Using the scanning technique to make an ordinary operating system accessible to motor-impaired users. The "Autonomia" system.

Constantine E. Steriadis, Philip Constantinou
Mobile Radiocommunications Laboratory
National Technical University of Athens
P.O.Box 15773 Zografou, Athens, Greece
Tel: +30107723974, Fax: +30107723851
conster@mobile.ece.ntua.gr

ABSTRACT

In this paper we present the "Autonomia" system that aims to link a severely motor-impaired user to the information society, through ordinary means (ordinary operating system, applications, etc). The system aims to provide the user with the ability to make full use of any commercial application, whether it has been designed for disabled users or not. Another important feature of the presented system is that the user can take over the functioning of electrical and electronic appliances, and as a result, customize his environment.

Keywords

Assistive technology, User Interface design, Human-computer interaction, motor-impairments

INTRODUCTION

Many assistive systems are based upon a common personal computer, where a custom agent-application is active, awaiting for the user's input. For instance, in the system presented in [4], the agent-application takes over the control of the computer and it is responsible for the correct response to the user's input signals. The user interacts with the application's specially designed interface, but in most of the cases, cannot use common commercial software applications. As a result, the application's designer has to consume a lot of time to design new special applications that will cooperate with the application, and will be compatible with upgraded versions of popular applications (word-processors, spreadsheets, internet browsers, etc).

Autonomia is primarily addressed to severely motor-impaired users. By that term we refer to people who suffer from severe paralysis (quadriplegia) and due to that they are completely bed-ridden.
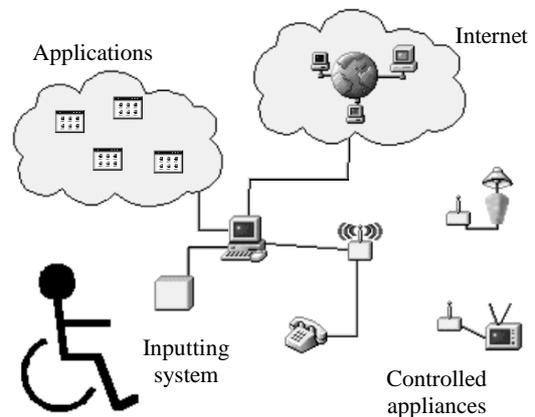

Figure 1 : Autonomia's architecture

Figure 1 depicts the architecture of the Autonomia system. The computer is the heart of the system, and enables the user to use any software application, access the Internet, and control the functionality of any electrical and electronic appliance.

DESIGN CONSIDERATIONS

In our effort to design a flexible and low-cost system, we have decided to build Autonomia for the popular Microsoft Windows^{TM} operating system (all versions latter to Windows'95^{TM} are supported), since, many commercial applications are available in the market for the particular OS.

The user of Autonomia virtually controls the functioning of the usual inputting devices (keyboard and mouse), through a simple sensoring device. The keyboard and mouse are input devices of many degrees of freedom according to the taxonomy presented in [3], and that is a fact that makes them totally useless for a paralysed person. We implement the simulation of those devices' functionality, by using a one-degree-of-freedom sensor (simple button, blow sensor, etc) combined to the scanning technique.

The Scanning Technique

The scanning technique is a rather simple algorithm for selecting from a group of items, with a single user action. For an average user the issue of interacting with the computer is analyzed into a two-step process: (i) focus the

input-element (physical button of the keyboard, graphical element on the screen) and (ii) perform the selection (click on the element). If the user has to select from a group of virtual buttons (graphical elements), with the use of the scanning technique, we periodically highlight each of these buttons [4]. If the desired button is highlighted then the user may perform an input, as long as it remains highlighted. This input is equivalent to a 'click', which is captured by the sensoring device, and it is transferred to the computer in a format that is recognizable for the computer (serial data).

In that way the individual can instruct the computer to perform mouse events (cursor movements, mouse clicks, drag&drop processes) and keyboard events (keystrokes). So, he can practically simulate the usage of a keyboard and a mouse.

USER INTERFACE

As denoted in the previous section, the user-interface part of the system is responsible for: (i) the correct translation of the user's single degree-of-freedom input into keyboard and mouse events and (ii) the secure transmission of the events to the computer. A very important issue is how we design and group the graphical elements, so that the interaction between the user and the system is characterized by efficiency, effectiveness, acceptance and satisfaction [2]. Therefore, in order to design our system, we have decided to take into account the following principles: (i) commonly used functionality must be easily accessible, (ii) the user should be able to customize the provided functionality, (iii) a pleasant graphical interface should be designed, and (iv) the visual elements of our application should occupy the less possible area on the display.

Based on those principles, Autonomia has been developed as a single window application (Figure 2).

Figure 2 : Autonomia's main window on the desktop

The application's window is placed on a corner of the screen and cannot be overlapped by other windows. The provided functionality is categorized into 'screens', which are different forms of the application's main window. A 'screen' is a 2D area where a group of graphical elements are placed in a specific order. Each graphical element is assigned with a unique task, and the scanning technique is

used to enable the user to select the desired element. Each time the user moves from one 'screen' to another, the main window's dimensions are rearranged, so as to present in a satisfactory way the current 'screen'.

The individual can manage the Windows OS with the use of 3 basic 'screens': (i) the cursor screen, (ii) the virtual-keyboard (VK) screen and (iii) the console screen. Figure 3 presents how the mentioned screens are interconnected:
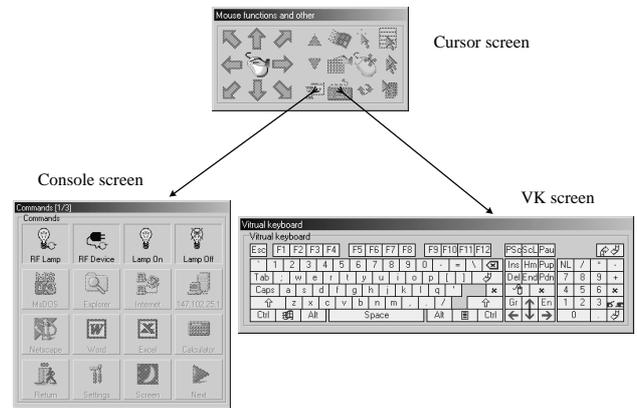

Figure 3 : The application's flow chart

The cursor screen is the starting screen and from there the individual can move to the other two screens.

The Cursor Screen

This is the screen that enables the user to instruct the movement of the cursor, and to implement a number of common mouse functions. A snapshot of the screen is illustrated in figure 4.
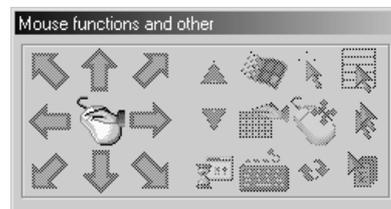

Figure 4 : A snapshot of the cursor screen

The graphical elements presented on the screen are highlighted in the numerical sequence shown in figure 5.
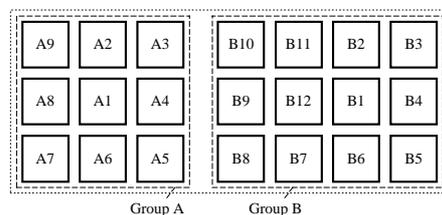

Figure 5 : Highlighting order in two groups

The graphical elements are organized in two groups (A and B). The first element of each group (A1 or B1) is used for switching from one group to the other.

Group A contains the elements that lead to the cursor's movement. The individual can move the cursor towards 8

directions (figure 4). In order to move the cursor towards a single direction, the user waits until the desired element is highlighted, and then he selects it. A timer is triggered and the cursor starts moving with a predefined speed (i.e. 5 pixels per second). When the cursor arrives to the desired position the user enters a click, and the cursor stops moving.

Group B contains the elements used to implement other mouse functions (clicks, wheel, drag&drop), as well as elements that lead to the VK screen (B7) or to the console screen (B8). For instance, if the user wants to perform a click at the display position (x,y), then he must switch to group A, move the cursor to (x,y), then switch to group B and select the B2 element.

The Virtual Keyboard (VK) Screen

In this screen a virtual keyboard (QWERTY keyboard format) is designed on the computer's display, as depicted in figure 6. The VK screen enables the user to perform keyboard events (keystrokes). The keystrokes are sent to the active application, in the same way as the keyboard is used for typing text.

Figure 6 : A snapshot of the VK screen

Due to the large amount of graphical elements (101 virtual keys), it would be very inconvenient for the user, if we implemented the scanning technique by highlighting all elements one-by-one. Therefore we have grouped the elements into 4 major groups (figure 7), depending on their location on the QWERTY keyboard.
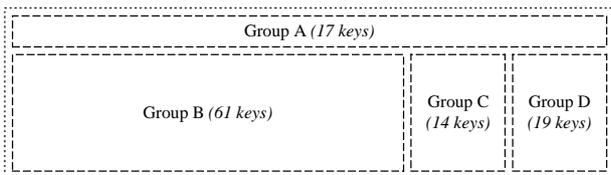

Figure 7 : Organizing the VK screen into groups

In order to select a specific key the individual has to first define the group where the desired key belongs, through the scanning technique, and after that to select the desired key.

For group A the keys are highlighted in left-to-right order, and when a selection is performed the application returns to the previous phase. If the user omits to produce an input for 4 sequential sweeps of the group's keys, then the application returns to the previous state.

In groups B, C, D the keys are organized into rows and columns. That means that the user must select a key in a two-step process: first the row and then the column. After the user has selected a key from the groups B, C or D the application remains in the selected group, making it easier

for the user to select a key of the same group. For instance, when the user is typing text, the application remains in Group B, where all the letters are located.

The user can exit the VK screen by selecting the key with the symbol ⌐ (Group C, row 3, column 1). The colors that appear on the VK screen can be customized according to the user's preferences.

The Console Screen

Through this screen the user may perform tasks such as starting other software applications, setting an electrical/electronic appliance on or off, etc.

Figure 8 : A snapshot of the console screen

Figure 8 depicts a snapshot of the screen. 16 virtual buttons are presented on the screen, and each one of them links to a predefined task, in general. The user can establish a dial-up connection or a phone-call. The button's selection, is done in 2 steps, first select the row and then select the column of the button.

The user can place up to 255 buttons on the screen, grouped in pages of dozens of buttons. A button can be a link to another group of buttons, as the menu-submenu structure. The four buttons of the last line are reserved for exiting the screen, changing the scanning frequency or moving in the pages.

The data regarding the configuration of each button (picture, label, linked process) is stored into a text file, and can be easily customized by the user.

Display's occupied space

An important issue for the user interface is how much of the total display is occupied by our application. The significant part of the display is what is outside the application's window, because our application has an assistive role, only. Therefore the majority of the display's space should be left uncovered, in order to be used by other applications. Table 1 presents the percentage of occupied space by our application, for the 3 basic screens, in 800x600 and 1024x768 desktop analysis.

Screen	Width	Height	Occupied space	
			800x600	1024x768
Cursor screen	252	134	7.0%	4.3%
VK screen	502	150	15.7%	9.6%
Console screen	312	301	19.6%	11.9%

Table 1 : Percentage of occupied desktop for each screen

THE INPUT SYSTEM

The input system, shown in figure 9, is a one-degree-of-freedom device used to capture the user's inputs. Depending on the level of the person's disability, the device should be designed to offer convenience and reliability.
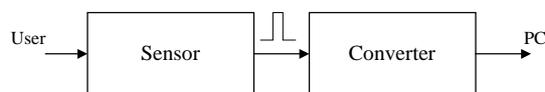

Figure 9 : The input system

The first part of the device is an electromechanical sensor that transforms a user's input into an electrical signal (pulse). By the term user's input we mean a slight movement, which the user can successfully perform. There should be no chance for the movement to be produced unconsciously. The sensor must not cause inconvenience to the disabled person, and must be designed for the specific user.

In the second part the electrical signal is transformed into serial data that is sent via an RS-232 port to the computer.

CONTROL OF THE ENVIRONMENT

Environment's control means to control a series of electrical and electronic appliances (lamps, TV, Hi-Fi, etc). A wireless switch is attached to each appliance, and receives instructions from an RF-transmitter (figure 1) for setting the appliance on or off. The system can control up to 255 appliances that are located in a range of 20 meters away from the transmitter.

EVALUATION

Our patient learned five and a half years ago that he had 'Amyothroph Lateral Sclerosis' (ALS). Today his legs and hands are paralyzed, and he lost his ability to speak about two years ago, due to his disease.

Our system was installed on a desktop computer with a conventional 15" monitor. The user's back is slightly raised with a pillow, so that he can watch the display. The input device is a simple button that has been placed between his knees. The simple button was a choice of the user because it was convenient for him to apply slight pressure between his knees, whenever input is needed.

Training and Usage

Our patient uses Autonomia on a daily base, for about 14 hours per day, for the last 5 months. He had almost no prior computer experience, and the basic training lasted about two hours. After the training procedure we fixed the scanning cycle time at 1 second.

He primarily writes in an ordinary word-processor, and plays games, such as chess, backgammon and card games. Through the system he can control the lights of his room and the functioning of a device that supports his respiration. He can also perform phone calls to his wife's mobile phone, in case of an emergency. We have recorded voice messages, such as 'food', 'help', 'toilet', etc, to express his needs. He can also perform dial-up connections to the Internet.

Good Overall Satisfaction

Our patient has expressed his gratitude and satisfaction for using the system. Among all features he is especially pleased to be able to write his thoughts on his own, compared to the previous situation, where he needed his wife to scan the alphabet, in order to build words and phrases. He has requested from us to install the system on a laptop computer, so that he could be able to transfer the system to another room, with the help of his wife, of course.

FUTURE WORK

Following the evaluation results we plan to introduce new features to the Autonomia system, such as mobility, speech synthesis in Greek and English, and to implement additional screens, that will help the individual to access several Internet services. The integration towards the BlueTooth™ wireless technology will also be considered.

Our future directions also include the realization of a series of evaluation sessions. We also, intend to make the system available in other languages, besides Greek and English.

REFERENCES

1. Akoumianakis, D. & Stephanidis, C. Supporting user adapted interface design: The USE-IT system, in *International Journal of Interacting with Computers, 9 (1), 1997.*

2. B.E.R. de Ruyter and J.H.M. de Vet Usability evaluations in user-centred design, in *IPO Annual Progress Report 32 1997*, 27-35.

3. Bleser T.W. An input model of interactive systems design. *Doctoral Dissertation, The George Washington Univ.* (1991).

4. Steriadis C., and Constantinou P. Telematics' application for people with special needs, in *Proceedings of the IMACS-CSC '98 Conference on Circuits, Systems and Computers.*

5. Steve Teixeira, Xavier Pacheco Delphi 5 Developer's Guide (Developer's Guide) *Macmillan USA Publishing ISBN: 0672317818*

ABOUT THE AUTHORS

Constantine E. Steriadis is a PhD candidate in the Mobile Radiocommunications Laboratory of the National Technical University of Athens (NTUA). His current interests include the design of a special environment for people with severe disabilities, telematic applications, and personal wireless communications. He received his Diploma in Electrical & Computer Engineering from the NTUA in 1997.

Dr. Philip Constantinou (Professor-NTUA) is the Director of the Mobile Radiocommunications Laboratory. His current research interests include personal communications, mobile satellite communications, interference problems on digital communications systems and assistive technology. He received his Diploma in Physics from the National University of Athens in 1972, the Master of Applied Science in Electrical Engineering from the University of Ottawa, Ontario, Canada in 1976 and the PhD degree in Electrical Engineering in 1983 from Carleton University, Ottawa, Ontario, Canada. In 1989 he joined the NTUA where he is currently a Professor.