Proceedings of the International Conference on Universal Access in Human-Computer Interaction 2001. August 5-10, 2001. New Orleans, LA.

ViKI: A Virtual Keyboard Interface for the Handicapped

Blaise W. Liffick, Ph.D.

Department of Computer Science
Millersville University
Millersville, PA 17551 USA
1.717.872.3536
liffick@cs.millersville.edu

ABSTRACT

Although computerized assistive devices for most handicaps are becoming more widely available, one major drawback is that users must currently provide customized interfaces for every computer system they use. In addition to specialized input and output devices, such assistive devices also frequently require customized software. This paper proposes an alternative solution, by providing a customized interface (both hardware and controlling software) on a laptop computer that is then used to control standard software on any other host computer. In this way, the user can carry their specialized interface to essentially any general computer, and can interact with any standard software supported on the host. The laptop system becomes a virtual keyboard to the host machine.

1.0 INTRODUCTION

As accessible as assistive devices have made computers for handicapped users of all types, there is still at least one situation in which such users of specialized interfaces are at a distinct disadvantage. Assistive devices generally allow a user to set up a one-to-one relationship between the specialized device(s) and a particular computer. While this certainly helps users to create a customized workstation that satisfies their personal needs, it is, nonetheless, allowing customized access to just one computer. There are many situations that come easily to mind where this is insufficient: laboratory settings where particular computers may have specific software or hardware needed by the user, that cannot be easily made available on their own workstation; academic settings where the user may need to use computers in more than one lab; business and industry settings where the user needs to collaborate with others on a project, and so must be able to access computers other than their own; any time the user travels but needs to access a computer.

The ViKI (**Vi**rtual **K**eyboard **I**nterface) project is an attempt to demonstrate the efficacy of using a laptop computer as a virtual keyboard to access a host computer that is running standard software. Figure 1 shows the Specialized Interface Laptop (SIL) system connected to a host computer running standard software that will be controlled by the user of SIL. SIL provides any specialized connections required for assistive devices, and any special screen display the user might need in order to interact with those devices (e.g. a scanning keyboard representation). Upon entering an input into SIL, the input is then passed on to the host computer in a form as if it were generated by a standard keyboard and mouse combination.

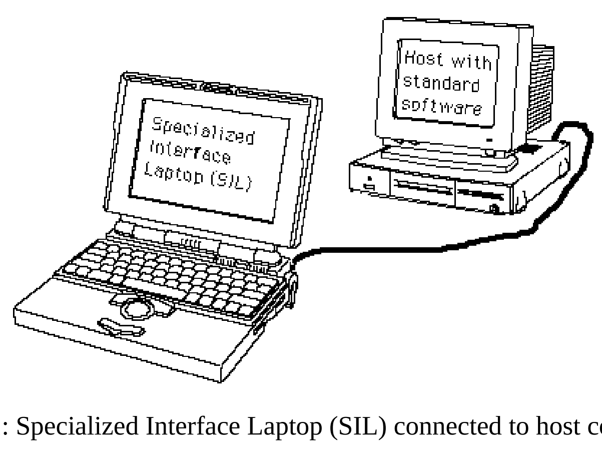

Figure 1: Specialized Interface Laptop (SIL) connected to host computer.

The hardware and software on SIL can be customized to the specific needs of the user, while the hardware and software on the host remain unchanged. In this way, the user can connect SIL to any host computer he or she likes, while maintaining his or her customized interface on a portable laptop.

2.0 BACKGROUND

The impetus for this project comes mainly from personal experience with setting up specialized interfaces for disabled students. There have been occasional attempts on campus to set up a “handicapped workstation” in one lab or another in order to assist a particular student. This typically entails purchasing the specialized hardware and/or software that a student with a particular handicap could use to access a computer in one of the campus labs. One specific computer in the lab is designated as the special “handicapped station,” which is then used almost exclusively by the one student for whom it was modified.

This has been greatly unsatisfying for a number of reasons. First, such modifications often have made the workstation unusable by non-handicapped students. This has meant that this particular station in the lab has been greatly underutilized compared to all other lab systems. Consider the need of a particular individual for a multi-modal access mechanism, such as suggested by Smith *et al* [10], where the interface might consist of several devices such as a head-mounted pointing device in conjunction with foot controls and speech recognition. Second, customizing the station for one particular handicap has often meant that the station is unusable by students with other, different handicaps, making it necessary to provide additional special workstations within the lab in order to accommodate multiple handicapped students. This is also a problem when there is more than one student in a particular class requiring accommodation, even if they could make use of the exact same hardware and software. Third, most accommodations for handicapped users are truly customized to the individual, so that any generically configured system will likely fall short of the needs of an individual user. Finally, on our campus we have numerous general-purpose labs, as well as many special-purpose labs dedicated to a particular department and/or application area. It is unrealistic to imagine that all of these labs can provide assistive devices to accommodate all handicapped students, yet we are legally (and ethically) required to provide such accommodation in order to avoid limiting a particular student’s access to a specific educational experience.

Further impetus is provided by the observation of the increasing ubiquitousness of computer technology. The growing availability of automatic teller machines (ATM), personal data assistants (PDA), cellular telephony, interactive kiosks, interactive television, Internet access, etc., all points to a need for providing handicapped users with personal portable interfaces [1, 9]. The need for such “universal access” continues to increase as assistive devices continue to make handicapped users more mobile and independent [11]. Consider, further, that the number of people with handicaps continues to increase, and should probably also include the fact that people are living longer but are developing disabling conditions as they age [7, 12]).

The notion of an electronic notebook [2] suggests that non-disabled users would benefit from a portable system customized to intellectual needs. While such a device typically includes general tools such as an appointment calendar, “To Do” list, or email system, there is some indication that how the system would be used by a particular individual would make the system in some fashion desirable. At the very least, particular professional activities might require specialized software in order to accommodate, say, a physician’s notebook vs. a biologist’s notebook vs. a writer’s notebook, etc.

Perhaps the ultimate extension of such personal devices is the wearable computer [2, 5]. In these cases the system takes on a very personal relationship to the user, and in a way becomes like an additional sense, i.e. as an additional way for the user to interact with the surrounding world. The portable nature of these systems, and the aspect of these systems being an additional means for gathering and interpreting input from the external world are quite important.

Feiner [4] has suggested that there has been a growing impetus to “decouple the personal computer and its interface...” His ultimate extension of this notion is hybrid devices that marry small, precise controls for use over small domains with less precise devices controlling access to larger virtual spaces. In our case, the small, precise controls might be our laptop device, while the larger domain is the host computer.

Specialized mobile systems are also of interest with respect to this project. Mobile information browsing [1, 5] is likely to increase substantially in the near future. The use of mapping systems in automobiles, electronic

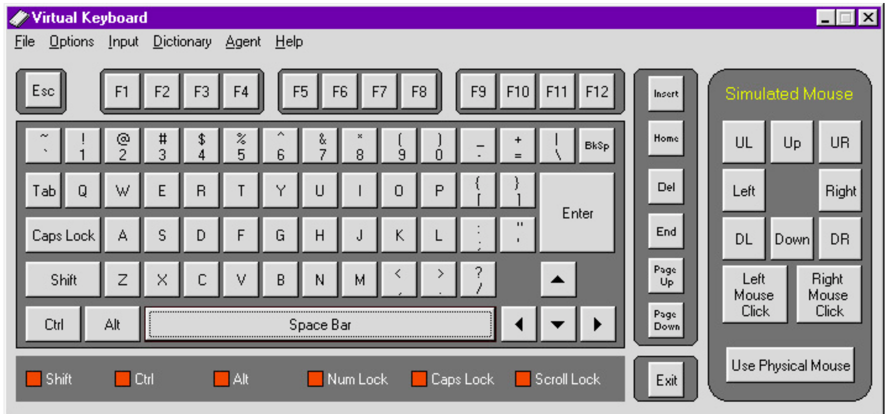

Figure 2: The Virtual Keyboard screen layout.

books, personal electronic assistants [3, 9], etc. all point to a trend in the use of portable electronic devices to access an increasing amount of information.

3.0 THE PROJECT

The ViKI project was envisioned in three main phases. Phase 1, the main subject of this paper, developed the necessary software for the SIL system to mimic all of the interaction methods available from standard keyboard and mouse, with communication between SIL and the host system via a serial interface. Phase 2 of the project will be to replace the serial communications with a direct connection to the keyboard port of the host system, thus eliminating the need for the TSR communication program for the host. Phase 3 will be to replace the hard-wired connection between SIL and host with an infrared wireless system.

3.1 Phase 1

The SIL software runs under a Microsoft Windows® environment and was developed in Visual Basic®. It consists primarily of the virtual keyboard emulator (VKE) and software for communicating with the host system (see Figure 2). The host system is also Windows® and Visual Basic® based. The two systems connect together through their serial ports using a null modem cable.

The VKE software provides an on-screen representation of a keyboard, which can be manipulated with either the laptop's keyboard or through pointing and clicking with the laptop's pointing device (we connected an external mouse for demonstration purposes). We are not proposing this representation of the display as an actual useful interaction method, but are using it as a means to demonstrate the efficacy of the idea. Later research will explore options for providing actual interfaces that might be suitable for assisting particular handicapped users.

There was no attempt in this first phase to in any way optimize the virtual keyboard or the interaction methods of the user. As seen in Figure 2, the on-screen VKE is simply mimicking a fairly standard keyboard layout. The VKE resembles a standard keyboard and has most of the features of one. When the VKE software is run, data is read from the system registry and loads either the appropriate customized features or a set of defaults. Since this demo system has also been set up for speech recognition, the speech dictionary is also loaded at startup time.

The keyboard buttons can be clicked using the pointing device to send a keystroke equivalent to the host system. The VKE also provides a mouse simulator (the right side of Figure 2), which will move the host's mouse cursor in the indicated direction. The VKE's menu bar houses access to customizing controls, such as selecting the main input device, setting background color, mouse speed, etc (see Figure 3). Most options within the menu bar are saved in the system registry, allowing the user to customize the VKE to his or her need.

At this time, interaction with the VKE is provided through external pointing devices such as a mouse, trackball, or joystick, through the laptop's keyboard, and through a commercial voice recognition system. The pointing devices can be used to either directly control a cursor on the SIL display (in order, for instance, to point to and select by clicking a particular key on the VKE), or to remotely control the cursor on the host system.

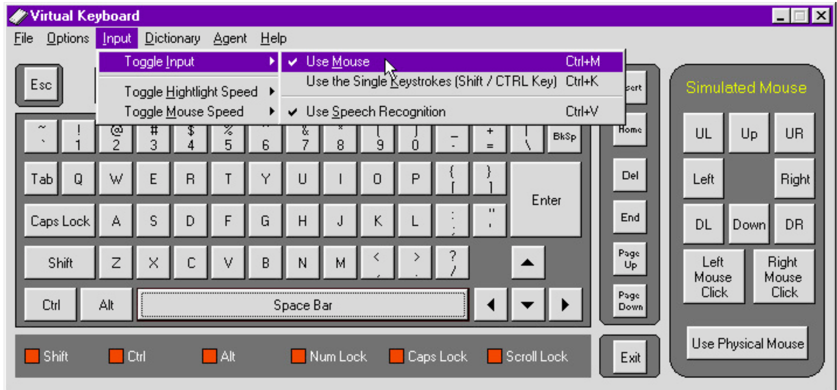

Figure 3: A look at the nested **Input** menu, containing user controls

Interaction can also be set to a scanning keyboard mode, whereby the user can select a keystroke to be sent to the host using a single-switch input. In this mode, the rows of buttons begin to highlight one by one. If the user presses the *shift* key, that row is selected and scanning continues by then highlighting one button at a time through the row. A second pressing of the *shift* key will select that one key for transmission to the host computer. After selecting one key, scanning is resumed. Scanning speed can be set through a menu control. Another control that can be set by the user is sticky keys, so that multi-key combinations can be entered one key at a time.

The SIL system can also act in a stand-alone mode while still using the VKE software. This option allows information to be sent to an active application running on the laptop. In order for the VKE to perform this action, VKE must know which application program to send data to. To accomplish this, the VKE presents a list of all applications found on the laptop's hard drive, allowing the user to select one to execute. Upon doing so, the VKE obtains the ID of the application, which allows VKE to send data to the application.

From the host system side, a small terminate-and-stay-resident (TSR) program (the Communicator) is required at this time. Communicator accepts SIL signals as input via the host's serial port, and reroutes those signals as input to some application running on the host. The Communicator must have primary control of the host system, although the application software must have also been started but suspended. Once the application has been started, Communicator is run and immediately minimized and placed in the system tray.

3.2 Phase 2

The second phase of this project will be to replace the serial connection on the host side with a direct connection to the keyboard port. The goal is to eliminate the need for the Communicator software on the host computer, so that there is no need whatsoever to modify anything on the host system. This will make the SIL truly portable (at least with respect to Windows-based systems), allowing the SIL to be connected to any compatible system in any lab on campus.

3.3 Phase 3

Most laptops today come equipped with an infrared port. This may make it possible to eliminate the need to physically plug the SIL into the host computer. We envision being able to simply carry the SIL within carrier distance of the host computer and being instantly linked. Infrared interfaces are already used as a means of transferring files between systems. In addition, wireless keyboards are widely available from a number of sources. Our hope is to replace the wireless keyboard with a laptop acting in its place.

Admittedly this idea seems to reintroduce the drawback of the Phase 1 Communicator software, i.e. that the host system must be modified in some way in order for the SIL system to communicate with the host. This is true - every system the user wishes to interact with must have an infrared interface installed. Our thinking, however, is that someday such interfaces will be installed as a standard feature not only on PC's, but also on all types of consumer products and service systems. ATM's, kiosks, etc. could all be outfitted with a simple infrared interface at a very low cost, finally making such specialized computers accessible to the handicapped. This is now becoming a reality with the introduction of the Bluetooth protocol for wireless communication of electronic devices. [8]

4.0 FUTURE RESEARCH

There are two areas of potential future research we are interested in pursuing once Phase 3 of ViKI is completed. First is the development of a multitude of interfaces that would be of actual use to handicapped users. While our demonstration system virtual keyboard may be of some practical use, there is at least an argument to be made for experimenting with alternative keyboard layouts. The QWERTY layout, while ubiquitous and arguably well known to most users, is probably not the most efficient layout for many users. A strictly alphabetic layout might be preferable, or even something more radical such as a circular design based on digrams (2-letter combinations) or letter frequencies. With a virtual keyboard, it should be possible to easily investigate many types of specialized layouts.

The second area for future research would be in exploring two-way communication between the SIL system and the host. The current project is one-way communication only, from SIL to host. What might be accomplished if the communication were two-way? This is of particular interest for exploring systems for the vision impaired, since such users might not be able to read the screen of a device with which they wish to interact, such as an ATM machine. Could a duplex communication system be used to allow such users to have the host system's screen read to them using the reading technology housed in their personal interface?

6.0 ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. DUE-9551245. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the National Science Foundation. Special thanks to Rob Shneider for the development of the ViKI demonstration system.

REFERENCES

[1] Ark, W. S. and Selker, T. A look at human interaction with pervasive computers. *IBM Systems Journal.* Vol. 38, No. 4.

[2] Bass, L., Kasabach, C., Martin, R., Siewiorek, D., Smailagic, A., and Stivoric, J. (1997). The design of a wearable computer. *Proceedings of the Conference on Human Factors in Computing Systems (CHI 97),* March 1997. Atlanta, GA:139-146.

[3] Erickson, T. (1996) The design and long-term use of a personal electronic notebook: a reflective analysis. *Proceedings of the Conference on Human Factors in Computing Systems (CHI 96),* April 1996. Vancouver, British Columbia, Canada:11-18.

[4] Feiner, S. and Shamash, A. (1991) Hybrid user interfaces: breeding virtually bigger interfaces for physically smaller computers. *Proceedings of the ACM Symposium on User Interface Software and Technology*, November 1991. Hilton Head, South Carolina:9-17.

[5] Kawachiya, K. and Ishikawa, H. (1998) NaviPoint: an input device for mobile information browsing. *Factors in Computing Systems (CHI 98),* April 1998. Los Angeles, CA:1-8.

[6] Mann, S. (1996) Smart clothing: the shift to wearable computing. *Communications of the ACM*, August 1996:23-24.

[7] Miller, B., Bisdikian, C. (2000) *Bluetooth Revealed*. Prentice-Hall. 2000.

[8] Muller, M., Wharton, C., McIver, W., and Laux, L. (1997) Toward an HCI research and practice agenda based on human needs and social responsibility. *Factors in Computing Systems (CHI 97),* March 1997. Atlanta, GA:155-161.

[9] Robertson, S., Wharton, C., Ashworth, C., and Franzke, M. (1996) Dual device user interface design: PDAs and interactive television. *Proceedings of the Conference on Human Factors in Computing Systems (CHI 96),* April 1996. Vancouver, British Columbia, Canada:79-86.

[10] Smith, A., Dunaway, J., Demasco, P. Peischl, D. (1996) Multimodal input for computer access and augmentative communication. *Proceedings of the Second Annual ACM Conference on Assistive Technologies* (ASSETS '96), April 1996. Vancouver, British Columbia, Canada:80-85.

[11] Shneiderman, B., Between hope and fear: Universal access, medical records, and educational computing, *Communications of the ACM* 40, 2 (February 1997): 59-62.

[12] Warden, A., Walker, N., Bharat, K., and Hudson, S. (1997) Making computers easier for older adults to use: area cursors and sticky icons. *Factors in Computing Systems (CHI 97)*, March 1997. Atlanta, GA:266-271.