Department of
Computer Science

Human-Compter Interface Specifications for People with Blindness

Jeffrey Glenn Snyder

Technical Report CS-94-112

May 11, 1994

Colorado State University

HUMAN-COMPUTER INTERFACE SPECIFICATIONS FOR PEOPLE WITH BLINDNESS

JEFFREY GLENN SNYDER

COMPUTER SCIENCE DEPARTMENT

COLORADO STATE UNIVERSITY

FORT COLLINS, CO 80523

ABSTRACT

Human-computer interfaces increasingly rely on vision as graphical user interfaces are becoming more prevalent. Unfortunately, as this trend continues people with blindness face an increasing number of obstacles in gaining the same kind of high-powered information access and computing that sighted users are coming to enjoy. Many attempts to provide non-visual interfaces use unsuitable techniques or leave considerable gaps in the support the interface offers the user with blindness.

Users with blindness should have the same computational and information access opportunities as sighted users. Only with such opportunities can users with blindness be fully integrated into an information age society. The development of software tools that allow efficient information access and full use of computers is therefore crucial.

This paper attempts to create an increased awareness of the interface needs of computer users with blindness. Several considerations are presented that a non-visual interface designer should address in the development of interfaces for users with blindness. Solutions to many obstacles encountered in the development of such interfaces are offered. Finally, this paper consolidates many of the findings of research for information presentation in the area of non-visual human-computer interfacing.

1. INTRODUCTION

The evolution of computer technology in the past decade has brought increased opportunities for the integration of people with blindness into an information age society (Scadden, 1984). The development of the skills necessary for a person with blindness to function to his or her fullest potentials in such a society has, in the past, been impeded by the limited availability of textbooks and educational materials usable by the person with blindness (Lazarro, 1990). Similar obstacles had to be overcome in the transcription of information generated by individuals with blindness into media accessible by sighted users (Hatlen and Curry, 1987; Croft, 1985). Though people with blindness were able to read and write in Braille, the typical sighted person was not able to make use of Braille coded information. In other words, not only was access to information limited, but when access was available, communication between people literate in Braille and people without Braille skills presented a problem. The solution to all these hindrances has come in the form of modern computers, especially in the area of information access, where it became possible to provide electronic text easily retrieved and processed by computer users who are blind (Croft, 1985). Not all of these opportunities have been fully exploited, however, for there are few systems that allow persons with blindness to reach the highest level of productivity.

Full integration of people with blindness into this society means not only providing access to computers, but also developing tools equivalent in power and efficiency to those available to sighted users. The required tools must be truly interactive, provide immediate representation of the manipulated objects, and make any changes to manipulated objects apparent at once (Schmidt-Lademann and Dürre, 1984; Dürre, 1986). Furthermore, integration should be enhanced by providing a means of textual information exchange between users with sight and users without sight. The definition of information exchange should not merely be inferred as a transfer of data, but as an environment that provides simultaneous collaboration between both sighted users and users with blindness, without sighted users having to know Braille or any other special means of non-sighted communication (Schmidt-Lademann and Dürre, 1984; Dürre and Dürre, 1986; Dürre and Glander, 1991). Finally, improving integration means not only using the computer as a valuable tool, but as an active aid to support the user at every step of an operation and in the various activities of day-to-day life.

To design a system with the goal of achieving full integration, it is necessary to understand the specific needs of the user. For non-visual interface (NVI) users, the absence of sight means a lack of the most powerful mode of data acquisition (Dürre). While this may seem a very superficial statement, it encompasses many subtle aspects the visual interface (VI) user may not be aware of. The eye provides a way of simultaneously taking in far-ranging spatial images and perceiving many pieces of information about our world, even though our focus may only be directed at a few pieces at any given time. There is no other human sense about which this claim can be made. The user without sight has to rely on two primary methods of data acquisition: Audition and Haptics. Audition is the attainment of information through the sense of hearing. Haptics is the perception provided by the sense of touch and orientation of one's limbs to oneself. The main drawback to audition is its temporal nature, which makes conveyance of spatial relationships difficult, if not impossible (Bliss, 1980). It also provides a narrow channel for the user, in that the user can focus on very few auditory items before loss of information occurs. One need only examine the age-old courtesy of allowing just one orator at a time to speak to realize that information reaching the ear can only be selected from very few sources at any instance. Haptics can make up for the lack of dimensionality in audition, but only in a limited way. All spatial relationships beyond the touching range of arms and hands (and to a certain degree extensions of the hand, such as a cane) are literally out of reach. The usable area for sensory information acquisition is considerably curtailed compared to the area that is explorable by sight.

One may also note that in the end it is only the fingers of an individual with blindness that reveal the fine-grain details of any object, which further limits the sensing of relevant information from the individual's environment.

These limitations result in a more complex mental model of the individual's environment since the various states of the environment, and the activities needed to perform tasks in that environment, are more difficult to perceive. A walk across a room, for instance, is not merely the homing in on the target location and real-time avoidance of obstacles (as with the sighted user), but requires the computation of the target's probable spatial relationship to one's current position and the subsequent mapping of a route that will lead to the target location without encountering known barriers. The result is the need by the person with blindness for more "operators" in the achievement of any task, that is, additional steps in building mental images and added processes in creating mental mappings. The net effect is an overall increase in complexity to any task. The NVI designer has to be aware that seemingly simple tasks to a sighted user may require considerably more effort on the part of the user who is blind. Therefore, it is important to remove any user operations that do not immediately lead to the goal that he/she is striving for.

The focus of this paper will be on the software and hardware tools available to blind users, since the tools determine the constraints of the NVIs, and those aspects of Graphical User Interfaces (GUIs) that are difficult or inadvisable to create as NVI equivalents. While text-based interfaces do require some changes to better address the user's needs, they are already in a form that generally accommodates the NVI user and thus do not call for complex transformations (Edwards, 1989 {2}). The increasing prevalence of GUIs, on the other hand, will make access increasingly burdensome for NVI users as GUIs capitalize on qualities provided through vision (Edwards, 1989 {2}; Scadden, 1984; Ternlund, 1991). This paper will attempt to offer solutions to the problems encountered in making such access available and suggest some techniques for re-creating the power and efficiency of GUIs in an NVI environment.

2. PHYSICAL INTERFACE OPTIONS FOR PERSONS WITH BLINDNESS

In today's design of interfaces for VI users, many hardware features through which the user performs file and data manipulations, such as the standard keyboard and monitor, are taken for granted. Even the mouse has become so commonplace that many user environments are tailored toward its use. Would a current day GUI designer even consider gearing an interface towards punch-card input or suffering through hardcopy-only output? Yet some of the tools available to the NVI user show similar traits of user unfriendliness as the above input/output (I/O) devices, albeit not to that extreme.

This chapter is devoted to the description of the I/O devices commonly available to NVI users, since any interface designer who wishes to accommodate access for users with blindness must consider the limitations and requirements of these devices. There are two forms of interfacing possible, one involving "manual" manipulation and the sense of touch, the other making use of voice and hearing.

2.1. Input devices

Input devices fall into two categories: tactual and voice input. "Tactual" refers to anything pertaining to the sense of touch. In the context of computer interface devices the meaning is narrowed down to any activity that requires the user to explore the I/O interface equipment with the fingers. Tactual input devices are all, except for one, borrowed from "the visual world."

2.1.1 Tactual input devices

The most accessible device is the standard Qwerty keyboard. As with sighted users, any user with blindness who is able to type will be able to make use of the keyboard as a primary input device. Users who cannot or are unable to type will have to make use of voice-input devices, potentially combined with some of the pointing devices mentioned later.

The only difficulty and slow-down for the NVI user in operating the Qwerty keyboard is created through the lack of visual feedback while the user places his/her hands on the board. More time is needed for further tactual checks on the keyboard or for more frequent feedback from the output, since the user cannot affirm correct positioning of the hands through a simple glance.

The main alternative to the Qwerty keyboard is found in the Braille keyboard (Lazarro, 1990). In its simplest form, it consists of 9 keys: four for each hand and the space bar. As with the Qwerty keyboard, the thumb's sole function lies in depressing the space bar. Each letter or character is formed by depressing a combination of the keys. For example, depressing keys one and six will yield the character '1', while the combination of keys one, two and six translates into '2'. Though this may seem difficult to learn in the eyes of a sighted user, experience has shown that Braille keyboard users can do as well as or better than Qwerty keyboard users. One only needs to consider that each key on the Braille board corresponds to a dot in Braille code to understand the advantage of this keyboard over the "key-to-character" type keyboards. Braille key combinations can directly be translated into Braille dot configurations (Lazarro, 1990). Consequently there is no time wasted in reaching for distinct keys as with the Qwerty keyboard, since the fingers never leave the assigned keys. A further advantage is that the user will find it next to impossible to put the fingers in a wrong position: either each finger is touching a key, or it is not (Schmidt-Lademann and Dürre, 1984). The main drawback of the Braille keyboard is found in its limited availability. It is highly unlikely for a NVI user to find a Braille board unless he/she has made prior arrangements or is carrying one along to his/her destination. The limited customer base for Braille keyboards also makes them quite expensive (Bowe, 1987).

A hybrid of readily available keyboard hardware crossed with Braille Board characteristics can be found in Braille keyboard emulation. This technique assigns the functions of the Braille keys to selected keys on a Qwerty keyboard. The home-row for the Qwerty keyboard, for instance, may substitute for the eight keys of the Braille keyboard. In this context they no longer carry the original character values, but their corresponding dot codes instead. The user does not have to abandon the preferred way of information input in situations where the Braille keyboard is not available. The only sacrifice to the user is the decreased security in finger placement.

In considering non-textual input devices, it is important to understand that the person with blindness navigates with an "egocentric frame of reference" rather than "a geographic frame" (Easton and Bentzen, 1987). Unlike a sighted user, who may map out his/her route of travel by envisioning a path from point A to point B, the individual with blindness prefers to envision that path as a movement from his/her current position to the final destination referencing many smaller sub-points. Thus, planning a trip to the favorite grocery store is not a matter of "I will walk to the store at avenue A and boulevard B" but instead of "I will move in this direction until I hit this landmark and then make a 90 degree turn and proceed until I arrive at the store." Such referencing directly influences the benefits and drawbacks of the various navigational input devices since only those with an inherent "null" position will provide the most direct correspondence to the "self-referenced" frame (Edwards, 1989 {2}). Common "null" point devices are the joystick and the arrow keys.

The joy stick, the primary "null" point device, "emulates" in its inactive state the cursor at rest. Upon deflection of the stick the cursor is moved in the direction of the deflection, the speed controlled by the degree of deflection. A joystick provides the blind user with the familiar sense of "move-from-here" action. The advantage of a control displaying this type of behavior can be seen in the many applications involving man-vehicle interactions in which stick-like devices are used. Pilots of combat planes, for instance, use joy-stick related controls which provide a better sense of direction in each control-activated movement than is true for other types of steering devices. As an added advantage, the concept of movement associated with the joystick provides a natural binding of the cursor to a large scale grid, such as a "line and column" raster, which in most cases will approach the highest resolution an NVI user can take advantage of.

The arrow keys may also be classified as a "null" point instrument. This option should be almost universally available, since the arrow keys are part of practically any Qwerty keyboard. The biggest obstacle in their use may be the non-standardized placement, requiring users with blindness to spend more time familiarizing themselves with the particular keyboard layout whenever they encounter different keyboards. As with the Qwerty keyboard, there is also the higher risk of wrong placement of the fingers, aggravated by the variable layout between different keyboards. Arrow keys, additionally, suffer from a lack of speed control afforded by the joystick. When the keys are pressed, the cursor keeps moving at the same speed, making the keys less friendly for browsing through or skipping of text than the variable-speed joystick.

The mouse and its counterpart the track ball are, at best, substitutes for the joystick (Edwards, 1989 {2}; Bowe, 1987). Mouse and track ball require a high degree of visual feedback and cannot give the NVI user the secure frame of reference the joystick provides, since their principle is based on "drawing the path" for the cursor. The grid of operation for the mouse is essentially pixel-sized. In normal situations such a grid of operation is too fine-grained to be of use to the person who is blind. Unfortunately, mice are too prevalent for the NVI user to completely avoid them. The partial solution is to simulate a joystick by confining the mouse movements to a large scale raster. The movement of the mouse can then emulate the action of a joystick.

A final option for navigational input is found with Outspoken which is an auditory user interface for people with blindness (Edwards, 1991). Outspoken assigns GUI control functions to the keypad, sacrificing conventional keypad number input and manipulation (Figure 1). The redesigned pad includes four keys for vertical and horizontal movement, one for selecting items and another to control "drag and release" operations in the GUI. Two more keys are reserved for special functions: the menu key places the cursor in the menu bar, while the window key brings up a special menu listing all open windows. As with the arrow keys, universal availability of this navigational mechanism is ensured, although there is a greater propensity for incorrect finger placement. A further drawback lies in the need for the user to memorize the key function layout, which at least initially might affect productivity.

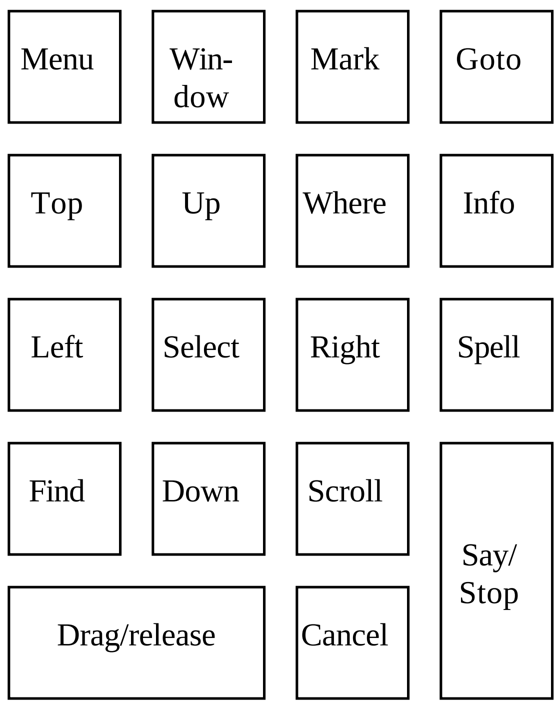

Figure 1: Layout of Outspoken's redefined keypad (Edwards, 1991)

2.1.2 Voice input

Voice input, though widely available, is still in its infancy, making it practical only as a replacement for textual command entry. At the current rate of development, however, this may not hold true for much longer. There are quite a few barriers, as compared to tactual input, that must be taken into account when using voice medium. First, for the system to clearly recognize spoken input the user must possess relatively good pronunciation, pitch and other speech qualities (Normand, 1991). Foreigners or users with speech impairments may find themselves at a disadvantage if any of their speech characteristics do not fall within the range expected by the voice recognition system. Secondly, the systems have to be retrained with each new user, creating more of a time overhead (Bowe, 1987). Finally there is a psychological factor involved in that many users express inhibitions to interacting vocally with a machine (Normand, 1991). A common example of these inhibitions is found in the reluctance of many people to speak to an answering machine. Such inhibitions may be aggravated by the presence of other people, especially if the operations are to be kept confidential. For example, most users would probably prefer not to verbally initiate a change in a password while in the company of fellow employees, or if there is a possibility of being overheard by some unknown listener. Voice input also poses the problem of lack of guidance with respect to syntax and vocabulary (Normand, 1991). Discovering the specifics of a query format may be tedious, especially for NVI users, who cannot fall back onto visual help. For instance, there could be a multitude of possible commands for unmaking the last voice entry: undo, unmake, erase, delete, and so on. Then there could be combinations involving the former commands: undo last, erase previous, etc. Another problem is noisy environments, where some of the background noise can be picked up and turned into erroneous input, or mask the input entirely.

While there are problems with the use of voice as input, there are some indications that voice commands can be more efficient than the corresponding keyboard entries. One study found that an overall speed-up of between five and ten percent can be achieved when hot key input is replaced with its voice counterpart (Pausch and Leatherby, 1991). (Hot keys are also commonly referred to as keyboard macros). Reasons for the acceleration are (1) cognitive, in that the user does not have to perform a cognitive context switch for voicing the command, as takes place with the use of hot keys; and (2) physical, in that the user does not have to spend time homing in on the hot key and then returning to the original position. The same should hold true for any navigational or pointing commands since they too involve the mental translation and physical actions necessary to activate the appropriate keys. In text-based user environments, such as DOS, the advantage might be greater yet since each operation there consists of an entire sequence of key entries. In summary, voice commands may not be the desirable medium for any control sequences requiring strings of terms but may be a suitable replacement for entering short commands.

2.1.3 Tactual input devices versus voice input

Currently only tactual input provides the flexibility and accuracy needed to operate a software system effectively. As the population becomes more computer literate, tactual input should become less of an obstacle to people with non-congenital blindness. Voice input is still too limited to serve as a complete substitute for tactual input. While voice input may function well as a command, navigational and hot key supplement to tactual input, voice input by itself is too rudimentary in its word recognition capabilities and too prone to errors. Tentative studies also seem to indicate that the incidence of repetitive motion injury is higher with voice input than with other means of input. (While repetitive motion injury is usually associated with the arms and hands, it can apply to any part of the body, as in this case, the larynx).

2.2. Output devices

Output devices may be grouped into three categories: Tactual devices, tactile devices and speech output. Tactual and tactile devices are similar in nature and might also be seen as subgroups of the same category.

The definition of tactual, with respect to output, incorporates the understanding that the displayed information is of a static nature, much like Braille in textbooks for people with blindness. The user has to run his or her fingers over the "touch" display to gain a mental image of the information provided, not unlike a sighted user visually scanning a text to unlock its meaning. One can experience the process by feeling some of the Braille codes provided on many elevator control panels, on room number signs and on the panels of some ATM machines.

"Tactile" adds the dimension of animation to the qualities of a tactual output device. Imagine an elevator Braille code "plate" which is capable of raising and lowering the "little bumps". Now the "plate" would have the ability to represent various other characters. To represent a word or a sentence, the "plate" would only have to supply all characters consecutively in that word or sentence, without the user having to move his or her finger from the plate. The same principle applies to tactile output devices. They create the sensation of movement beneath the user's fingers, eliminating the need for active exploration on the part of the user. Visual examples can be found in many of the display boards in football stadiums and elsewhere that use arrays of individual lights to form letters. By "moving" the letters across the board, the viewer experiences a sense of scanning the displayed text. Though the display permits only a few characters at a time, one could envision reading entire books in such a fashion. The definition of tactile also limits its use to output devices, since autonomous movement in any input devices would most likely not be welcome by any user.

2.2.1 Common features of tactual/tactile devices

Unlike tactual input devices, tactual/tactile output devices bear no resemblance to their visual counterparts. All are based on mechanical stimulation of the user's sensory perception in the finger tips. Currently, the only way to accomplish this is through the use of arrays of little pins, which are lifted and lowered to form larger patterns (Lazarro, 1990). The idea behind these devices is similar to what forms images on monochrome monitors. The pin is replaced by a pixel, and the equivalents to "lowered" and "lifted" are "dark" and "illuminated," respectively. The analogy ends at the granularity. Arrays of pixels are measured in the thousands; tactile arrays are, at best, in the tens. The relative coarseness of tactual/tactile output devices stems in part from physical limitations in trying to arrange large arrays of mechanically activated pins in a way that is still affordable to the user, but also from the limitations in "fine-feel" that the human sense of touch can provide for information intake. Though it is possible to sense even minute differences in surface texture, for a person to sense the distinct features that make up the texture, the features have to be sufficiently large, somewhere in the measure of millimeters. The closest visual example might be found in the aforementioned light array billboards found at many football stadiums.

The most profound consequence of touch-only output is the loss of the "third dimension". GUIs make use of this three dimensional effect in abundance through visual metaphors. We recognize folders in these interfaces because they look somewhat like the "real thing". The Macintosh trash can is easily recognized, because it has the familiar garbage can outline. Even the desktop is named as such, because stacking up windows is vaguely similar to stashing various piles of documents on the desk. Even though the windows are displayed on a two dimensional surface, the visual image allows for a limited added dimension by creating a "real-world" visual mock-up. This is not true for sensory output devices. Using the example of the light-array billboard, creating an acceptably user-friendly GUI with simple on-off imaging is almost impossible when such imaging is combined with very low resolution, which is the nature of tactual/tactile output. Furthermore, the sense of touch cannot find the equivalent sensation of a "portrayed" third dimension, when it is in contact with a flat surface, such as a sensory output device. The Macintosh trash can, presented on a tactile output device as the equivalent formation of lines does not "feel" anything like the pictorial object it represents. It merely comes across as a "bunch of lines". Neither does the outline of a folder translate tactually into the real-world item.

Taking this into account, it is impossible to represent the three dimensionality and imagery of GUIs as a direct tactual/tactile translation.

A further consideration in the use of tactual/tactile output devices is the relatively narrow information intake. Six pins is the minimum number needed to represent any standard character, excluding "special" features, such as capitalization. (While it is possible to represent capitalization through two-character codes, each character in the code is still a separate entity). Six pins also approaches the maximum number of pins a user with blindness can effectively use in distinguishing separate pieces of information. Compared to visual information intake, it means a considerable reduction of the possible options in the display of information.

2.2.2 Tactual output devices

The Braille display is the primary example of a tactual output device. In its basic form, it consists of a row of 20 cells, each cell providing six or eight Braille pins for displaying static configurations of Braille text. Incidentally, one may note that the Braille code used for computers is not the same as the conventional Braille code. Although it is possible to use a conventional Braille code in computing, it does not offer the context-free mapping of computer Braille to ASCII or other widely accepted character codes. Many words in conventional Braille, for instance, may be expressed as single symbols, picking up their final meaning from the surrounding text. This results in an additional level of translation between the user interface and any applications, requiring further resources not needed with computer Braille.

With six points for each Braille display cell it is possible to encode 26 letters, 10 digits and 20 signs. To expand the repertoire it is necessary to add a seventh pin, which then permits representation of capitalization and other more specialized textual features. The same can be accomplished through vibration in the pins. This is not the equivalent to a tactile display, since the formation of the characters takes place through the elevation of the pins, not their movement.

The user has to actively scan the constellation of pins to derive any meaning. Many displays also offer eight pins and vibration to vastly expand representation of characters and, at the same time, offer a means of pointing out special textual features, such as bold, italic, etc.

It might seem that a display of 20 sets of six pins (in the most basic display) may be more limiting than a larger array of pins, but one needs to remember that in either case the user is still limited to the sensory assimilation of information. The small display provides a nice means of defining the current work area by binding the display window to the software cursor (Schmidt-Lademann and Dürre, 1984; Dürre and Glander, 1991). In other words, the cursor is always contained within the 20 character display, usually, but not necessarily, in the center thereof. When the computer "jumps" the cursor to a new position, such as in a word search, the user is automatically lead along. The user does not have to spend time finding the new working position by searching for the cursor, as would be the case in a larger display. In addition, the computer automatically knows where the user is pointing without the user having to perform a special point operation. This also fulfills the "see and point" (or in this context "feel and point") guideline for the Apple Desktop Interface (Apple, 1987). The user senses the item of interest and simultaneously makes the item's position known to the computer. Along the same line, "feel and point" addresses the egocentric frame of reference mentioned before. The user does not have to worry about the path to the cursor; the cursor acts as the anchor point from which to initiate further actions. Unfortunately, the main drawback to a computer Braille based system lies in the currently low Braille proficiency rate of only ten percent for potential users. It severely limits the audience to which a Braille-based system may appeal. One needs to remember, however, that most users with congenital blindness, blindness dating from birth, will be familiar with Braille and that this system is primarily targeted at this group of users. The same is true for the following example of a tactile device, the Optacon, though to a lesser extent.

2.2.3 Tactile output devices

The Optacon was originally designed without the computer in mind, as its full name "Optical-to-Tactile Converter" may suggest. It was designed to make written material available in tactile form. With the rise of the computer, the Optacon attained a new purpose and is now the primary means of displaying computer output. The earliest Optacon provided a pad of 144 tactile pins, assembled in an array of 24 rows and 6 columns (the new Optacon combines 20 rows with 5 pins each). The Optacon was connected to a camera through which text or diagrams were scanned. The "black" portion of the scanned images "showed" as patterns of vibrating pins on the pad. A character of written text essentially showed as a vibrational outline on the pad. Once hooked up to a computer, however, the Optacon's role becomes much more variable (Dürre et al., 1984). Now the display of information in Braille as well as plain text, with the convenience of variable sizes, is possible. Graphical output can also be displayed in its rudimentary form, with the added ability to zoom-in on, and out of, any section of the displayed graphics.

The Optacon is placed in the tactile category because the vibrational pattern defines the output, and not inactive states of lowered and raised pins; the user observes the stimulus provided and does not need to actively feel for the differences in pin information. Although the display is larger now, the Optacon does not necessarily provide an advantage over the Braille display. Even though the total number of pins may be roughly equivalent among both displays, the Braille display uses subdivisions of six to eight pins as the smallest units of information (in text applications). The user does not have to search each pin, only each subdivision. This is not true for the Optacon, where each pin may be considered a separate data-carrying unit. Furthermore, the larger units of the Braille display are arranged in one row, alleviating a search in two dimensions, as is necessary with the Optacon.

A further drawback to the Optacon lies in the layout of most textual material. Standard text is provided in rows of words. The Optacon, conversely, displays text in a column-like fashion. Due to the Optacon's layout, parts of several sentences may be visible, but only small segments of each sentence. The Braille display closely conforms to the textual structure in that it accommodates a comparatively long segment of a visible sentence, while no other sentence fragments are shown.

A positive feature of the Optacon is the option to display text in a form other than Braille, giving some other means of tactually/tactilely showing information. This might especially appeal to an audience with non-congenital blindness, and thus familiar with traditional character shapes. The Optacon may also offer the best way to explore graphical information since the larger display will provide a fairly coherent picture for non-textual data. A user proficient in Braille may desire both displays, one for textual, the other for graphical output.

2.2.4 Speech output

Speech output may seem quite appealing when taking into account that no special skills are needed to comprehend it. Speech output can serve as a colorful means of communication through its variability of background tone, timbre, volume and other characteristics. All this, however, cannot overcome its main drawback found in its temporal nature. Speech is an excellent means of providing a continuous flow of related information, but it serves poorly when having to retrace parts or scrutinize sections of the information. Retracing with speech output is similar to the problems one might have listening to a book on tape versus reading the same book. As long as the audience is content with the flow of the spoken output, the tape might well be preferable to the book. However, as soon as a reference back to a particular spot becomes necessary, there is no means of statically displaying the place of interest for further examination (Schmidt-Lademann and Dürre, 1984; Croft, 1985; Pitt and Edwards, 1989). The only choice left is to continuously replay the section, without a reduction in speed. To carry this example further, if the book has a diagram, the only option with tape output is description of the diagram which is a poor patch for visually (or tactually) exploring the same diagram.

Applying this example to user interfaces one may note that any speech representation will include a large portion of description of the interface, more so for interfaces that merely try to represent the existing GUIs without providing further functionalities for the NVI user. Description in itself might still be acceptable except that it puts a large burden on the user's memory. Any relevant spoken information has to be memorized for subsequent information to make sense. Consequently, the more description that takes place in the output, the greater the load that is put on the user's memory; a load that may be irrelevant to the final objective of the user. This was demonstrated in a project by Boyd et al., 1990, who initially set out to construct a speech-only based system. In the last of the three project stages, the authors stated the following: "The resurrection of the puck/mouse to provide screen information through a channel other than speech is a central ingredient in Stage 3 efforts to increase the blind person's ability to interpret complex graphical images on the graphical user interface. It also restores more of the functionalities of scanning, browsing, and memory jogging ..." (Boyd et al., 1990, p. 501). They further noted that the "... problem with using speech-based solutions only is that it is difficult to keep track continuously of where things are on the screen" (Boyd et al., 1990, p. 501).

This statement is further supported by the algorithm used to locate an object on an auditory GUI, as presented by Edwards, 1991. This path follows four steps: choosing the target, planning the route, moving to the target and clicking the mouse (Figure 2). In the first step, selecting the target, the cognitive activities for both VI and NVI users should be similar. The user first needs to identify the next task (for instance, to copy a file) and then recall the method to achieve that task (dragging the file icon to a disk drive icon). The latter could be seen as the first procedural activity requiring the involvement of memory. Next the user has to decide on the first sub-task of the method chosen (click and drag the file icon) and finally has to recall which object corresponds to the desired method (the icon of the file to be copied and the disk drive icon), which again necessitates the use of memory in both visual and auditory GUIs.

1. Choose Target
2. Plan the Route
3. Move to Target
4. Click Mouse

Figure 2: Four steps to selecting a GUI object

Visual and auditory implementations should differ substantially in the next step. To plan the route, the user with blindness has to recall the position of the chosen target (the location of the icon for the file to be copied) and then remember the screen layout (the location of the disk icon in relation to the file icon). Both of these activities rely completely on memory mappings of the layout of the auditory screen, putting the burden entirely on the user. In the tactual/tactile equivalent such complete memorization is either limited or completely unnecessary. Obviously a sensory "full-screen" layout will provide as close to an explicit map of the screen as is possible non-visually. Even a very small display, such as the Braille display, will permit the user to perform tactual sweeps of "concrete" tactual objects, eliminating some of the abstract operations needed to maintain a mental picture of the intangible auditory layout.

The final two steps should not provide additional strain on the user's memory in either the visual or auditory implementations. Moving to the target is a matter of translating auditory or visual signals into meaningful information, requiring no memory operations. The last step, clicking the mouse, needs no further elaboration.

In a paper by Edwards, 1989 {1}, the author states that the main problem observed in the use of auditory GUIs lies in the long time taken to accomplish various tasks, substantiating the notion that sound-based systems rely heavily on the user for effective operation.

There are two more findings on the drawbacks of auditory systems. Both were derived in the study on route information for travelers who are blind, mentioned earlier (Easton and Bentzen, 1987). One is that verbal directions are transformed into spatial images. Obviously such a translation does not have to take place in tactual/tactile systems, since the nature of the information display is already spatial. This is further supported by the other finding presented in the study, that tactile maps may be of better value than verbal information. In essence, the GUI can be seen as a map to the user's computer work environment, a map that may serve the user with blindness better when in its tactile form.

A final limitation of speech output is found in the restricted method of conveying textual features, such as highlighting through different fonts, indentations, etc. These have to be either announced explicitly or indicated through background tones or speech characteristics. The first disrupts the reading of the actual text, the latter limits the ways in which special features are presented. Tactual/tactile displays on the other hand can emulate special features with the implicitness of visual output devices and can provide ways in which tactual/tactile reading is not disrupted by explicit text feature messages. Since text is usually presented in a spatial arrangement, the spatial qualities of text are easier preserved in tactual/tactile displays than in the temporal environment of speech.

The temporal side of speech does have uses in contexts where immediate and brief messages have to be conveyed to the user, such as with warnings or system initiated signals (Bliss, 1980). Messages such as "Are you sure?" or "File not found" are immediately noticed without the user having to read the information off the primary output device. Speech may also be used to present texts in which the structure is not important and lingering and retracing are not required, as for instance in pleasure reading (Dürre; Dürre and Glander, 1991). Lastly, pitch information was found to be of minimal value to most users (Edwards, 1989 {1,2}).

Prime examples of GUI-based speech-output systems are Outspoken, an icon translation system based on the Macintosh (Edwards, 1991) and Soundtrack, a menu-based editor (Griffith, 1990).

2.2.5 Tactual/tactile output devices versus speech output

Tactual/tactile output is the only output that can preserve some characteristics of visual output. Tactual/tactile output can convey the layout of the presented information and allows for static display of the output. Tactual/tactile output, unlike speech output, places a minimal burden on the user's memory, creating an environment that is more goal-oriented than procedure-oriented (That is, the user may focus on the task at hand rather than the memorization of the details needed to support the achievement of the task).

In conclusion, speech has some merit as a supplemental channel to any tactual/tactile system, but due to its temporal nature it simply cannot convey all textual information and spatial relationships when used as the only means of output (Dürre and Glander, 1991). For some individuals it may be the only viable option. Nevertheless, if possible, tactual/tactile output should be considered first.

2.3 Complete Systems Comparisons

The next three sections will analyze some of the effects of combining the various input and output devices. The first section will cover the more "traditional" approach a designer might take; traditional in the sense that the I/O devices resemble the standard keyboard and computer screen in a non-visual environment. The main audience might be people who are well versed in Braille and comfortable with typing. The second section observes the effect of an "innovative" approach in which emphasis is put on abilities that most of the population will have gained as a matter of course: speech and hearing. This approach might target people who have lost their sense of sight late in life and are not able to learn tactual/tactile input/output. The last section, finally, will consider combinations of tactual/tactile and voice/speech input/output.

2.3.1 Tactual/tactile-only input/output systems

To this point, tactual/tactile input and output have been considered separately. The question that arises now is how well the two work in combination. The main limitation in combining the two sense-of-touch mechanisms is the restricted opportunity for simultaneous reading and writing. Obviously, when both hands are tied up entering data they cannot be used at the same time to read a Braille display. Reading, on the other hand, requires the use of at least one hand, limiting textual input but allowing for simultaneous navigation through non-textual input devices. Thus reading and writing become mostly sequential activities, requiring more time for establishing feedback than is the case for sighted users. Detection of mistakes in typed entries may not be immediate, since the user has to leave the keyboard to reconfirm the entered data. Wrong placement of the fingers, or operating in the incorrect mode (that is, command mode versus text entry mode), may create problems more severe for an NVI user than a VI user would experience.

A final consideration in the use of touch-only devices is the target customer. While individuals with blindness may have all the facilities to fully use tactual and tactile input/output devices, there is also a segment of users that may not. People who lost their vision to diabetes, for instance, may experience other side-effects of the disease, such as loss of sensitivity or dexterity (Edwards, 1989 {2}). In such cases the designer might opt for voice and speech instead.

2.3.2. Voice medium-only systems

As has been discussed, speech input and voice output are already quite limiting as separate entities. When combined to form the entire user interface, those limitations are preserved, offering a very narrow input/output domain. Additionally, while tactual/tactile input and output provide limited concurrency, auditory-based systems permit none. For example, one may consider the common occurrence of two people trying to initiate a conversation at the same time, only for both to come to an abrupt halt. This is even more clearly encountered in communications by phone. Speech/voice command systems may be the only alternative for some users, but any user with a degree of tactual ability may well benefit from acquiring the skills necessary to work with tactual/tactile input/output, opening a wider array of I/O options and the potential for more effective interface use.

2.3.3. Tactual/tactile and speech/voice command combinations

Combining voice input/speech output with tactual/tactile input and output permits enhancements in both media (Pitt and Edwards, 1989). Taking into account the restraints of the media, a limited expansion of input/output concurrency can be achieved. Specifically, combining voice output with tactual/tactile input would allow the system to generate error messages and warnings even while the user's hands are off the Braille display, providing greater security while navigating the operating system or entering text (Bliss, 1980). A similar technique could be used to teach Braille, not unlike computer-aided typing courses for VI users. Each Braille character could be announced as the NVI user enters it, or entire words and sentences could be spoken while the user is practicing Braille typing. Another possibility is the announcement of text attributes in the tactual output, providing uninterrupted reading with full awareness of any textual features (Vanderheiden, 1989). The advantage of this combination may depend on the user, because the simultaneous tactual and auditory flow of information might prove confusing.

Another suggestion, presented by Vanderheiden, 1989, would attach speech output to the mouse cursor. His paper does not address the problems that might occur in determining how much text is spoken at any one point and how to handle fast mouse movements. Moreover, in instances where the user wishes to cross large segments of text, would the system output large amounts of incoherent speech as the user sweeps across the text, or would the system be silenced, requiring another means of conveying information (Pitt and Edwards, 1989)?

Oral commands as input to the system would enable the user to navigate with both hands on the tactile display improving the tactile exploration of graphical data. Furthermore, voice input gives the user additional tools for special features, such as highlighting text. Instead of searching for the desired text attribute in a table, the user could simply announce the appropriate feature, such as bold, underline, etc.

Perhaps the most significant factor in dealing with the tradeoffs between the various media is flexibility. Permitting a wide array of configurations will allow the user to customize the system to his or her individual preferences and abilities and, additionally, provide a higher range of adaptability to various environments and hardware systems (Bowe, 1987).

3. CONSIDERATIONS ARISING FROM THE INTERFACING LIMITATIONS

Having surveyed the input and output constraints the NVI user has to contend with, this section will focus on the considerations an interface designer may encounter in developing a system for individuals who are blind. Design factors can be grouped into three categories: general, applicable to all aspects of the system; text related, essentially those aspects concerned with the editing and manipulation of text files; and non-textual, aspects which involve issues of systems and files management with an emphasis on adaptations of GUIs to the environment of the NVI user.

3.1. General considerations

As seen earlier, the NVI user is at a disadvantage, compared to the sighted user, in dealing with GUIs in that all manipulations a sighted user performs in a visual environment cannot be made available to the NVI user in a form that provides the same high level of information conveyance. Consequently, all operations that are inherently visual, such as text editing or even operating system navigation, require the user with blindness to compensate with more involved mental mappings of the environment. Since this creates a heavier mental burden on the user, as much of that burden as possible should be alleviated through the simplification of the environment. In essence, this translates into giving as many memory aids to the NVI user as can be built into the system without creating further stumbling blocks. Such aids can range from displaying mnemonics for the various possible commands available at the current state of the system to creating system-generated reminders for user action.

A second consideration in the design of an NVI is the unavailability of an overall impression of the environment in which the user with blindness is operating. While a sighted user will always have a general notion of his or her position in the GUI environment, the spreadsheet or text editor, the same overall feel is not necessarily experienced by the NVI user, who cannot rely on the visual frame of reference provided by the window border, spreadsheet grid or scrollbar. In an NVI derived from simple translation of the visual foundation, that is, a system that does not provide any NVI specific aides, but merely attempts to present the visual GUI as an unaltered tactual/tactile or auditory equivalent, the user has to establish his or her position by scanning for the borders. Such scanning can become quite extensive in applications that allow for virtual surfaces viewed in the window (such as a spreadsheet or text file) to extend well beyond the window frame. To eliminate this problem it is essential to provide the user with functions detailing the exact locational information within an application work area. The user should be able to call up absolute references (e.g., line x and column y in a text file) as well as relative ones (e.g., line x, column y on page z of the same text file). Ensuring easy access to locational information eliminates time-consuming and tedious sweeps for the purpose of establishing locational frames of references.

A further consideration is the need for an immediate display of the effect of a user's action in a software system. Such a display is an important point that has come to be expected in any GUI, yet has been ignored at times in systems for people with blindness. Any sighted user expects to instantly see the results of his or her actions in a graphical environment, be that copying files or deleting characters in a document. It would be undesirable to make a change in a text file and not experience the results at once, as was the case with line editors. Why should the "what you see is what you get" principle (Apple, 1987) not be carried over into the domain of the NVI user as "what you feel is what you get" (WYFIWYG) (Schmidt-Lademann and Dürre, 1984). After all, the NVI user will have to put in substantially more effort in acquiring the overall impression of his/her work environment than the VI user, especially considering the limited concurrency of input and output. Therefore, any changes caused by the user's actions should be displayed clearly and immediately, allowing for perception of the environment in its most up-to-date state (Schmidt-Lademann and Dürre, 1984).

Also induced by the non-concurrency of input and output is the possibility of the user mistakenly operating in one mode, while believing to be in another. The user might, for instance, enter a sequence of characters believing to be responding to a system's prompt, while he/she is unwittingly operating in command mode. The character stream may then lead to a chain of unwanted processes. Such an occurrence, while rare, may lead to considerable damage. To a sighted user this may seem to be an extremely unlikely scenario, since he or she will receive immediate feedback to his or her actions, but that does not hold true for the NVI user who may, at best, have only limited opportunity to receive output information. This is especially valid for users who rely solely on either audio or tactual/tactile media for both input and output, since neither will allow fully concurrent input/output. Thus, changes in modality (such as switching from edit to command mode) must be clearly noticeable, either by announcing the change through some type of warning sound or by forcing the user to perform a special action (such as hitting the "escape" key first), or by providing different environments for the various modes. One solution might be to have all commands spoken aloud while in command mode, whereas text editing mode is silent. Additionally, making changes in a text file could be accompanied by a clicking sound when the user is in strike-over mode rather than in insert mode. Whichever technique is used, it is essential to provide explicit modality information.

3.2. Text related considerations

Two items fall into the group of text related considerations, both primarily concerned with the reading of plain text documents. The first is the preservation of "landscape" information, while the second is the need for a separate browsing mode. The need for the preservation of landscape touches on the WYFIWYG principle presented earlier (Schmidt-Lademann and Dürre, 1984). The main concern here is to allow the NVI user to make as much use out of the layout of the text as the sighted user (Dürre and Glander, 1991). Any user will anticipate a change in the direction in the flow of a paper when coming upon an indentation or will focus on the main idea presented in a title. This added information provided by the look or "landscape" of the text should be conveyed just as much to the NVI user as it is to the VI user. In short, the virtual surface in the text editor should emulate the sheet of paper it is trying to mimic (Dürre). The preservation should not only include the usual white space in any text, but also any other text attributes, such as inverse and color highlights (Brown, 1992). Obviously the attributes may not be directly representable in a tactual/tactile/auditory form, so suitable replacements have to be found, some of which have been mentioned in previous sections. Of equal importance as the preservation of attributes is the NVI user's potential need to use those attributes when composing certain text documents. Though such qualities may be non-essential to the NVI user, to be competitive in today's graphically oriented society many users cannot afford to forego them. A text editor needs to provide functions to incorporate these features into the text.

The second text-related issue is the need for a reading mode specifically for the NVI user (Brown, 1992). Unlike the sighted user, the NVI user cannot survey a page without frequently advancing the display window since the most common tactual/tactile displays are by far too small to permit the continuous reading provided by visual output displays. Especially in the case of the Braille display, such advances may have to take place several times in a line. Thus the NVI user with a 20 character display may have to view 240 sections to read a page of 60 lines, assuming each line is of 80 character width. The VI user, on the other hand, will have to view only three sections, if he/she is using a 20 line full-width display. While scrolling the window provided by the display is possible, "jumping" the display ahead a distance less than or equal to the length of the display may be much more efficient, especially if one considers scrolling to be nothing else but window "jumps" of one character or line at a time (Schmidt-Lademann and Dürre, 1984). The page down/page up functions available in most visual editors are essentially of the same nature: these functions allow the user to advance through the text one screen-full at a time without the need to spend time tediously scrolling to the next page. Considering the frequency at which the NVI user may have to perform window advances, the reading mode becomes much more significant than even the page down/page up functions for the VI user. The text review/reading mode is usually passive, in that few or no text alteration functions are available (Brown, 1992); conversely, the text alteration and movement modes may provide only limited support for reading. Providing full text alteration functionality in text reading mode would require the use of separate movement commands for cursor movement and window read advances. The split permits the sharing of the same commands, whereby the reading mode could be seen as an "intensification" of the movement mode commands.

The reading mode should have three configurations (Figure 3). One allows advances of the window in fixed increments, showing each text segment as it is "cut" by the window, that is, the window moves without regard to where its vertical boundaries end up in the text. Words that fall on the boundaries of the window show only partially. The user should be able to adjust the increments to suit his or her preference, permitting either full-length window jumps, or fractions thereof. This configuration is especially suitable for tables with fixed column widths. It may also provide a good overview of the layout of the text.

The next two modes share the same movement technique. Each movement brings the window to the beginning of the last word that was cut by the end of the window after the previous move. Thus both configurations never show incomplete words at the beginning of the window. They differ, however, in that one may show the word cut by the end, while the other does not. Thus the first may split the last word in the display, while the second will display only complete words. In both cases the movements are completely variable. In the rare event that a word is too large for the display, the user either needs to leave the reading mode and scroll forward, or the display may advance to show the remaining portion of the word with the next movement. Under the last condition only will the displayed information begin mid-word.

Showing only complete words may ease the reading of texts, but it may not relate the layout of the text as well as the fixed length jumps do. Users may also prefer this technique in spreadsheets with variable width columns. Again, the user will have to choose the appropriate configuration. Making the NVI system flexible and easy to adjust for different needs will certainly broaden the system's appeal.

```
        This configuration is especially suitable for tables ...
                         (a) Sample Sentence

       | This configuration i|
                             |s especially suitabl|
                  (b) Fixed window advancement

       | This configuration i|
                            |is especially suitab|
       (c) Variable window advancement with cut word shown

       | This configuration  |
                            |is especially       |
  (d) Variable window advancement with partial word omitted
```

Figure 3: Three window advancement modes

3.3. Considerations for graphical information

This section is devoted to some of the problems associated with directly translating a GUI to an NVI user's environment. Designers will be tempted to make GUIs a platform for adaptations for the needs of NVI users since GUIs are abundant on the market. The difficulty they face in doing so lies with the conveyance of the visual metaphors, which form the primary basis for the display of information to GUI users (Boyd et al., 1990). (The Windows file icon and Macintosh trash can are well-known examples of metaphors, each representing computer objects through association with real-world items). The metaphors carry meaning through four properties, all of which present stumbling blocks when attempting to translate them for the NVI user.

The first property of a metaphor, location, requires the user to take the time to establish a mental survey of the entire interface, assuming that this property will also be incorporated into the interface for the NVI user. As pointed out previously, the user will have to maintain a much more complex mental mapping of the various metaphors, putting the burden of the interaction with the system on the user. Establishing this characteristic on the tactual/tactile or audio display does not benefit the user, but instead requires the user to make up for the absence of vision with additional effort on his or her part, not the computer's.

The second property includes all those features that are intrinsic to the metaphor, such as its shape or color. The sighted user benefits from the intrinsic features through the association they provide with real-world "looks" of objects. This is not true when the metaphors are directly shown as their tactual/tactile counterparts. They simply do not "feel" like the real-world objects. Thus the Macintosh trash can may have the visual connotation of a garbage bin, but it will not have the tactual association.

The third property, the relationship of one metaphor to the other metaphors, faces the combination of the problems inherent to the locational and intrinsic properties. As an example, one may consider the relationship among windows in a Windows environment. The NVI user may be restricted to operating in the active window and be required to perform a special function to switch among windows. In this case the display of other windows is completely irrelevant, since the user may view them merely as different sessions in the same interface and the relative size and location of various windows bears no meaning. If the user is allowed to wander outside the active window, it will be difficult to convey the meaning of the objects he or she then encounters. How, for instance, would one differentiate between entering another window versus "touching" on the desktop? If this is done textually or verbally, what is the benefit of a Windows environment in the first place?

The fourth property, the need for mouse control, relates to the problems associated with the mouse as an input device. The mouse exhibits some undesirable qualities which result in a negative impact on anything that requires mouse control.

3.4. Representation of icons and images

Graphical information can be categorized into four general groups each unique in its translation for the NVI user (Vanderheiden, 1989). While the solutions for the representation of each to the NVI user are not optimal, they are the only options current technology will permit.

Icons can usually be translated in only one way: by giving its name and/or icon type directly. However, if the system is also designed with children in mind, this may not work out to be a good alternative. Hatlen and Curry (1987, p.8) state that "Blind children should not be expected to have a more highly developed reading vocabulary than do sighted children. Therefore, substituting words for pictures will not work." Merely translating icons into text, in these cases, may not be of any more value then presenting the icons graphically. A better approach would be to design a system completely geared toward children with blindness, in which the child has the opportunity to develop a more direct and concrete understanding of the computer's functionalities.

Pictographic images currently can only be explored through tactual/tactile means. This may either take place with a full tactile tablet, or a virtual tablet, in which a smaller display provides the "larger picture" by showing display-sized sections as the user moves the window across the virtual image, much like a radar may map its surrounding space by putting together small cross sections. Most ideas relating to the display of pictographic images are shared with those applying to textual data. Again employment of the "What You Feel Is What You Get" principle is desirable. Further, there should be separate exploration and graphics alteration modes. Finally, zooming functions for close-ups and zoom-outs should be incorporated to allow as detailed or general a view as the user desires.

The next group, stereotypic images, such as pie charts and bar graphs, are best given in table format. Frequently the system will need the displayed information in some type of numerical arrangement first, in which case extraction and alternative representation should be viable. If not, the only other solution is the one given for the pictographic images.

The final group, encompassing animated images, defies easy answers. Currently the only viable way may be to freeze the image completely or give snapshots of it in its various stages. This is an area that would well benefit from further study.

In summary, the key to representing graphical information is in providing it textually, if possible. If this is not feasible, the graphical data should be displayed as closely to the original image as is achievable in a tactual/tactile fashion. The display of graphical data clearly precludes the use of spoken output.

4. TYPES OF SYSTEMS FOR NON-VISUAL INTERFACE USERS

Systems for NVI users fall into two general categories: those that build on existing interfaces for sighted users and those that are specifically designed for the user who is blind (Edwards, 1989 {2}). The first are known as overlay or screen reading systems (Figure 4), since their design is based on the assumption that all the information displayed in a GUI can be turned back into pure information without the need to access data below the level of the GUI (Vanderheiden, 1989). Due to the high level acquisition of user interface information, overlay interfaces show a strong technical advantage over specifically designed systems in that they can be integrated with a wide range of existing software packages. GUI platforms for constructing overlay interfaces are readily available (Edwards, 1989 {2}), since most GUIs provide tools to tap into the information flow from and to the operating system and application programs. Microsoft Windows is a basic example of a graphical environment with considerable facilities for user customization. Specifically designed systems, which will be addressed later in this chapter, cannot make use of existing platforms since they are developed from "the ground up" with the NVI user in mind.

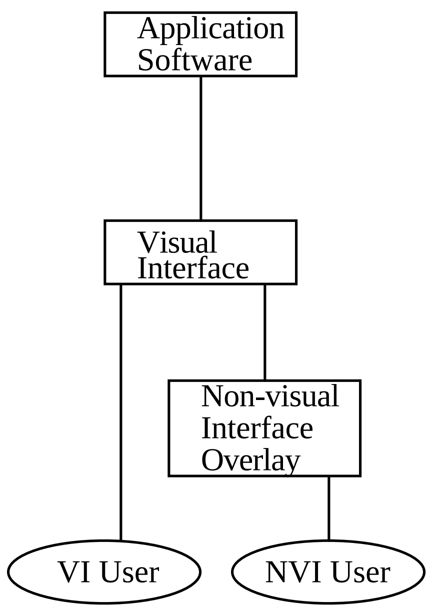

Figure 4: Non-visual overlay of visual interface (Dürre and Glander, 1991)

One well-known overlay system, Outspoken, was created to permit NVI users access to the Macintosh Finder operating system (Edwards, 1991). Some functions provided by Outspoken, such as the redefinition of the number key pad or the special menus listing open windows and icons, have already been covered in earlier sections. Further functions that are specific to the NVI user deal with the scanning technique used to show text and icons. With each down movement of the cursor the system automatically advances across a row of icons or line of text (as seen on the visual screen) and reads them back to the user. The user then has to move the cursor manually back to the beginning of the line or row. While this may be of value for reading text files, this technique is rather cumbersome for finding the locations of the various icons. Another feature of Outspoken moves the cursor automatically into any dialog or information windows so that, like a VI user, the NVI user becomes immediately aware of any input or system requirements.

While these added features try to mimic the sighted user's interactions, they still cannot overcome the sequential nature of tactual or auditory screening (as can be seen in the automated scanning). The result is the creation of an environment that essentially uses an interactive system non-interactively, as noted by Dürre, 1990. Such systems, by their very nature, also run into the aforementioned problem of translating visual metaphors into a form useful to individuals who are blind. Finally, contrary to one of the main principles promoted in this paper, overlay systems do not offer the exhaustive memory aids software systems can and should provide to the NVI user. To exemplify this once more on a system that directly presents a GUI using speech output: "Most subjects were able to develop methods, which were invariably verbal (e.g. mnemonics based on initial letters of window names) to help them remember the screen layout ..." (Edwards, 1989 {1}, p.588). The question again arises as to why the user has to perform all the "memory work", when it is well within the capability of the automated system to provide the same memory functions.

Specifically designed systems eliminate the need to patch a visual environment to suit NVI users. All functionality will naturally be geared toward accommodating users who are blind (Edwards, 1989 {2}). This strategy, however, does not require sighted users to be excluded from the interface, which may be the case for some dedicated systems in which the system is completely separated from any visual interaction. A better approach is to strive for an integrated system in which both VI and NVI users may access the same information simultaneously (Figure 5). An integrated system essentially calls for an extension of the WYFIWYG principle into "What You Feel Is What You Get and the others see". Provision of both a visual and a non-visual interface in parallel is essential to integration, giving the NVI user Braille and speech and the VI user visual output (Dürre and Glander, 1991; Bowe, 1987). The difference between specifically designed systems and overlay systems is in the environment which, in specifically designed systems, can be created to fully suit the NVI user's needs, while still providing the VI user with all the functionality of a GUI, albeit in a less graphical way. Only in such a system can the levels between the sighted user and the user with blindness be equalized to the point where true collaboration can take place.

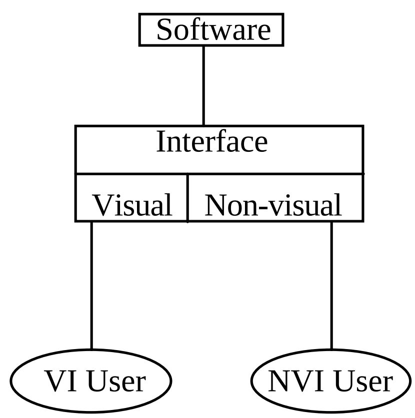

Figure 5: Integrated system (Dürre and Glander, 1991)

A primary example of an integrated system can be seen in the BrailleButler (Dürre, 1990; Schmidt-Lademann and Dürre, 1984). The BrailleButler provides both tactual and visual output to an interactive text editor with additional speech capability. As the NVI user performs operations in the system, the sighted user may follow along on a screen. The NVI user, likewise, can receive information on any actions initiated by the sighted user. While the NVI user perceives a primarily text-based system, in which each step leads to another textual menu or command, the sighted user is presented with pull-down menus and command bars. Although the BrailleButler does not offer the elegance and ease of a GUI to the sighted user, it does provide the tool for both parties to interact without the need to revert to completely text-based systems. One can also envision a fancier version for the future, in which operations by the VI user on icons and windows are converted in such a fashion that the NVI user may follow along (Ternlund, 1991). A copy by pulling a file icon from one window to the next, for instance, might be shown as the necessary non-visual steps to the NVI user (e.g., the user might be shown the copy menu with both the source and destination files shown).

Another example may be seen in the recent development of NonVisual GUIs (NV GUIs) by Dürre and Garza-Salazar, 1992, which transform the graphical user interface into the equivalent nonvisual interface. The premise for NV GUIs is the strict and consistent definition of objects, such as windows, menu bars, pull down menu options, etc. With such definitions any GUI can be interpreted as a navigation tree, which permits all navigational functions to take place with relatively few operators. In this environment, down and up are equivalent to "more and less information" in that each "down key" stroke leads to a more detailed description of the current item pointed to while each "up key" stroke performs the reverse. The user, for instance, might initially only be given the first letters of each available command. Upon depressing the down arrow, a menu of the same commands is shown. Pointing takes place through the left and right arrow keys, essentially forcing choices in one node of the tree (which visually might be portrayed as a two-dimensional array of objects) to be shown non-visually as a series of options or items that can be traversed linearly. A VI user thus may be viewing a window with file icons, while the NVI user views the same as a line of filenames. Navigation takes place either by entering the first lower-case letter of an available command, by selecting the command or by "backing up" and returning to the higher parent node of the current one.

NV GUIs demonstrate clearly that it is possible for both VI users and NVI users to coexist without the need for sacrifice on either part. Unlike overlay systems, which employ a kind of "sweep and scan" technique by the NVI user in that he/she has to perform sweeps of his or her environment to develop an abstract map of the layout, NV GUIs are sequential in nature. The user is always provided with an egocentric point of reference. In more general terms, any specifically designed system can be laid out on an "abstract grid" in which horizontal navigational movements represent traversal of "same level" information, while down and up movements determine the generality or specificity of the current state in the system. Any system built on such an abstract grid will provide the user with the preferred navigational environment. True GUIs, and their overlay counterparts, neither provide that reference nor do they convey the importance of an object non-visually by the position of that object on the grid.

5. SPECIFIC GUIDELINES

To this point, many general and fundamental concepts and considerations have been covered. This final section turns to more specific suggestions of what a system should incorporate and offer to become a more effective tool to the NVI user. The following sections are not exhaustive and will frequently touch on items mentioned earlier. They may also not all apply to every system, depending on the system's intended target group.

5.1. Is start-up of the system easy?

As with systems designed for sighted users not interested in the underlying workings of the computer (for example, the target audience of Macintosh computers), the person with blindness should not have to perform complicated rituals in either configuring or calling up the software system (Betts et al., 1987). Installation of software should require little or no outside help and should take place automatically once the initial set-up program has been started. After the loading, all further call-ups of the new piece of software should occur without the need to traverse directories or to pass systems parameters.

Most software packages today provide enough written guidance so that a beginner does not require much, if any, assistance. The NVI user should have the same tools, which could be provided as "getting-started" tapes or Braille documents. Additionally, some type of user-support hotline would be beneficial (Brown, 1992). Brown gives a rule-of-thumb of about one hour learning time to acquire the basic knowledge necessary to function within a software system.

5.2. Is on-line help readily available? Are manuals electronically stored?

Systems designed for NVI users have the primary purpose of giving them an efficient means for storage, retrieval and manipulation of data. To require the user to access off-line materials for help or systems references is counterproductive. All aides should be immediately available through the system (Brown, 1992; Griffith, 1990; Betts et al., 1987). Furthermore, to minimize reliance on outside help inclusion of a guided tutorial in the software package might be beneficial, as is already the practice with most commercial applications for sighted users. Manuals and on-line help are also good potential candidates for speech output, since text landscape information is usually of little concern.

5.3. Does the system tailor towards the various skill levels?

As a user becomes familiar with a system, he/she should be able to speed up certain actions through built-in short-cuts. While the novice may need guidance by the system at every step, the experienced user should be able to operate without feeling impeded by the system (Dürre, 1986; Betts et al., 1987).

For a novice user navigating through the various menus, each new menu might provide a full array of available options. By highlighting any of the options and pressing "Enter", the desired action is achieved. As experience with the system is gained, the user might make use of a list of letters shown initially at the opening of a menu. Each letter serves as a simple memory aid to an option available in that menu. By entering the letter of the corresponding menu option the user does not have to traverse the entire list of options. As the user becomes completely comfortable with the system, call-up of the more frequently used actions through hot key combinations should be possible.

A similar skill level adaptability could be provided by giving the user the choice of manipulating files through the highlighting of items in path- and filename indices, or by simply entering the same information manually at a dialog prompt. Thus, when opening a file, the user may find the cursor at an "input prompt" which asks for the input of the appropriate filename. He/she may then hit, for instance, the down arrow key which brings up a list of available filenames that in turn can be highlighted for selection. In this way a file can be manipulated without the user having to recall the path- or filename from memory.

5.4. Is sufficient feedback provided?

Most users will prefer to be in complete control of their machine's operation. The secure feeling of such control can only be provided if the user is aware of its actions at all times, and this can only be accomplished if the system keeps the user updated and informed. Feedback ranges from offering directions on the next expected step to error messages in case of system failures. The system should provide enough information in both settings to let the user continue comfortably without having to refer to any manuals. (This includes such dreaded messages as: "The system has experienced a severe error. Please reboot.") Another use can be seen in actions that vary in time, such as the saving of a file. As soon as the operation has been completed, the computer should make the user aware of it. (This could be a short auditory signal after the task has been finished.)

5.5. Are the information access needs of different users accommodated?

Considerations here range from different output media to suit individual users' preferences, to arrays of modes within the same media (Betts et al., 1987). A user with no tactual/tactile input/output skills should have full speech support, while a user with complete Braille skills should be able to rely solely on his/her sense of touch, if so desired. Individual preferences should also be satisfied within the same medium. For users in tactile text reading/browsing mode, the software should enable the user to read with at least three separate settings, as mentioned before, i.e., fixed interval window advancement and variable window advancement with visible word cut-off and without visible cut-off. Similarly, speech output might offer two settings: one in which the spoken output stops at the end of each line on a page; another in which the output stops at the end of each sentence, regardless of the page structure. For spreadsheets it should be possible to alter the length of the window jump to fit the width of the individual columns.

5.6. Are navigational devices completely interchangeable?

The system should cater to individual preferences. Some users may wish to use the standard arrow keys, others may prefer Braille input, while yet others may be limited to voice commands. Any user should be able to create a comfortable work environment and have the choice of switching between any of the available forms of navigational input. Certainly there are users that prefer the direct control of a joystick when browsing or reading, yet appreciate the immediacy of Braille board navigation when editing text.

5.7. How easy is the call-up of locational information?

It has been explained earlier that the user, at any time, should be able to access information about his or her location within a text file, spreadsheet or any other type of software that requires operating in a large two-dimensional environment. The information should be available both in "absolute" reference to the "home point" of the file, such as the current row and column location in the entire file, and in a relative reference to the current logical unit of the working environment, such as "line y on page x, column z", or "row x, column z on the lower quarter of the graphics sheet." Furthermore, similar information should be available for the work area that is visible to the VI user, but normally of little concern to the NVI user, so as to permit a common work frame for collaboration between VI and NVI users (i.e.: "Go to line 20, on the right side of the window"). Not only should the NVI user be able to call up such information, but he/she should also be able to use it as a means of specifying the next target location, such as "I want to go to page 20." A subset of these functions may lend itself well to the use of hot keys.

5.8. Does the system facilitate window-to-cursor binding?

Window-to-cursor binding is a technique in which the window harbors the cursor at all times, thus following any move the cursor makes. In essence, the window and the cursor have become one unit. The binding promotes a working location awareness in the user, since he/she will always follow any motion of the cursor, be it continuous, such as scrolling up and down, or discrete, such as jumping to a new location. This mostly applies to small displays, such as the Braille display, since the physical window has to be small enough in size so as to not require the user to have to engage in a wider search of the cursor position. To justify this finding, one needs to examine the typical exploration scan of the NVI user. It takes place in two parts, the movement of the window and the subsequent scan with the fingers. With a small window, the area to be scanned is restricted to where the user has to perform few finger motions, essentially eliminating the finger scanning. When the window is bound to the cursor, the user will always realize the immediate working area without wasted time for the cursor search. The fingers, the display and the cursor become one unit, promoting direct manipulation of any object, be it a text file or a command menu, which in turn adheres to the WYFIWYG principle. Each movement of the window automatically translates into a "pointing" to an item, without either the user or the computer having to perform additional tasks. The user can, for instance, mark text without having to split the operation into separate read and cursor movement processes. At the same time, the computer can skip to a search item without the user having to engage in an elaborate search of the cursor. The window will already be at the desired location (Schmidt-Lademann and Dürre, 1984; Dürre, 1986).

5.9. Can the functional keys be redefined?

Reassigning different tasks to functional keys may be a primary concern to the NVI user in an effort to create short cuts for tedious keystroke sequences or macros for various application command routes. It is especially important to the advanced user to be able to customize such features, since it may provide greater security on complex operations, given that there may be no immediate feedback on whether the user performed the operations correctly. Once the operation assigned to a functional key has been worked out and tested, one stroke of the key will guarantee the desired effect. Unfortunately, in giving the user such freedom, one also has to consider the possibility of interference between the functional key definitions of different application programs. If the keys retain the user-defined properties from one system's module to the next, the definitions may result in undesirable actions in the various modules. If each application provides separate key definitions, integration of the system modules and system consistency might be compromised. Thus there is a trade-off in which user-flexibility should be constrained by system considerations, but only to the point needed to guarantee integrity and overall consistency (Brown, 1992).

5.10. Are the limitations of computer Braille taken into consideration?

With the advance of the computer as an aid to the user with blindness came the differentiation between regular Braille and Braille used in computing. Just as there are some features available in handwriting that are inconvenient to provide in an automated fashion (such as the conventional notation of division), there are features in regular Braille that would create obstacles not easily overcome in an automated environment. Computer Braille, specifically designed for the computer, alleviates many problems but is more limited in the number of characters than the conventional counterpart. In Computer Braille many special symbols are not represented specifically. One may be tempted to provide a solution in additional special codes, such as specific math codes, but these too bear limitations in that they require the user to switch to another character set for any characters not found in the ordinary Braille set. One solution lies in circumscription, which uses multiple character codes to describe symbols not represented in the computer Braille alphabet, as is the case in many programming languages. (For instance, the use of '**' in Pascal to represent 'to the power of'.) Through circumscription the user is not required to use or know special codes, but still has the ability to work with any desired symbols. While this may ease the burden on the user, the NVI designer may face additional tasks in providing the necessary translations for various circumscribed characters, particularly if the characters are part of code that controls the computer's functions, such as a user-created program (Schmidt-Lademann and Dürre, 1984; Dürre, 1986).

5.11. Are user choices restricted to those that make sense?

Most GUIs now restrict the user to those choices within a menu or query box that are available in the given context. It does not make sense to be able to choose "close" on a file that is not open in the first place. The same should apply to any system for NVI users, eliminating the need for the user to consider actions that are not appropriate at the time (Edwards, 1991). This could be accomplished by either skipping choices that are not available (Dürre, 1986) or marking them through some auditory or tactile means (such as a clicking sound when such choices are first encountered) as being unavailable.

5.12. Is error-recovery relatively easy?

The main principle to this point can be expressed in one word: "forgiveness". When the user makes a mistake, the system should provide as much aid as possible in reversing the erroneous action (Betts et al., 1987). This ranges from the simple "undo" command in text editing, to back-up files for word processing documents currently being changed, to the recovery of deleted files. If any action by a user is not reversible, he/she should be made well aware of the consequences (Apple, 1987).

5.13. In Braille-based systems: Is there enhancement through an auditory channel?

Most software systems these days employ some sort of auditory enhancement, mostly to alert users of possible dangers and mistakes, or to indicate readiness for the next action. Such functions are especially important to the NVI user, since the limited concurrency of input and output affects the immediacy of the perception of warnings, errors and changes of state. Considering that the user does not have the luxury of merely casting a glance at the message accompanying the alert, messages with different priorities should be distinguishable by their tone or volume or, as noted earlier, such applications may present excellent opportunities for the use of speech, eliminating the need for the user to change from one output context to another. Similar consideration for sound may be given to text editing, where changes in the appearance of text segments, such as italics or underlining, could be indicated through various sounds, rather than tactual or tactile cues. Any messages are temporal in nature, whether they come from warnings, dialog boxes, or state changes, and can benefit from that quality in sound output. Sounds may be used to represent a wide variety of contexts, so long as the user does not have to strain to memorize the significance of each. A short, sharp sound, for instance, might indicate the presence of a severe error, while a soft "hum" may announce underlining in text.

If speech is used in text editing, showing text features may be potentially easier than through any other method. Each special segment, when first encountered in reading, could be indicated by the word "start" being spoken, followed by a one-word description of the feature, such as "bold-face". When the last character of the segment is passed, the same might be repeated through a word combination incorporating the word "end". As another example, blank lines may be declared to be empty without the user having to search the entire line. Such statements, of course, should only take place if the user has opted for them.

In any case, speech and sound enhancement may be of considerable value in a tactile/tactual system. Sound, however, should not be turned into a gimmick overwhelming the user with various beeps, gongs and buzzes. It should be provided in a restrained manner, so that each carries its own distinct meaning (Pitt and Edwards, 1989; Apple, 1987). Unlike speech output, sound cannot convey the exact warning or message it is meant to announce and thus should only be redundant to the appearance of any message. The user still should have the ability to tactually/tactilely check on any message. Sound should also be unobtrusive enough so as to not become an annoyance. Finally the user should be able to adjust the enhancement to his or her individual preference, mainly by determining the volume of any sound signals.

5.14. Can the system be used in a noisy or quiet environment?

While the previous point emphasized the usefulness of speech and sound in augmentation of tactual/tactile systems, it should be noted that speech and sound can limit the range of environments in which the user can use the software system, if the ability to turn the audio output off is not provided. In the case of a student user it might be undesirable for the system to give off any type of audible information, since it might prove to be distracting or unwelcome in a classroom setting. Headphones may alleviate such problems in some situations, yet they also may interfere with the student's assimilation of lecture material and adversely affect participation in group activities (Schmidt-Lademann and Dürre, 1984). On the other end of the spectrum the user may be operating in a noisy environment, rendering some of the spoken output unintelligible. Thus, users with tactual/tactile abilities should be able to adjust system's output configurations to the most suitable ones for any environment. Sound and speech should serve in such cases as augmentation, not as the primary basis of information relay.

5.15. Can the system be run on a portable computer?

With the proliferation of laptop and notebook computers, the ability of an individual with blindness to take his or her system to any desired location becomes more of a necessity than a luxury. Any system should be designed with the use of small machines in mind. Though almost any machine these days is quite powerful, it still might not be able to support an all-encompassing software system for NVI users. In consideration of this, it would be prudent to allow the user to take a scaled-down version of the original non-visual software package for the smaller hardware system, with a standard set of minimum functions to support the most needed routines. Alternatively the user might be provided with the ability to specify the desired functions on the portable computer. The system would ideally provide a list of options from which the user could choose and assist in setting up the selected items on the smaller machine, without the user being subjected to extensive reconfiguration procedures. As the race for smaller and more powerful machines continues, hopefully this point will soon be of no concern.

5.16. Are useful day-to-day functions provided?

Software systems for people with blindness not only offer opportunities to aid the user by enabling access to a large informational and computational domain, but also provide an excellent medium for making life easier on a small scale. Most GUIs now supply the sighted user with calculators, calendars, alarms, address books and games. These functions should be offered to the NVI user as well, who otherwise may have more difficulty obtaining the same information in traditional ways. The designer should consider going well beyond what one considers standard amenities to sighted users and incorporate additional features the NVI user might appreciate, including, but not limited to, check writing applications, automated form letter support for routine written messages, automated envelope addressing, and so on (Dürre et al., 1983). With the rise of multimedia, further opportunities will abound.

5.17. Does the system support a wide range of special peripheral devices?

In conjunction with the previous point, that the system should support a wide array of software aids, such as a check-printing feature and a clock, the system should also do the same for peripheral devices, in part to enable the previously mentioned functions. The system should be able to accommodate not only a standard printer but also a special check writing printer and perhaps a scanner and fax modem. Again they should be supported without the user having detailed operating system's knowledge or needing to go through complicated switching procedures. An "all-around" system might allow simultaneous or "as-needed" connections to the regular keyboard, a Braille keyboard, a Braille display, a graphics display, a laser printer, a tractor feed printer (for check writing and other non-standard printing applications) and a joy stick. Though such a scenario might be pushing the physical limits, a software system should at least attempt to support any desired configuration (Dürre et al., 1983).

5.18. Does the system support expansion and modification?

The software for persons with blindness should be easily upgradeable, expandable and modifiable, as is true for most modern environments (Betts et al., 1987). The user should be capable of installing upgrades without outside help and in-depth knowledge of the workings of the system. The user should also have the option of adding desired functions or deleting unneeded ones without hassle. Rather than requiring the user to browse through technical manuals to gain the knowledge for making any changes, the system should use automated scripts that query the user on the needed information. The key to expansion and modification is modularity of the software (Betts et al., 1987). Blocks in the system, from hardware drivers to software functions, should be removable and exchangeable as the user's requirements change.

5.19. Is compatibility with other software systems provided?

In the tradition of software flexibility recommended in the previous sections, the system should provide good interfacing to other application packages (Betts et al., 1987). The NVI user should be able to call up and edit, for instance, WordPerfect files without having to switch to the other word processor. At the same time the user should have the ability to put a file from his or her system into a format compatible with word processors preferred by VI users. The system should provide interfacing to entirely different operating systems, such as between UNIX and DOS. While this is easier said than done, this may be one of the most fundamental steps in providing the bridge between the working environment of the user with blindness and the rest of the world, without forcing either VI or NVI user to give up their preferred tools. Compatibility is one of the principles that will enable full collaboration between both parties.

5.20. Are terminal emulation and networking capabilities provided?

A further step towards the integration of the user with blindness into a computing environment is access to networked systems. The ability to incorporate some type of networking and terminal emulation software into an NVI system is essential.

Ideally, some form of interchangeable software module, which features a well-known terminal-type interface to the network with a custom front-end specifically developed for the NVI user, would be desirable. Depending on the target network, the system may require very little modification of host information for the user interface, as would be the case for a text-based host system (such as UNIX), or may need to be elaborate for systems that use a more graphical approach (such as the upcoming Windows NT). It is to be hoped that the latter host software would provide tools for intermediate information access similar to the widgets in Xwindows since such tools would allow the designer to tap into lower-level systems information. Access to the lower-level information would provide the designer with the platform for a custom-made NVI alleviating the need for an overlay system. Finally, user-friendly modem support should be included to allow the user access to the wealth of information available through dial-in systems, both commercial and educational.

5.21. Does the system bridge the gap between users with sight and users with blindness?

The previous guidelines will lead to improved communication between the NVI user and his/her surrounding world. To effect complete integration of users with blindness, though, the system has to promote simultaneous use by both VI and NVI users. While the system should primarily be designed for the user with blindness, the sighted user should also feel comfortable when working with the system and find elements familiar from GUIs. This requires showing the entire list of choices on a menu, even though the NVI user will be interested in only one menu choice at any given time. It necessitates the graphical display of a block of text with all the special features (such as underlining), though the NVI user may not be concerned with the special features at the time of the review. The sighted user should be able to track on the monitor each of the operations of the user with blindness. Similarly, the user who is blind should be able to follow the actions initiated by the sighted user. Perhaps the only inconvenience to the sighted user should be the substitution of the mouse with a device giving more continuous and grid-bound movement to prevent confusing "jumping around" by the sighted person.

While the ability to have simultaneous use by both VI and NVI users encompasses a wide range of possibilities and configurations, it is also the one that will guarantee a full integration of the user with blindness into the work environment. When addressed, this guideline transforms the computer from a basic tool for communication between NVI and VI users, to a true facilitator of communication (Dürre and Glander, 1991).

5.22. Some general guidelines

This section is devoted to items that are too short to warrant sections of their own, yet still should be considered in designing systems for NVI users. For screen reading systems it is important to make the user aware of the differences between windows, menus, pop-up boxes, etc., since each type of object will provide a different context for the user to operate in (Edwards, 1991). After all, if the differences were insignificant, why would the visual system incorporate them?

The inclusion of some sort of reminder, based on elapsed time since the appearance of the last system initiated query, may be advisable for some actions that require user-input. This will ensure that the user is aware of any additional steps needed for an operation, without wasting time while the machine is sitting idle waiting for the next user action. Thus the user may have entered a file deletion command, but forgotten the safeguard "Are you sure?" question. The system should generate a sound after a few seconds to remind the user that more input is required.

The system should provide consistency in the user's environment (Betts et al., 1987; Ternlund, 1991). As mentioned before, this may not always be possible, or it may have to be sacrificed to more important concerns, but the more the user knows what to expect in various contexts, the less time he/she will have to spend learning (Apple, 1987; Ternlund, 1991).

Communication with the user should be in concise and simple terms. It should not require a hacker to make sense of system messages. For instance, "File not found" is by far preferable to "I/O error 599".

5.23. Was the user involved in the design work?

Only a person who is blind will fully understand the needs of users with blindness, and only a representative group of such users will be able to come up with all the variations and optimal solutions in designing a software package for persons who are blind. It only makes sense, then, to include the target audience in every step of the design. What better way to find out about the functionality of a tool than through using that very tool to enhance it?

6. CONCLUSION

There are many details in developing a system for persons with blindness and most require various tradeoffs to accommodate specific groups of users. There are, however, three points of greatest importance in developing a system that will give the user who is blind the tools for a productive and comfortable computing environment.

First, the designer of an NVI must be aware of the narrowed information intake bandwidth, as compared to a sighted user's "massively parallel" visual capabilities. The user who is blind has to either contend with the low resolution of tactile/tactual exploration of interfaces or the temporal nature and one-dimensionality of auditory media. The simulation of visual aspects of GUIs in either media in the hopes of adapting the interface to the needs of a user with blindness is not possible without sacrificing effectiveness and ease of use. In short, users who are blind need tools specifically developed for non-sighted interfacing.

Secondly, the software designer needs to recognize the many potential aids the computer offers for minimizing the obstacles a person with blindness may have to contend with. These aides do not just appear as the software is written. Instead, they must be developed as the entire software is being formed, with the foremost objective of addressing each and every need the NVI user may encounter in an automated environment. This disagrees with the practice of adapting existing software for sighted users to a non-visual environment, and calls for the development of systems specifically designed for people who are blind. In its strongest sense, it requires creating every module with features accommodating the narrowed information bandwidth of the user with blindness, and incorporating into every piece sufficient memory aides to allow the user to spend as little of his/her mental activities on non-essential tasks.

Finally, in order for the user who is blind to be able to work as productively and effectively as his/her sighted counterpart, he/she must have the same access to information as any other user. Of equal importance, the user with blindness needs to have the same computational capabilities as any sighted user which includes equal opportunity for information exchange with other users. Any system for people with blindness must fully integrate both users who are blind and sighted users with a common platform on which both parties enjoy all aspects of information manipulation without differentiation between their capabilities. While this will eventually take place in the lower software layers, the goal should be to achieve integration at the highest possible stage of information processing. Merely providing simple file exchange facilities between VI and NVI users is insufficient. Instead the challenge lies in creating an environment in which both parties can directly and unconditionally interact while remaining with an interface that provides the paradigm of WYFIWYG to one, yet WYSIWYG to the other. Once this final goal has been achieved, the computer becomes a true facilitator of integration reaching well beyond the status of a mere tool of data access and processing.

REFERENCES

(Apple, 1987): Apple Computer, Inc.; "User Interface Guidelines: The Apple Desktop Interface", Addison-Wesley Publishing Company, Inc., 1987.

(Betts et al., 1987): Betts, B., Burlingame, D., Fischer, G., Foley, J., Green, M., Kasik, D., Kerr, S., Olsen, D., Thomas, J.; "Goals and Objectives for User Interface Software", Computer Graphics, vol. 21, no. 2, Apr. 1987, pp. 73.

(Bliss, 1980): Bliss, J.; "The Case for Volatile Braille", IEEE Spectrum, vol. 17, May 1980, p. 38.

(Bowe, 1987): Bowe, F.; "Making Computers Accessible to Disabled People", Technology Review, vol. 90, no. 1, Jan. 1987, pp. 52.

(Boyd et al., 1990): Boyd, L. H., Boyd, W. L., Vanderheiden, G. C.; "The Graphical User Interface: Crisis, Danger, and Opportunity", Journal of Visual Impairment & Blindness, Dec. 1990, vol. 84, no. 10, pp. 496-502.

(Brown, 1992): Brown, C.; "Assistive Technology Computers and Persons with Disabilities", Communications of the ACM, May 1992, vol. 35, no. 5, pp. 36-44.

(Croft, 1985): Croft, D.; "New Chances for Blind Workers", Popular Computing, vol. 4, no. 11, Sep. 1985, pp. 39.

(Dürre et al., 1983): Dürre, K., Schmidt-Lademann, F.-P., Mehring genannt Friehoff, T.; "Das Mikrorechner-System BrailleButler als Schreib- und Kommunikations-Hilfe im täglichen Leben Blinder", Universität Karlsuhe, Dec. 1983.

(Dürre et al., 1984): Dürre, K., Mehring genannt Friehoff, T., Schmidt-Lademann, F.-P.; "BrailleButler, A Successful Microcomputer-Based Aid for Mainstreaming Blind Children", Proceedings of the Third Annual Workshop on Computers and Handicapped, IEEE Press, Nov. 1984, pp. 17-22.

(Dürre, 1986): Dürre, K.; "Braille and Advanced Man-Machine-Interaction", Computerized Braille Production: Proceedings of the 5th International Workshop 1985, Schriftenreihe des Universitäts-Rechenzentrums, Nr. 1, Katholische Universität Eichstätt, Eichstätt, Germany 1986.

(Dürre and Dürre, 1986): Dürre, K., Dürre, I.; "Electronic Paper for Blind Children", Education & Computing, no. 2, 1986, North-Holland, pp. 101-106.

(Dürre, 1990): Dürre, K.; "BrailleButler: A New Approach to Non-Visual Computer Applications", Third Annual IEEE Symposium on Computer-Based Medical Systems, 1990, pp. 97-104.

(Dürre and Glander, 1991): Dürre, K., Glander, K.; "Design Considerations for Microcomputer Based Applications for the Blind", Human Jobs and Computer Interfaces, Nurminen and Weir, Elsevier Science Publishers B.V., 1991, North-Holland, pp.71-77.

(Dürre and Garza-Salazar, 1992): Dürre, K., Garza-Salazar, D. A.; "Graphical User Interfaces for Blind Users: A New Approach", Dept. of Computer Science, Colorado State University, CS-92-103, Dec. 1992.

(Dürre): Dürre, K.; "Computer-Mediated Communication between Blind and Sighted Individuals", Computer Science Dept., Colorado State University. (Intra-departmental paper).

(Easton and Bentzen, 1987): Easton, R. D., Bentzen, B. L.; "Memory for Verbally Presented Routes: A Comparison of Strategies Used by Blind and Sighted People", Journal of Visual Impairment & Blindness, Mar. 1987, pp. 100-105.

(Edwards, 1989 {1}): Edwards, A. D. N.; "Modelling Blind Users' Interactions with an Auditory Computer Interface", International Journal of Man-Machine Studies, no. 30, 1989, pp. 575-589.

(Edwards, 1989 {2}): Edwards, A. D. N.; "Soundtrack: An Auditory Interface for Blind Users", Human-Computer Interaction, vol. 4, 1989, pp. 45.

(Edwards, 1991): Edwards, A. D. N.; "Evaluation of Outspoken Software for Blind Users", Dept. of Computer Science, York, England, YCS 150, December 1991.

(Griffith, 1992): Griffith, D.; "Computer Access for Persons Who are Blind or Visually Impaired: Human Factors Issues", Human Factors, vol. 32, no. 4, Aug. 1990, pp. 467.

(Hatlen and Curry, 1987): Hatlen, P. H., Curry, S. A.; "In Support of Specialized Programs for Blind and Visually Impaired Children: The Impact of Vision Loss on Learning", Journal of Visual Impairment & Blindness, Jan. 1987, pp. 7-13.

(Lazarro, 1990): Lazarro, J.; "Opening Doors for the Disabled", Byte, vol. 15, Aug. 1990, pp. 258.

(Normand, 1991): Normand, V.; "Spoken Language Translation in a Travel Information Domain: Combining Voice and Graphic Interaction", School of Computer Science, Carnegie Mellon University, CMU-CS-91-104, Jan. 1991.

(Pausch and Leatherby, 1991): Pausch, R., Leatherby, J. H.; "Voice Input vs. Keyboard Accelerators: A User Study", Dept. of Computer Science, University of Virginia, TR-91-22, Oct. 7, 1991.

(Pitt and Edwards, 1989): Pitt, I., Edwards, A.; "Navigating the Interface by Sound for Blind Users", Computers and People, no. 5, 1989.

(Scadden, 1984): Scadden, L.; "Blindness in the Information Age: Equality or Irony?", Journal of Visual Impairment & Blindness, vol. 78, Nov. 1984, pp. 394.

(Schmidt-Lademann and Dürre, 1984): Schmidt-Lademann, F.-P., Dürre, K.: "Rechner-arbeitsplätze für Blinde", Informatik-Fachberichte 92, Springer Verlag, Berlin/Heidelberg, 1984, pp. 303-316.

(Ternlund, 1991): "Non-Visual Representation of Graphical User Interface Objects. Similarities and Differences between 3 Popular Graphical User Interfaces at Programming and User Levels: Macintosh, Microsoft Windows, Motif", Independent Study Paper, CS Department, CSU, 1991.

(Vanderheiden, 1989): Vanderheiden, G. C.; "Nonvisual Alternative Display Techniques for Output from Graphics-Based Computer", Journal of Visual Impairment & Blindness, Oct. 1989, vol. 93, pp. 383-390.