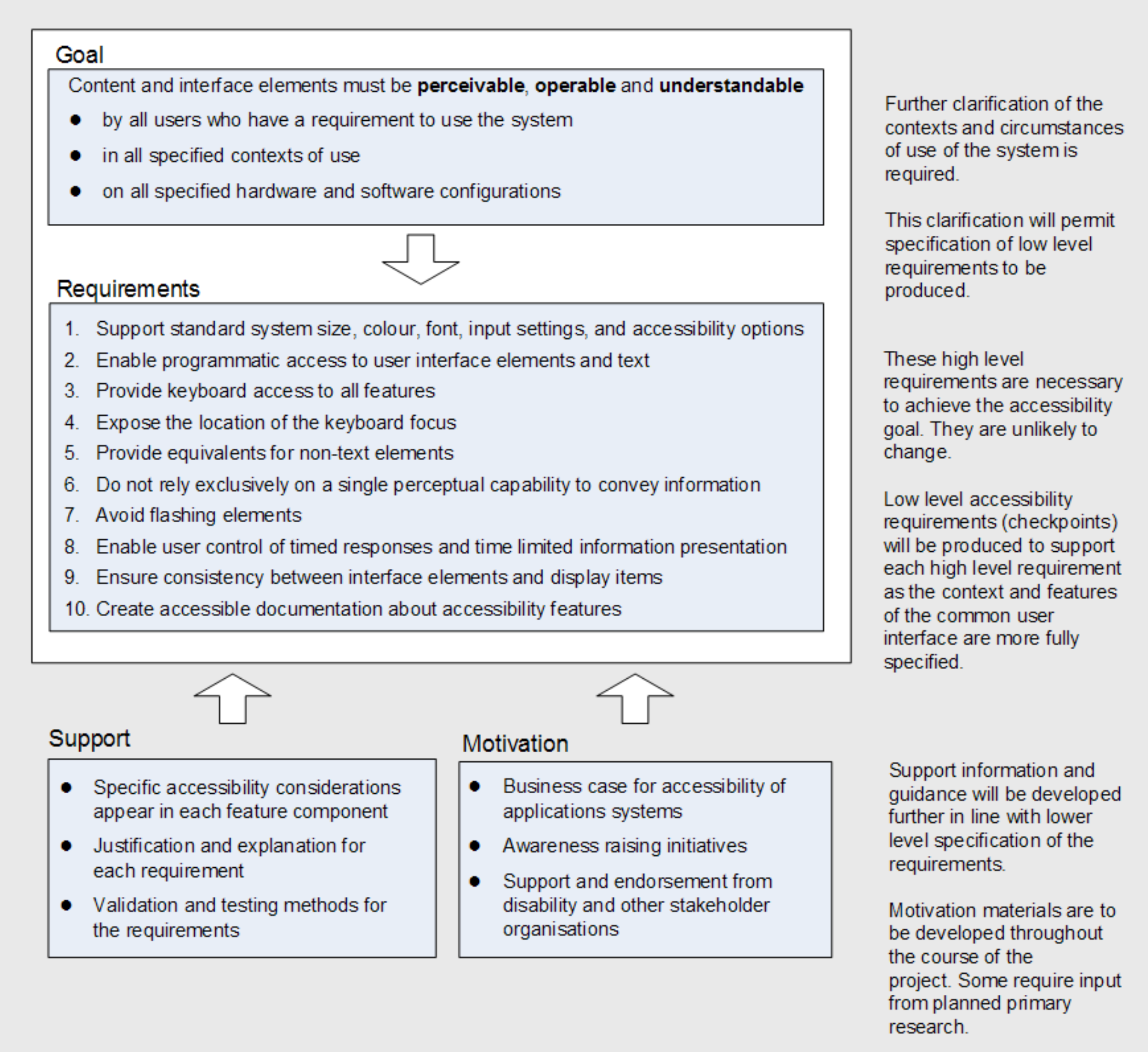
APPENDIX A Current Assistive Technologies 19

1 EXECUTIVE SUMMARY

Accessibility is an essential requirement for the success of clinical applications. This requirement is underlined by commitment to equality, an obligation under the anti-discrimination legislation.

Based on reviews of relevant literature, and current best practices, this document presents the goals, requirements, guidance and support initiatives which describe a framework for how clinical application user interfaces should implement accessibility.

At this stage, only high level information is presented. Future work will expand the depth of information to provide a complete framework for developing and validating accessibility. Figure 1 displays a summary of the Accessibility Framework.



Goal

Content and interface elements must be **perceivable**, **operable** and **understandable**

- by all users who have a requirement to use the system
- in all specified contexts of use
- on all specified hardware and software configurations

Further clarification of the contexts and circumstances of use of the system is required.

This clarification will permit specification of low level requirements to be produced.

Requirements

1. Support standard system size, colour, font, input settings, and accessibility options
2. Enable programmatic access to user interface elements and text
3. Provide keyboard access to all features
4. Expose the location of the keyboard focus
5. Provide equivalents for non-text elements
6. Do not rely exclusively on a single perceptual capability to convey information
7. Avoid flashing elements
8. Enable user control of timed responses and time limited information presentation
9. Ensure consistency between interface elements and display items
10. Create accessible documentation about accessibility features

These high level requirements are necessary to achieve the accessibility goal. They are unlikely to change.

Low level accessibility requirements (checkpoints) will be produced to support each high level requirement as the context and features of the common user interface are more fully specified.

Support

- Specific accessibility considerations appear in each feature component
- Justification and explanation for each requirement
- Validation and testing methods for the requirements

Motivation

- Business case for accessibility of applications systems
- Awareness raising initiatives
- Support and endorsement from disability and other stakeholder organisations

Support information and guidance will be developed further in line with lower level specification of the requirements.

Motivation materials are to be developed throughout the course of the project. Some require input from planned primary research.

Figure 1: Summary of Accessibility Framework

2 RELATED DOCUMENTS

All components of the Design Guidance contain explicit accessibility considerations and have been developed with accessibility in mind. You should consult these for component-specific recommendations. To date these are:

- *Design Guidance – Address Information Display* {**R1**}
- *Design Guidance – Date Display* {**R2**}
- *Design Guidance – Patient Identification Number Display* {**R3**}

In addition, accessibility forms an integrated part of the research and testing strategy, and many of the outputs of this ongoing research will provide detail for future versions of this guidance.

3 Purpose and Scope of Work

This guidance outlines a framework, proposed at the time of writing, for the achievement of accessibility within clinical applications. The information provided is to assist application designers and technical implementers during the development of user interface (UI) components, and provide a means of validating whether accessibility has been achieved.

This framework sets the goal, identifies the high level requirements necessary to achieve that goal, and provides material to support and motivate the adoption and implementation of accessibility.

At this stage the proposed framework is populated by high level information. Further work will refine the framework and add detail to include:

- Further specification of the scope and constraints relating to accessibility provision within particular circumstances
- Additional low level accessibility requirements in the form of checkpoints to support the high level requirements specified in this guidance
- Additional detailed guidance relating to the technology specific implementation of all specified requirements, and procedures and methodologies for validating whether requirements have been met
- Additional material to support the raising of awareness of accessibility issues, and pointers to bodies of material generated throughout the awareness raising initiatives planned. These are to include:
  - Personas of health care workers with accessibility needs, to illustrate the reality of the situation
  - A video library that shows how people who are reliant on assistive technology cope with good and bad software. This will also illustrate how each of the proposed feature components perform for people with special needs

In addition, accessibility considerations and accessibility testing are key features that will support the development of all component features that will appear in the Microsoft Health Common User Interface (CUI).

4 Accessibility Goal for Clinical Applications

This section presents the proposed accessibility goal for clinical applications. The proposal takes into account the fact that clinical applications will be deployed and used in specific contexts that to a certain extent can be defined.

These contexts will require clarification by means of further primary research, further development of component features, and by input from other factors such as hardware platforms. Upon collection of this information, the accessibility requirements for specific interface components can be more rigidly defined and detailed.

4.1 Accessibility Goal

The accessibility goal is defined as:

Content and interface elements must be **perceivable**, **operable** and **understandable**

- By all users who have a requirement to use the system
- In all specified contexts of use
- On all specified hardware and software configurations

4.1.1 Basis of the Definition

The above definition is based in part on the 'overview of design principles' contained in the working draft of the new Web Content Accessibility Guidelines version 2.0[1] {**R4**}:

"The overall goal is to create Web content that is perceivable, operable and understandable by the broadest possible range of users and compatible with their wide range of assistive technologies, now and in the future. The basic principles include:

1. *Content must be perceivable*
2. *Interface elements in the content must be operable*
3. *Content and controls must be understandable*
4. *Content must be robust enough to work with current and future technologies"*

Similar principles appear elsewhere in modern accessibly literature. These principles are chosen as a base because together they represent current thought about the true nature of accessibility, namely:

- Accessibility is a practical goal that specifies real use by people rather than being defined as a technical compliance standard. Perception, ease of operation, and understanding are human factors and this is important to keep in mind.
- Accessibility is not restricted to issues of disability rather it can affect all potential users of a system depending on their environment, hardware devices or personal circumstances. This recognises that accessibility has a wide scope with many potential benefits and should not be restricted to considerations of limited classes of users such as screen reader users who are blind.

[1] The World Wide Web Consortium (W3C) Web Content Accessibility Guidelines 2.0: http://www.w3.org/TR/2004/WD-WCAG20-20040311/#overview-design-principles

4.1.2 A More Constrained Scope for Clinical Applications

The deployment of clinical applications is commonly under more controlled conditions than the Web and general publicly available software. This permits the formulation of a more constrained and practical definition of context that recognises knowledge of the following factors:

- People
  The range of abilities and disabilities of healthcare workers and how these relate to the different job roles and therefore interactions with specific system features or components.
- Environment
  The range of physical locations and situational and environmental circumstances where machines running the applications will be deployed and how this relates to interactions of specific system features or components.
- Equipment
  The range of specified hardware and software devices that will be supported and how this relates to interactions with specific system features or components.

Information on all of the above factors needs to be gathered and analysed. Note that there are also interactions between people, environment and equipment that specify the full context of use. These complex interactions will need to be fully researched. The information should be used to set specific accessibility requirements within the context of how they will actually be used in practice.

> **Note**
>
> This is important because for particular specified components of healthcare applications certain accessibility requirements will not apply, and will therefore be excluded from any validation or conformance process.

Refining the scope should be an ongoing process involving primary user research, analysis of available existing documentation and statistics, interaction with design and technical resources, and incorporation of specifications such as the supported hardware platforms.

Following are some example hypotheses that remain to be proved or disproved. They are provided to illustrate how information pertaining to people, environment and equipment will help refine the accessibility requirements.

4.1.2.1 Example Hypothesis 1 (People)

Healthcare workers who are blind are never required to rely on information that is only available in an X-ray image.

Comment:

If the hypothesis was proved to be true then no *equivalent* for this non text element (the X-ray) would be required *(*See Section 5, *High Level Requirements for Accessibility of Clinical Applications,* requirement 5*).* A clear indication that it was an X-ray image would be required, but a description of what the X-ray showed (the equivalent) would not be required.

Similar considerations would also be likely to apply to other information sources that exist as purely visual representations.

4.1.2.2 Example Hypothesis 2 (People and Equipment)

Healthcare workers with special needs caused by disability will be supplied with the latest versions of a specified set of adaptive software in order to fulfil their roles.

Comment:

If this hypothesis was proved true then only the latest versions of adaptive software need be supported. This would influence for example the choice of recommended keyboard shortcuts to avoid conflict with adaptive technologies. See APPENDIX A for a listing of current adaptive technology sources.

4.1.2.3 Example Hypothesis 3 (Equipment)

All software will run on a minimum specification of hardware and software.

Comment:

Knowledge of this minimum specification will mean that accessibility considerations relating to legacy hardware and software can be excluded from the accessibility recommendations.

> **Note**
>
> At present no available strong hypothesis relates to environment alone. Environmental factors include lighting levels, noise levels, likelihood of distraction and disruption, position of monitors and input devices.
>
> It is likely that an examination of the possible environmental factors will reveal a very broad variety that will only strengthen the requirement for the flexibility that comes from good accessibility.

5 High Level Requirements for Accessibility of Clinical Applications

Ten high level requirements have been derived from an analysis of eight existing publicly available documents (see Section 5.1, *Basis of the Requirements*) pertaining to the accessibility of desktop and web applications:

To achieve accessibility the following 10 requirements must be met:

1. Support standard system size, colour, font, input settings, and accessibility options
2. Enable programmatic access to user interface elements and text
3. Provide keyboard access to all features
4. Expose the location of the keyboard focus
5. Provide equivalents for non-text elements
6. Do not rely exclusively on a single perceptual capability to convey information
7. Avoid flashing elements
8. Enable user control of timed information presentation and responses
9. Ensure consistency between interface elements and display items
10. Create accessible documentation about accessibility features

These high level requirements are concepts that are to a large extent technology independent and thus are not liable to change significantly regardless of the underlying technical specification used to implement components of clinical applications. The requirements however may be modified slightly and additional requirements specified in the light of further work performed. Slight modifications may also be made to increase clarity.

In addition to the ten requirements above it is further recommended that:

- User customisation at the application/interface level is included whenever beneficial for usability and accessibility
- Attention is applied at the design and implementation stages to the navigational flow and contextual feedback mechanisms – these must be logical and where needed modifications made to make them more accessible
- Applications, interface components and display items are user tested with participants that include users of adaptive technology. This is important if the accessibility goal is to be truly realised

5.1 Basis of the Requirements

The ten high level requirements above are derived from an analysis of 180 guidelines/requirements extracted from eight authoritative and publicly available documents that pertain to accessibility for software and web applications. These sources are:

- ISO/TS 16071:2003: Ergonomics of human-system interaction – Guidance on accessibility for human-computer interfaces[2] {**R5**}

[2] International Organization for Standardization, ISO/TS 16071:2003: Ergonomics of human-system interaction – Guidance on accessibility for human-computer interfaces: http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=30858&ICS1=13&ICS2=180&ICS3=

- Microsoft Optimized for Accessibility Guidelines from "Designed for Windows XP" Application Specification[3] {**R6**}
- IBM Software Accessibility Checklist[4] {**R7**}
- W3C WAI Authoring Tool Accessibility Guidelines 1.0[5] {**R8**}
- W3C WAI Web Content Accessibility Guidelines 1.0[6] {**R9**}
- W3C User Agent Accessibility Guidelines 1.0[7] {**R10**}
- United States Access Board: Software Applications and Operating Systems(1194.21)[8] {**R11**}
- Irish National Disability Authority IT Accessibility Guidelines: Accessibility Guidelines for Application Software[9] {**R12**}

Guidelines/requirements from each of the documents were extracted and listed in a spreadsheet. The aim at this stage was to extract high level themes and concepts that are vital to the accessibility of software applications. This was performed by an analysis that grouped the extracted guidelines/requirements into discrete categories. The categories were formed such that all members of a category share a common theme, and there is minimal cross over between categories. The theme/concept that formed each of the categories created the ten requirements listed above.

The eight documents analysed differed significantly in their level of detail and granularity, and in their specificity to software applications. This was taken into account when choosing which level of guidelines from each document to include in the analysis. The process was to first include top level guidelines and include lower level guidelines only when they increased the scope and clarity of the guidance rather than the implementation specificity. Information provided at the lower levels was used to aid interpretation and assist with the categorisation.

5.2 Justification and Explanation of the Requirements

Numbers 1 to 5 of the proposed recommendations are for the most part technical implementation issues. Numbers 6 to 10 are for the most part interface design issues. While this is a broad distinction you should be aware of all issues whether you are a technical implementer or interface designer. Considerations of all appropriate issues appear in the component feature accessibility sections.

The sections below explain key aspects of each of the ten requirements. Reasons for the importance of each requirement are provided along with guidance for achieving the requirement. Potential exclusions and implementation details are also given although these remain to be refined in future versions of this document.

[3] Microsoft: Designed for Windows XP Application Specification: http://www.microsoft.com/downloads/details.aspx?FamilyID=209e3d65-f0be-4eef-8602-73bb9bc29d54&DisplayLang=en

[4] IBM: IBM Software Accessibility Checklist: http://www-306.ibm.com/able/guidelines/software/accesssoftware.html

[5] The World Wide Web Consortium (W3C): Authoring Tool Accessibility Guidlines 1.0: http://www.w3.org/TR/WAI-AUTOOLS/

[6] The World Wide Web Consortium (W3C): Web Content Accessibility Guidelines 2.0: http://www.w3.org/TR/2004/WD-WCAG20-20040311/#overview-design-principles

[7] The World Wide Web Consortium (W3C): User Agent Accessibility Guidelines 1.0: http://www.w3.org/TR/WAI-USERAGENT/

[8] United States Access Board: Software Applications and Operating Systems (1194.21): http://www.access-board.gov/sec508/guide/1194.21.htm

[9] Irish National Disability Authority IT Accessibility Guidelines: Accessibility Guidelines for Application Software: http://accessit.nda.ie/technologyindex_4.html

Support standard system size, colour, font, input settings, and accessibility options

Reasons for the Requirement

Many users of the system will need to adjust system settings, or utilise system accessibility options in order to make application use easier or even possible at all.

In Windows® XP, a wizard takes users through a customisation process designed to match their particular needs. This wizard modifies font size, screen resolution, the size of borders and window controls, icon size, mouse cursor appearance and behaviour, cursor blink rate and width, setting of visual warnings of system events, special options of keyboard behaviour. In addition a special high contrast setting can also be applied. Supported modifications include:

- Sticky Keys - Enables the user to generate combination key presses, such as Ctrl+Alt+F, by pressing the keys one at a time.
- Toggle Keys - Causes audible and visible alerts to be generated when the Caps Lock, Num Lock or Scroll Lock keys are pressed.
- Sound Sentry - Ensures that system sounds (beeps) are accompanied by a visible alert, such as a screen flash.
- Repeat Keys - Enables the user to adjust the rate at which keys repeat when held down.
- Slow Keys - Enables the user to adjust the length of time keys must be held down before the key press is accepted.
- Bounce Keys or Filter Keys - Prevents the keyboard from accepting quick consecutive presses of the same key.
- Mouse Keys - Enables the user to move the mouse pointer using the arrow keys.
- In-built Screen reader (narrator) - Reads aloud the screen contents in a synthetic voice.
- In built Screen magnifier - Magnifies the area of the screen around the focus point and displays it in a separate window.
- On-screen keyboard - A software keyboard, displayed on the screen, which emulates the hardware keyboard.

Support for these settings will benefit a wide range of people, for example:

- People with visual impairments who require larger text and higher contrast in order to be able read the screen.
- People with dyslexia who find reading is made easier using specific fonts and certain background and foreground colour combinations.
- Any individual who finds that larger text is easier to read, avoiding eyestrain.
- People who use a mouse, especially those with visual impairments, or mobility impairments, who find large targets easier to locate and activate than smaller ones.
- People with mobility impairments who use the 'sticky keys' setting because they have difficulty with pressing multiple keys simultaneously.
- People with hearing impairments, or in noisy environments, who will benefit from visual representations of system events.

Guidance

Standard Windows controls support the system-wide settings. You should use these where possible.

Standard controls written to W3C HTML 4.01, XHTML 1.0 and CSS2 specifications automatically support this requirement when executed by the Microsoft HTML parsing and rendering engine. When using these and other W3C technologies you should use the latest release of the technology and ensure that you code to the published standard.

Care should be taken in the following cases:

- Creating custom controls
- Creating owner-drawn controls
- Altering normal standard control behaviour
- Executing custom message handling
- Handling low-level input that bypasses normal mouse and keyboard messages

Exclusions

The High Contrast requirements do not apply to certain application features where the use of colour is intrinsic and indispensable to the goal of the feature. Examples include:

- Palettes or swatches where the user selects from a range of displayed colours. In this case, the application can display the colour but should provide a text description such as a name (light blue) or numeric value (RGB 0, 255, 255).
- Palettes Animation, video, and graphic images when the content is available through other means.

Enable programmatic access to user interface elements and text

Reasons for the requirement

People who depend upon assistive technologies such as screen readers or speech recognition software, depend upon information about the user interface components and text being made available to those technologies. Their use of the system would be impossible if programmatic access was not provided.

Guidance

At the time of writing, Microsoft Active Accessibility® (MSAA) is the recommended technology for providing programmatic access to UI elements and text. MSAA is the technology supported by most modern assistive technologies. See Microsoft Active Accessibility[10] {**R13**} for more information.

Your application must provide programmatic access to the following user interface elements and text.

- All UI elements - allows assistive technologies to identify and manipulate your application’s UI elements.
- Descriptive titles on windows, frames, objects, and pages - allows people using assistive technologies, especially screen readers, to use the title to understand the context of the frame, object, or page in the navigation scheme.
- Alternative text - allows assistive technologies to provide text descriptions of non-textual UI elements, such as graphics.

[10] Microsoft Active Accessibility: http://msdn2.microsoft.com/en-us/library/ms697707.aspx

- Text content - allows assistive technologies to access the content of an application, such as the text in a document, and describe it to the user.
- Data tables - allows assistive technologies to help users understand the information in a table.

Provide keyboard access to all features

Reasons for the requirement

Many people prefer to operate software applications via the keyboard rather than a mouse. Some people depend upon the keyboard in order to use applications at all, these include:

- People who are blind
- People with mobility impairments
- People who use assistive technologies that translates their actions into keystrokes
- People who do not have a functional mouse available

Guidance

Standard Windows controls support all of the required keyboard behaviour when given the correct labels and attributes.

You should explicitly provide keyboard support in the following cases:

- Creating custom window classes or controls
- Altering the normal behaviour of a standard window or control. The altered element should support keyboard behaviour that is equivalent to the behaviour of the standard control.
- Assigning keyboard navigation in windows or controls, such as dialog boxes
- Using speech recognition technology for user input or commands
- Creating controls using markup language
- Using client-side scripting or plug-ins for user input or commands

Exclusions

- Application components that rely exclusively on specialised input devices, for example graphing tablets.

Expose the location of the keyboard focus

Reasons for the requirement

Some assistive technologies depend on the location of the keyboard focus to provide appropriate information to the user. Without this information, some people would be unable to use the application at all. For example:

- Screen reader software uses the location of keyboard focus to determine which text or object information to translate into synthesised speech for people who are blind.
- Screen magnification software uses the location of keyboard focus to determine which area of the screen to enlarge for people who are visually impaired.

Guidance

The application’s active window should display a visual focus indicator at all times so that users can anticipate the effects of their keystrokes.

Applications should programmatically expose the keyboard focus through Microsoft Active Accessibility. If using Active Accessibility is not feasible, the application can indicate the focus location by moving the system caret. The caret is normally the blinking vertical bar that the user sees when editing text, but it can be placed anywhere on the screen, made any shape or size, and even made invisible. If it is invisible, it can be moved to indicate the focus location to applications without disturbing what the user sees on the screen.

Exclusions

- Application components that rely exclusively on specialised input devices, for example graphing tablets.

Provide equivalents for non-text elements

Reasons for the requirement

Some users of the system will not be able to perceive non-text elements unless a text description which can be translated into alternate forms, or another suitable equivalent representation, is provided. For example:

- People who are blind or visually impaired may not be able to see essential non-text elements and may need a textual description to effectively use the application.
- People who are deaf may not be able to hear essential non-text elements and also may need an alternative, such as text captioning, to effectively use the application.

Non-text elements include:

- graphics
- animations
- sounds
- video
- Web elements such as image maps, embedded applets, non-accessible plug-ins
- certain document formats, such as pdf, if not rendered for accessibility

Guidance

All graphics require an appropriate text equivalent. The equivalent should not be a literal description of the element but convey the same information, or designate the same function.

Complex graphics and animations may require alternatives such as long descriptions or alternative text versions. In some cases, they may also benefit from additional alternatives such as audio narration.

Auditory information requires text descriptions and/or captions.

Audio video content requires synchronised captions and audio descriptions.

In some cases of multimedia presentation a transcript may be all that is required, however this must offer equivalent information such that the transcript reader is not disadvantaged.

Exclusions

- Non-essential information such as decoration. A literal description should not be provided, instead a NULL value should be given that would be ignored by assistive technologies such as screen readers.
- Non-text elements which have the equivalent information presented elsewhere in an appropriately accessible form, for example an icon with a redundant text label.
- Where non-text information is impossible to translate into text, for example information that is purely visual.

Do not rely exclusively on a single perceptual capability to convey information

Reasons for the requirement

For reasons of disability, circumstance or preference, some people may not be able to perceive information presented in certain forms. Consequently, you should avoid presenting information only in a single representation that relies on a single sense or perceptual capability.

Examples of people affected include:

- People with visual impairments who cannot access visual elements
- People with different forms of colour blindness who cannot distinguish certain colour combinations from each other
- People with hearing impairments who will be unable to distinguish sound differences
- People using monochrome displays who will be unable to distinguish colour differences

Typical cases where this occurs include:

- Alarms and alerts indicated by sound alone.
- Differences in the status of items indicated by colour (hue) alone, for example on graphs and charts, or in lists or tables of items.

Guidance

You should not depend on a single modality (sense) for imparting information, or a single perceptual capability. All information items should use multiple redundant coding. For example:

- A new item in a list of items could be indicated by a colour change AND a visual icon (with appropriate text equivalent)
- Categories in a graph could be distinguished by colour AND contrast differences, or colour AND pattern differences
- Alerts should not rely on sound alone. They should also have a visual representation
- Alert symbols should not rely on colour alone, they should also use other properties such as shape and intensity as the basis of information provision

Exclusions

- Text information that can be interpreted and rendered in alternative forms. Provided the full meaning is contained in the text, redundant coding need not be supplied.
- Visual information that has appropriate text equivalents

- System events that appear by default in one form but may be changed by operating system settings to also appear in alternative forms (for example, additional visual or auditory alerts of system events).

Avoid flashing elements

Reasons for the requirement

Displays which flicker or flash can cause photosensitive epileptic seizures in susceptible individuals, particularly if the flash has a high intensity and is within the frequency range between 2 Hz and 59 Hz.

Guidance

- Avoid flickering or flashing between the 2-59Hz ranges.
- Take care with alerts and animations. Where animations are included, ensure smooth transitions between key frames.

Enable user control of timed responses and time limited information presentation

Reasons for the requirement

Completing an operation may require people to carry out a number of separate activities. These may include reading and understanding instructions, choosing the appropriate action, recalling information and making the inputs. Each of these activities will take some time and different users will require different amounts of time, depending on their abilities, their experience with the system, and circumstances of their use of the system. For example:

- Recalling information such as passwords or personal details is more difficult for people who are tired or stressed
- People who have limited experience with the system will take longer to interpret and react to alerts
- People who have physical impairments will generally take longer to press buttons or type information using a keypad
- People reliant on assistive technologies such as screen readers or screen magnification will generally take longer over actions such as filling out forms and reading information

Guidance

Time sensitive responses and time limited information should accommodate the slowest users.

- As a rule of thumb ten times the average response time for each activity should be allowed
- Time outs should be avoided unless critical for function or security
- People should be permitted to adapt the time sensitivities to match their individual requirements. For instance, expert users that are quick may choose short timeouts to increase security. Slower users who need more time to assimilate information and act upon it may choose longer timeouts.

Ensure consistency between interface elements and display items

Reasons for the requirement

Consistency increases the efficiency and effectiveness of use of applications for all users. Where the output from the system is presented in an alternative or restricted form then consistency is even more important and may be critical for effective use at all.

For example:

- Screen reader users typically interact with systems through sound output and keyboard input. Sound only gives a temporal and transient representation of a page or dialogue item as opposed to a more holistic and static visual representation that sighted users typically perceive. Constant changes in where key items appear and their form can critically disrupt use of a system.

  A screen reader may have to constantly search to find a critical item or may miss critical features because they appear in an unfamiliar or unexpected place. Consistency will increase the effectiveness of use, the confidence of use and reduce errors.

- People using screen magnification software only have a limited viewport of information presented on screens. When items appear in familiar and expected locations they are able to find them quickly and use them effectively by shifting their viewport to the appropriate location. Inconsistency of location may mean that items that do not appear in the viewport are difficult to locate or may be missed entirely. In addition, when the visual display is limited to only a few characters the form of information items greatly assist with the correct interpretation of the kind of information on display. For example if telephone numbers are always of the same distinct form they will be immediately recognised as phone numbers even when viewed in isolation and the label is not visible to the user.

Guidance

- Information display items should be displayed in a standard and consistent form.
- Standard dialogues should be used where possible
- The default linear order of elements, as would be presented by a screen reader should be consistent
- The default tab order of elements should be consistent
- The spatial location of interface elements should be consistent
- Keyboard shortcuts should remain consistent across application components unless clearly indicated through changes in context. For example, it may be appropriate to assign alternative keyboard mapping in text editing mode as opposed to common application use.

Create accessible documentation about accessibility features

Reasons for the requirement

Many people have difficulty reading or handling conventionally printed material or using online documentation. Documentation that is accessible enables users who need to use the accessibility features to discover information about these features. This is important to empower them to use the features and considerations you will have built in by addressing the other nine requirements.

Guidance

- All features that benefit accessibility should be documented, for example:
  - Keyboard shortcuts
  - Customisation options such as font, colour combinations, element size, keyboard operations such as sticky keys
  - Features of non standard controls or dialogues
- The documents themselves should be accessible and conform to the accessibility requirements in this guidance
- Documentation should appear in a variety of alternative formats to meet user needs for example:
  - Print
  - Online
  - Multimedia
  - HTML help
  - CD-Rom tutorials

Exclusions

- Only brief explanations of OS supported features along with appropriate pointers to detailed guidance need be supplied

6 Document Information

6.1 Terms and Abbreviations

Abbreviation	Definition
OS	Operating system
MCUI	Microsoft Common User Interface
UI	User Interface

Table 1: Terms and Abbreviations

6.2 Definitions

Term	Definition
Current best practice	Current best practice is used rather than best practice, as over time best practice guidance may change or be revised due to changes to products, changes in technology, or simply the additional field deployment experience that comes over time.
Healthcare worker	A healthcare facility employee who has close contact with patients and their care

Table 2: Definitions

6.3 Nomenclature

This section shows how to interpret the different styles used in this document to denote various types of information.

All content subject to completion, agreement or verification is denoted with highlighting.

6.3.1 Body Text

Text	Style
Code	`Monospace`
Script	
Other markup languages	
Interface dialog names	**Bold**
Field names	
Controls	
Folder names	Title Case
File names	

Table 3: Body Text Styles

6.3.2 Cross References

Reference	Style
Current document – sections	Section number only
Current document – figures/tables	Caption number only
Other project documents	*Italics* and possibly a footnote

Reference	Style
Publicly available documents	*Italics* with a footnote
External Web-based content	*Italics* and a hyperlinked footnote

Table 4: Cross Reference Styles

6.4 References

Reference	Document	Version
R1.	Design Guidance – Address Information Display	1.0.0.0
R2.	Design Guidance – Date Display	1.0.0.0
R3.	Design Guidance – Patient Identification Number Display	1.0.0.0
R4.	The World Wide Web Consortium (W3C) Web Content Accessibility Guidelines 2.0: http://www.w3.org/TR/2004/WD-WCAG20-20040311/#overview-design-principles	2.0
R5.	International Organization for Standardization, ISO/TS 16071:2003: Ergonomics of human-system interaction – Guidance on accessibility for human-computer interfaces. http://www.iso.org/iso/en/CatalogueDetailPage.CatalogueDetail?CSNUMBER=30858&ICS1=13&ICS2=180&ICS3=	1
R6.	Microsoft: Designed for Windows XP Application Specification http://www.microsoft.com/downloads/details.aspx?FamilyID=209e3d65-f0be-4eef-8602-73bb9bc29d54&DisplayLang=en	2.3
R7.	IBM: IBM Software Accessibility Checklist http://www-306.ibm.com/able/guidelines/software/accesssoftware.html	3.5.1
R8.	W3C: Authoring Tool Accessibility Guidelines 1.0 http://www.w3.org/TR/WAI-AUTOOLS/	1.0
R9.	W3C: Web Content Accessibility Guidelines 1.0 http://www.w3.org/TR/WAI-WEBCONTENT/	1.0
R10.	W3C: User Agent Accessibility Guidelines 1.0 http://www.w3.org/TR/WAI-USERAGENT/	1.0
R11.	United States Access Board: Software Applications and Operating Systems (1194.21) http://www.access-board.gov/sec508/guide/1194.21.htm	
R12.	Irish National Disability Authority IT Accessibility Guidelines: Accessibility Guidelines for Application Software http://accessit.nda.ie/technologyindex_4.html	
R13.	Microsoft Active Accessibility: http://msdn2.microsoft.com/en-us/library/ms697707.aspx	2.0

Table 5: References

APPENDIX A CURRENT ASSISTIVE TECHNOLOGIES

The tables below list currently available assistive technologies. For application software the most important technologies to consider are likely to be screen readers and screen magnifiers. Some technologies also provide a combination of outputs such as magnification and speech. It remains to be determined which assistive technologies should be recommended in any testing procedure.

***** denotes technologies that are still in use but that have been superseded, are obsolete or are no longer supported.

Screen readers

Product	Manufacturer	Brief Description
JAWS® for Windows®	Freedom Scientific	(DOS, Windows 95/98/ME and NT) speech and Braille
ASAW	Microtalk	Automatic screen access program - provides access to modern DOS programs
HAL	Dolphin	(DOS, Windows 95/98/ME and NT) speech and Braille
SuperNova	Dolphin	(DOS, Windows 95/98/ME) speech
LookOUT	Choice Technology	(Windows 95/98/ME) screen reader
outspoken	Alva	(Windows 95/98/ME, Macintosh) speech and Braille
Screenreader/2	IBM®	(OS/2) speech and Braille
* Simply Talker	Econonet	(Windows 95/98/ME) speech
Slimware window bridge	SynthaVoice	(DOS, Windows 3.x and 95/98/ME) speech and Braille
* Virgo	Baum	
Window-Eyes™	GW Micro	(DOS, Windows 3.x and 95/98/ME) speech and Braille
* Winkline	Speech systems for the blind	(DOS, Windows 3.x, 95, NT) (May be obsolete)
WinVision	Artic Technologies	(Windows 3.x and 95/98/ME) speech
ZoomText Magnifier Reader	Ai Squared	Screen magnifier or integrated magnifier/reader
MAGic®	Freedom Scientific	Combines great magnification features with true low vision screen reading when purchased with the speech option
LunarPlus	Dolphin	Enhanced Screen Magnifier with speech

Screen magnifiers

Product	Manufacturer	Brief Description
Magnus	Choice Technology	Magnifies everything on the screen up to 16 times
SuperNova	Dolphin	Magnification, speech, Braille - or all three
ZoomText Magnifier	Ai Squared	Screen magnifier or integrated magnifier/reader
BigShot® Screen Magnifier	Ai Squared	Screen magnification for everyday eyestrain relief
MAGic®	Freedom Scientific	Combines great magnification features with true low vision screen reading when purchased with the speech option
Lunar	Dolphin	Screen magnifier
LunarPlus	Dolphin	Enhanced Screen Magnifier with speech

Browsers specifically designed for people with disabilities

Product	Manufacturer	Brief Description
* Braillesurf	Braillenet	Internet browser for visually impaired users
MozBraille	Mozdev	Extension to transform Mozilla or Firefox to a standalone accessible Internet browser
BrookesTalk	Oxford Brookes University	Intelligent web searching. Speech output, screen magnification (no longer supported)
* EIAD	Sarsfield Solutions	A browser which provides enhancements for people with special needs and learning difficulties. Touch screen, simplified language interface
Emacspeak	Emacspeak Inc	Speech enabled environment for EMACS, runs on UNIX or LINUX. Speech output, simple keyboard interface
HomePage Reader	IBM®	Speech based browser, uses Internet Explorer as its engine. Speech output and standard graphical user interface
* Marco Polo	Sonicon	Plug-in for Netscape Navigator with speech and auditory controls. Speech output, audio icons, simple keyboard interface
* MultiWeb	Deakin University	Speech output, screen magnification
Sensus Internet browser	Sensus	Speech output, braille support, special screen fonts
Simply Web 2000	Econonet	A talking interface using the Internet Explorer engine
* WebSound	University of Geneva	A talking interface using the Internet Explorer engine

Browsers with accessibility features

Product	Manufacturer	Brief Description
Amaya	W3C®	Test-bed browser. Implementing emerging web technologies. There are versions for Windows 95/98/ME, Windows NT and UNIX
Arachine	Arachne Labs	Graphical browser for MS-DOS
Lynx	Open Source	Text based browser for UNIX, Windows 95/NT, MS-DOS and MAC OS allowing flexible and powerful text-based access from older platforms
Internet Explorer®, accessibility features	Microsoft®	Microsoft has included many features in Internet Explorer to enhance accessibility
Net-Tamer	Net-Tamer Inc	This package runs under MS-DOS and includes both text-based and graphical browsing capabilities
Netscape Navigator®	Netscape®	Navigator enables enlargement of fonts
Opera for Windows	Opera software	This compact browser for Windows 95/98/ME offers enhanced keyboard navigation and screen magnification

Voice (input) Browsers

Product	Manufacturer	Brief Description
* webHearit	isSound	A telephone-based tool using the telephone keypad as an interface to navigate suitably configured pages
* SpeecHTML	Vocalis	Allows a participating site to provide telephone access using voice commands
* TelWeb	TelWeb Inc	An experimental telephone-based browser allowing access to any site using voice and dialled commands