Human Computer Interaction for Handicap persons.

Miss Rabiya Begum
Mr.Ramesh Manza*
Mrs.Ratnamala Selukar*
Miss Aparna Joshi**
*Department of Computer Science and IT, Dr. B. A. M.University Aurangabad.
** Department of Computer Science, Vivekanand College, Aurangabad.
India
srabiyabegum@rediffmail.com //ramesh_manza@yahoo.com //rsselukar@rediffmail.com// aajoshi2@gmail.com

Abstract

The field of human-computer interface design profits from understanding potential users and exploring difficult design problems. Addressing the design of enabling technology for users with special needs offers both those advantages. Adapting computer interfaces for access and use by people with various physical and cognitive impairments exposes many basic human-computer interface design issues. Likewise, these efforts will result in computer interfaces which are more attractive and usable by all.

Keywords

Disability, human-computer interaction, assistive technologies, rehabilitation engineering, users with special needs.

Introduction

A common creed of human-computer interface designers is "Know the User". The motivation of this paper is that there may large numbers of potential users that the designer does not know, and may not normally take into account when designing a system. Those users are people often identified as having "special" needs. There is an increasing level of recognition that those needs must be met in the design of interfaces [1, 2, and 3]. The objective of the paper is to identify some of those needs and to describe some of the possible approaches to meeting them.

The papers are structured around the needs of different groups of users: Mobility Impairment, Vision Impairment, Speech Impairment, Language Impairment, Hearing Impairment, and Cognitive Impairment. The specific challenges posed to designers are described accompanied by examples of how those can be met. Often these discussions involve the use of case studies. Of course, all the answers are not known, and areas requiring further research and possibly innovative solutions are identified.

This paper will serve to increase awareness of the scope and nature of these needs, and to stimulate interest in research and implementation of systems that enable access to information technologies by users with disabilities.

HCI And Users With Special Needs

Until the advent of interactive computers in the 1970s, the human-computer interface consisted of punched card readers and line printers and any interaction was at a distance in time and space, so the study of human- computer interaction is a relatively young discipline. Until now, it has largely been concerned with the design and development of interfaces which will be usable by the "average" person. But what is average? Lacking any theoretical base, interface designers have tended to rely on instinct and introspection. In other words, the implicit assumption is often that the user resembles the designer. This supposition had led to interfaces which have been designed for users who are 25-year-old males with a Ph.D. in Computer Science who are besotted with the technology [1, 6].

Designing for exceptional users has much broader significance than is often assumed. It is fallacious to think in terms of disabled people and normal people as if they are two clearly distinct groups. We all have a collection of abilities and weaknesses; furthermore we can all be handicapped by our environment [1].

The accommodation of users with special needs is part of the discipline of human-computer interaction because that is where it ought to be. At the same time, though, there is a powerful argument that it is also important because much of the research and development has a broader significance. Great strides have been made in research by addressing difficult problems. This method of research is a model that has long been successfully followed in medicine [1].

It is often not realized that many everyday products originated as inventions to aid people with disabilities. Alexander Graham Bell was interested in the design of hearing aids when he developed the telephone. The cassette tape was originally intended as a format suitable for blind people. The ballpoint pen was designed for people who lacked the dexterity to use a fountain pen [1].

The best contemporary example of this approach in practice is the IPSNI project at Dundee University, Scotland [7]. The aim of this project is to develop a rich, multimedia human-computer interface designed to maximize the communication ability of an operator with severe physical impairments. But the project is proceeding in collaboration with an avionics company. The company is tracing all developments in order to see what discoveries may be of use to pilots, who may be operating within a handicapping environment (such as blacked out, upside down with limbs weighing three times as much as they normally do). Interface designers generally rely to a great extent on the adaptability of the user. This assumption explains why many poor interfaces are still considered usable. But, many of the assumptions about adaptability do not apply if the user has a disability. Therefore the designer must work harder to create a usable system, producing a better interface for all users [1].

Many challenges in addressing the needs of people with physical and cognitive impairments who need or want to use computers still need to be overcome. These challenges conceal numerous basic issues in human- computer interface design. For example: -

Mobility Impairment

How can we support efficient interaction with the computer without the use of a standard keyboard or mouse? What software tools can we provide to minimize the use of difficult devices such as phones, paper, and environmental controls?

Vision Impairment

How can we completely transform our extremely visual interfaces into intuitive and efficient non visual interfaces? What is the actual model of interaction independent from graphical and spatial presentations? [5]

Speech Impairment

How can we support the generation of conversation at 150 words per minute with, at most, two simple switch inputs?

Hearing Impairment

How can we use computers to translate written or spoken English to American Sign Language which is structurally quite different from Western verbal and written languages? [8]

Cognitive Impairment

How can we adapt interfaces to suit many cognitive levels, not just adult novice and expert users? If we address these issues, we will significantly increase the usability and attractiveness of computers for everyone. We will be many steps closer to reaching goals such as:

- distance working
- novel input via speech, gesture or through everyday devices such as a phone
- the paperless office
- software usable by children and adults, hostile and friendly users, novice and expert

Many possible additions to computer interfaces, such as voice input and virtual reality, while merely interesting to "normal" users are required by people with disabilities. These additions allow them to perform daily activities such as working, communication, and controlling their environment [1].

About Disabilities: Background and Design Guidelines

In this section, we discuss some of the needs, capabilities, and assistive technologies used by people with disabilities, and we provide guidelines for improving application accessibility. The brief descriptions in this section do not constitute complete coverage of the wide range of disabilities, capabilities, needs, and individual differences across the population of people with disabilities--we have focused on providing a broad introduction to visual, hearing, and physical disabilities [2].

Use of assistive technologies varies across users and tasks. Our discussion of assistive technologies is not comprehensive, but it does cover many commonly used software and hardware solutions. In reading this section it is important to remember that as with all users, the needs of users with disabilities vary significantly from person to person. Many users with disabilities do not use assistive technologies, but can benefit from small design changes. Other users have significant investments in assistive technologies, but they too can benefit from software that better responds to their interaction needs.

About Physical Disabilities

Physical disabilities can be the result of congenital conditions, accidents, or excessive muscular strain. By the term "physical disability" we are referring to disabilities that affect the ability to move, manipulate objects, and interact with the physical world. Examples include spinal cord injuries, degenerative nerve diseases, stroke, and missing limbs. Repetitive stress injuries can result in physical disabilities, but because these injuries have a common root cause, we address that topic below under its own heading [2].

Many users with physical disabilities use computer systems without add-on assistive technologies. These users can especially benefit from small changes in interface accessibility.

Some users with physical disabilities use assistive technologies to aid their interactions (see Tables 1 and 2). Common hardware add-ons include alternative pointing devices such as head tracking systems and joysticks. The MouseKeys keyboard enhancement available for MS Windows, Macintosh, and X Windows-based workstations allows users to move the mouse pointer by pressing keys on the numeric keypad, using other keys to substitute for mouse button presses. Because system-level alternatives are available, it is not necessary for applications to provide mouse substitutes of their own. The problem of the mouse is a good example of the kind of generic issue that must be addressed at the system rather than application level [2].

Unfortunately, the MouseKeys feature is often time-consuming in comparison to keyboard accelerators, because it provides relatively crude directional control. For tasks requiring drag and drop or continuous movement (e.g., drawing), MouseKeys is also inefficient. On the other hand, because current systems are designed with the implicit assumption that the user has a mouse or equivalent pointing device, many tasks require selecting an object or pressing a control for which there is no keyboard alternative. In these cases, Mouse Keys provides an option. It is clear that future operating environments need to offer effective alternatives for users who may not use a pointing device. It is important that applications provide keyboard access to controls, features, and information in environments that have keyboard navigation. Comprehensive keyboard access helps users who cannot use a mouse [2].

In addition to keyboard navigation, keyboard accelerators and mnemonics are also helpful for users with physical disabilities (as well as blind and low vision users). Whenever practical, commonly used actions and application dialogs should be accessible through buttons or keyboard accelerators. Unfortunately few of the standard accelerator sequences were designed with disabilities in mind. Many key combinations are difficult for users with limited dexterity (e.g., in Motif, holding down Alt-Shift-Tab to change to the previous window in Motif). Nonetheless, use of key mapping consistent with guidelines for the local application environment not only speeds use of applications for users with movement difficulties, but it also increases the effectiveness of alternate input technologies such as speech recognition. Assistive technologies often allow users to define macro sequences to accelerate common tasks. The more keyboard access an application provides, the greater the user's ability to customize assistive technology to work with that application [2].

About Repetitive Strain Injuries (RSI)

Perhaps the fastest increasing disability in today's computerized workplace is repetitive strain injury (RSI). The Occupational and Health Safety Administration reported that 56 percent of all work place injuries reported during 1992 were due to RSI, up from 18 percent in 1981 (Furger, 1993). RSI is a cumulative trauma disorder that is caused by frequent and regular intervals of repetitive actions. Common repetitive stress injuries are tendonitis and carpal tunnel syndrome, although other types of injuries also occur. Symptoms of computer based RSI include headaches, radiating pain, numbness, tingling, and a reduction of hand function. For computer users, mouse movements and typing may be causes or contributors to RSI[2,9].

Sauter, Schliefer, and Knutson (1991) found that repetitive use of the right hand among VDT data entry operators was a factor in causing RSI. They suggested that a change to "more dynamic tasks" could help reduce the likelihood of RSI. In general, users should be given the choice of performing a task using a variety of both mouse and keyboard options. For custom applications involving highly repetitive tasks, consider providing automatic notification for users to take breaks at regular intervals if there is no such capability at the system level [2].

Frequently repeated keyboard tasks should not require excessive reach or be nested deep in a menu hierarchy. The needs of users already having symptoms of RSI overlap significantly with the needs of users with other types of physical disabilities. Assistive technologies such as alternate pointing devices, predictive dictionaries, and speech recognition can benefit these users by saving those keystrokes, reducing or eliminating use of the mouse, and allowing different methods of interacting with the system.

Toggle Keys	Indicates locking key state with a tone when pressed, e.g., Caps Lock.

About Low Vision

The common theme for low vision users is that it is challenging to read what is on the screen. All fonts, including those in text panes, menus, labels, and information messages, should be easily configurable by users. There is no way to anticipate how large is large enough. The larger fonts can be scaled, the more likely it is that users with low vision will be able to use software without additional magnification software or hardware. Although many users employ screen magnification hardware or software to enlarge their view, performance and image quality are improved if larger font sizes are available prior to magnification. A related problem for users with low vision is their limited field of view. Because they use large fonts or magnify the screen through hardware or software, a smaller amount of information is visible at one time. Some users have tunnel vision that restricts their view to a small portion of the screen, while others require magnification at levels that pushes much of an interface off-screen. A limited field of view means that these users easily lose context. Events in an interface outside of their field of view may go unnoticed. These limitations in field of view imply that physical proximity of actions and consequences is especially important to users with low vision. In addition, providing redundant audio cues (or the option of audio) can notify users about new information or state changes. In the future, operating environments should allow users to quickly navigate to regions where new information is posted. Interpreting information that depends on color (e.g., red=stop, green=go) can be difficult for people with visual impairments. A significant number of people with low vision are also unable to distinguish among some or any colors. As one legally blind user who had full vision as a child told, his vision is like "watching black and white TV". In any case, a significant portion of any population will be "color blind". For these reasons, color should never be used as the only source of information -- color should provide information that is redundant to text, textures, symbols and other information. Some combinations of background and text colors can result in text that is difficult to read for users with visual impairments. Again, the key is to provide both redundancy and choice. Users should also be given the ability to override default colors, so they can choose the colors that work best for them [2].

About Blindness

There is no clear demarcation between low vision and true blindness, but for our purposes, a blind person can be considered to be anybody who does not use a visual display at all. These are users who read Braille displays or listen to speech output to get information from their systems.

Screen reader software provides access to graphical user interfaces by providing navigation as well as a Braille display or speech synthesized reading of controls, text, and icons. The blind user typically uses tab and arrow controls to move through menus, buttons, icons, text areas, and other parts of the graphic interface. As the input focus moves, the screen reader provides Braille, speech, or non-speech audio feedback to indicate the user's position (see Mynatt 1994). For example, when focus moves to a button, the user might hear the words "button -- Search", or when focus moves to a text input region, the user might hear a typewriter sound. Some screen readers provide this kind of information only in audio form, while others provide a Braille display (a series of pins that raise and lower dynamically to form a row of Braille characters)[2].

Blind users rarely use a pointing device, and as discussed above, typically depend on keyboard navigation although blind users have screen reading software that can read the text contents of buttons, menus, and other control areas, screen readers cannot read the contents of an icon or image. In the future, systems should be designed that provide descriptive information for all non-text objects. Until the appropriate infrastructure for providing this information becomes available, there are some measures that may help blind users access this information. Software Without such descriptive information, blind or low vision users may find it difficult or impossible to interpret unlabeled, graphically labeled, or custom interface objects. Providing descriptive information may provide the only means for access in these cases. As an added selling point to developers, meaningful widget names make for code that is easier to document and debug. In addition to problems reading icons, blind users may have trouble reading areas of text that are not navigable via standard keyboard features. In Open Windows and MS Windows, for example, it is not possible to move the focus to footer messages. If this capability were built into the design, then blind users could easily navigate to footer messages in any application and have their screen reading software read the content[1,2,3].

TABLE 1. Assistive Technologies for Low Vision and Blind Users

Assistive Technology	Function Provided
Screen Reader Software	Allows users to navigate through windows, menus, and controls while receiving text and limited graphics information through speech output or Braille display.
Braille Display	Provides line by line Braille display of on-screen text using a series of pins to form Braille symbols that are constantly updated as the user navigates through the interface.
Text to Speech	Translates electronic text into speech via a speech synthesizer.
Screen Magnification	Provides magnification of a portion or all of a screen, including graphics and windows as well as text. Allows users to track position of the input focus.

CleaReader Document storage, can connect to an MP3 Player to read back documents, can be updated to read some European languages

Picture of the CleaReader

ScannaR – Scan, Read and Narrate Document storage with voice tagging for naming stored documents, natural voice and Braille output option.

Picture of the ScannaR

Verbose Text to Speech Converter

Verbose is a text to speech program which will read aloud any text or save it as mp3. After we have installed this text reading software we can assign a system wide hot key. Then whenever we want Verbose to read the text on our screen just push that key and the software will read it aloud. Verbose can also save our text documents or emails to mp3 or wav for us to store them on our Pocket PC or MP3 player, such as an iPod, so we can listen to them on our way home[1,2,14].

About Hearing Disabilities

People with hearing disabilities either cannot detect sound or may have difficulty distinguishing audio output from typical background noise. Because current user interfaces rely heavily on visual presentation, users with hearing related disabilities rarely have serious problems interacting with software. In fact, most users with hearing disabilities can use off-the-shelf computers and software. This situation may change as computers, telephones, and video become more integrated. As more systems are developed for multimedia, desktop videoconferencing, and telephone functions designers will have to give greater consideration to the needs of users with hearing impairments.

Interfaces should not depend on the assumption that users can hear an auditory notice. In addition to users who are deaf, users sitting in airplanes, in noisy offices, or in public places where sound must be turned off also need the visual notification. Additionally, some users can only hear audible cues at certain frequencies or volumes. Volume and frequency of audio feedback should be easily configurable by the user.

Currently, voice input is an option rather than an effective general means of interaction. As voice input becomes a more common method of interacting with systems, designers should remember that many deaf people have trouble speaking distinctly, and may not be able to use voice input reliably. Like the other methods of input already discussed, speech should not be the only way of interacting with a system [2].

TABLE 2. Assistive Technologies for Hearing Disabilities

Assistive Technology	Function Provided
Telecommunications Device for the Deaf (TDD)	Provides a means for users to communicate over telephone lines using text terminals.
Closed Captioning	Provides text translation of spoken material on video media. Important computer applications include distance learning, CD-ROM, video teleconferencing, and other forms of interactive video.
Show Sounds	Proposed standard would provide visual translation of sound information. Non-speech audio such as system beeps would be presented via screen flashing or similar methods. Video and still images would be described through closed captions or related technologies. This capability would be provided by the system infrastructure.

Design Guidelines

We have taken the design issues discussed in this paper and condensed them into a list of guidelines contained in Table 3. This table also indicates which users are most likely to benefit from designs that follow these guidelines [2,3].

TABLE 3. Design Guidelines

Design Guideline	Physical	RSI	Low Vision	Blind	Hearing
Provide keyboard access to all application features	X	X	X	X	
Use a logical tab order (left to right, top to bottom, or as appropriate for locale)	X			X	
Follow key mapping guidelines for the local environment	X	X	X	X	
Avoid conflicts with keyboard accessibility features (see Table 4)	X			X	
Where possible, provide more than one method to perform keyboard tasks	X	X			

Where possible, provide both keyboard and mouse access to functions	X	X	X	X	
Avoid requiring long reaches on frequently performed keyboard operations for people using one hand	X	X			
Avoid requiring repetitive use of chorded key presses	X	X			
Avoid placing frequently used functions deep in a menu structure	X	X	X	X	
Do not hard code application colors			X		
Do not hard code graphic attributes such as line, border, and shadow thickness			X		
Do not hard code font sizes and styles			X		
Provide descriptive names for all interface components and any object using graphics instead of text (e.g., palette item or icon)				X	
Do not design interactions to depend upon the assumption that a user will hear audio information					X
Provide visual information that is redundant with audible information					X
Allow users to configure frequency and volume of audible cues			X	X	X

Existing Keyboard Access Features

Designers of Microsoft Windows, Macintosh, and X Windows applications should be aware of existing key mappings used by access features built into the Macintosh and X Windows (and optionally available for MS Windows). These features provide basic keyboard accessibility typically used by people with physical disabilities (see table 2).

In order to avoid conflicts with current and future access products, applications should avoid using the key mappings indicated in Table 4[2].

TABLE 4. Reserved Key Mappings

Keyboard Mapping	Reserved For
5 consecutive clicks of Shift key	On/Off for Sticky Keys
Shift key held down for 8 seconds	On/Off for Slow Keys and Repeat Keys
6 consecutive clicks of Control key	On/Off for screen reader numeric keypad
6 consecutive clicks of Alt key	Future Access use

Conclusion

In this paper we have focused on the Assistive technologies for persons with disabilities. Our future work will be for blind persons as one software called Verbose is a text to speech converter which will read aloud any text or save it as mp3.presently it is working on English text but in future there can be such converters for regional languages.

References

1. sigchi.org/chi95/proceedings/tutors/toc.html
2. .www.sun.com/access/developers/updt.HCI.advance.html
3. Cress C.J. and Goltz, C. C. (1989) Cognitive Factors Affecting Accessibility of Computers and Electronic Devices, in Proceedings of RESNA 12th Annual Conference, New Orleans, Louisiana.
4. Edwards, A. D. N (editor), Extra-Ordinary Human-Computer Interaction, New York: Cambridge University Press (in press).
5. Glinert, E.P. and York, B.W. (1992) Computers and People with Disabilities, in Communication of the ACM, 35 (5) pp. 32-35.
6. Mynatt, E.D. and Weber, G., "Nonvisual Presentation of Graphical User Interfaces: Contrasting Two Approaches," in the Proceedings of the 1994 ACM Conference on Human Factors in Computing Systems (CHI'94)

7. Newell, A. F., (1993) Extra-Ordinary Human-Computer Interaction, in Edwards, A. D. N (editor), Extra- Ordinary Human-Computer Interaction, New York: Cambridge University Press (in press).
8. Newell, A. F., Arnott, J. L., Cairns, A. Y., Ricketts, I. W. and Gregor, P. (1993) Intelligent systems for speech and language impaired people: A portfolio of research, in Edwards, A. D. N (editor), Extra- Ordinary Human-Computer Interaction, New York: Cambridge University Press (in press).
9. Sacks, O. (1989) Seeing Voices, A Journey into the World of the Deaf, University of California Press, Berkeley.
10. Arons, Barry. *Tools for Building Asynchronous Servers to Support Speech and Audio Applications*, 5th Annual Symposium on User Interface Software and Technology, 71-78. ACM Press, 1992.

11. Blattner, M. M., Glinert, E.P., Jorge, J.A., and Ormsby, G.R., Metawidgets: Towards a theory of multimodal interface design. Proceedings: COMPASAC 92, pp 115-120, IEEE Press, 1992.

12. Brown, C. *Computer Access in Higher Education for Students with Disabilities*, 2nd Edition. George Lithograph Company, San Francisco. 1989.

13. Brown, C. Assistive Technology Computers and Persons with Disabilities, *Communications of the ACM*, pp 36-45, 35(5), 1992.

14. http://www.nch.com.au/index.html