Customizable Software Interface for Monitoring Applications

Manuel Merino, Isabel Gómez, Octavio Rivera, and Alberto J. Molina

Electronic Technology Department, University of Seville,
Avd. Reina Mercedes s/n,
41012, Seville, Spain
{Manmermon,Octavio}@dte.us.es {igomez,almolina}@us.es
http://matrix.dte.us.es/grupoimyt/

Abstract. In this paper we propose an application based on virtual keyboard and automatic scanning to communicate with a PC and the others people. The aim users are the people with disabilities. A high degree of customization is possible in the software. So the user can selected the color of buttons, position of system on screen, the kind of scanning, timer, the interface of communication, etc. Five people without disabilities tested our system. The results of the tests show the application reduce the fatigue of user and increased the text entry rate.

Keywords: interface, prediction system, control system, disability.

1 Introduction

Communication is a fundamental quality for human beings. People use different ways of Communication in their diary life. Communication skills are necessary in order to get a social integration. Nevertheless degenerative diseases, accidents, psychological disorders and others reason can make difficult the human communication. So, according to the Eurostat [1], the total population in Europe was 362 million in 1996 which 14.8% of the population between 6 and 64 years old have physical, psychological or sensorial disabilities. Therefore, assistive and adapted systems have to be developed. The aim of these ones is to help users in their diary tasks, letting them to live in a more comfortable way.

The development of a general purpose system is a complex task because every people have different skills and preferences. These systems have a degree of flexibility and customizable. However the limitations of these systems are heightened when people have severe disabilities.

Nowadays there are several tools and devices that allow disabled people to use specific applications. This paper describes a system located among three research main areas: Human-Computer Interface (HCI), Augmentative and Alternative Communication (AAC) y Assistive Technology (AT). A user can interact with this system using keyboard-handled software applications. An appropriate device which is adapted to user needs is only required to interact with it. The developed system can be handled with several kinds of input devices which increases the number of potential users. In our tests are performed using electromyography signal (EMG).

2 Related Work

Development of interfaces adapted to the user physical skills is necessary because the users can have different diseases. In most cases, some residual user movements are required to interact with these interfaces, like adapted switches or eye-tracker. Electric biosignals are used in some of these ones, such as electrooculography [2] or electromyography [2]. There exists another kind of interfaces that a user movement is not required to interact with it, like one is used in Brain Computer Interaction (BCI) [3]. This one is based on detecting some electric biosignals (electroencephalogram).

In addition to these input interfaces, an application to make possible the interaction between user and computer/PDA is necessary. Text entry is one of the most common features of this kind of applications. The main aim must be to reduce the numbers of user actions when users have some physical disabilities. To get this goal, prediction systems are used, like character prediction, word prediction or sentence prediction. Examples of this kind of applications are shown in [4, 5]. Both applications are virtual keyboards (VKs). First one [4] is a commercial application based on Windows Operative System. Different kinds of extended VKs can be selected but reduced VKs are not implemented. It uses an automatic scanning method and word prediction. On the other hand, the position of keys in the VK is not suitable to scan through column-rows or serial method. Prediction list length is setting to six words so it is not customizable. Configuration options (such as the color or size of buttons, position or size of application, etc.) cannot be changed by a handicapped.

Second one [5] is a fluctuating VK that uses a manual scanning method and character prediction. When a character is selected, the buttons around of this are changed based on the character prediction system. This property provokes the user fatigue is increased.

Dasher [6] is a different text input method using a character prediction. This one is based on a mouse pointer tracking system to select the wished character from a shown set. This is a text input system that allows a high rate character/minute. The selected text can be used by a voice synthesizer, but is not possible to send to another application. Also the operative system cannot be handled by this application.

3 Methodology

3.1 Software Description

The proposed application is an automatic scanning based VK using a prediction system. It runs on Windows XP and was implemented using Java 1.5. The prediction system requires a dictionary which is a database that was implemented using SQLite. Special features were implemented using C# 2.0.

Our software works like a daemon, it means that it is always visible in foreground and never catches the focus (fig. 1). Different input events can be generated by user to interact with the system. When an input event is received, it is processed by the Application Handler, and a set action is done depending on the state of the system. Computer applications can be handled by keyboard [7] and this software uses this fact. So, the selected action is send to Operative System like a real keystroke on a conventional keyboard, letting user to open/close computer applications, to send/receive e-mails, to browse the Internet, to enter texts, etc.

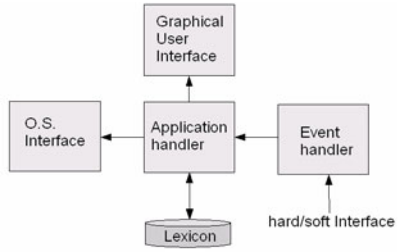

Fig. 1. Functional description

The Graphical User Interface (GUI) is composed of five different automatic scanning based VKs: a numeric VK (from 0 to 9), a punctuation mark VK, a command VK (including alt, shift, ctrl, enter, tab, f1, f2, etc., buttons), and an advanced command VK (this one can be customized setting on each button a sequence of commands, for example alt + f4, or alt + c), an application VK (is equivalent to start menu of Windows XP – fig. 3), and a letter VK (can be a reduced VK or an extend VK – fig. 2). Two alternatives have been implemented: a row-column scanning method or a serial scanning method.

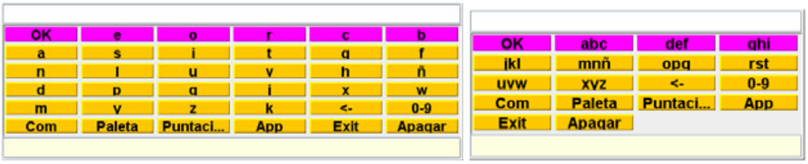

Fig. 2. GUI – Extend/reduced letter VK

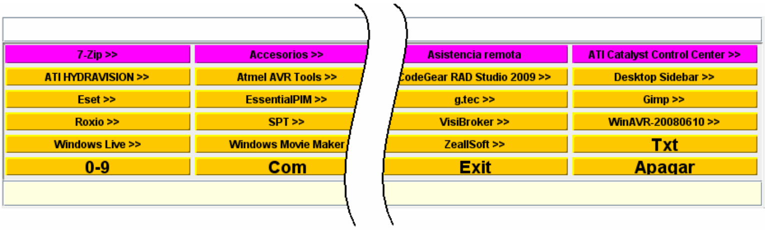

Fig. 3. GUI – Application VK

On the other hand, a personal VK can be defined by the user. This VK is based on an xml file and its structure is:

```
<teclado teclasPorFila="number of buttons in a row">
  <boton tipoBoton="button type" texto="text">
    <letra orden = "position"> letter </letra>
    ...
    <letra orden = "position"> letter </letra>
  </boton>
  <boton ...>
  ...
  </boton>
  ...
</teclado>
```

The maximum number of buttons in a row is indicated in the property *teclasPorFila* in *<teclado ...>...</teclado>*. The labels *<boton...>...</boton>* have two properties. One is *tipoBoton* that discriminates between letter, number, punctuation mark and command keys. The other property is *texto* that is used to descriptor text in the key. The last label is *<letra...>letter</letra>* where the *orden* indicates the position of the letter, number, command or punctuation mark.

Three text entry methods have been incorporated. One of them consists on a conventional text entry method. Characters of the wished word are entered one by one in this method. Other ones are word prediction systems. A study of different prediction systems is presented in [8, 9], in which we can check word prediction systems are the most commonly used. First of these ones is called NT and is only available when an ambiguous VK (more than one character per key) is used. NT is a modification of NT predictor system. When a key of VK is selected a new prediction list is made. In the new list, the words of the previous one whose characters match with those typed orderly so far are remained. Second one is called K-Word. In this method, first characters of the wished word have to be selected before a prediction is made. After that, K-Word predictor builds a list of suggestion using a k-gram algorithm with selected characters.

The user-application communication is realized by the user control interface. It has to be able to adapt dynamically to the user context. This dynamic adaptability and costumer of application are contained in the concept of plasticity [10]. Thus our application has been development to enable that the user personalizes the appearance of GUI, so the user can choose the color of scanned and no-scanned buttons, the color of button’s text, the size of buttons, the position of application in the screen, the scanned time, selected/no-select special VK, configure special VK, select and configure prediction system, select the type of scanned system, etc. A configure application allows to personalize the properties described before. The options of the configure application are automatic scanning to allow to person with disability can personalize the software. When an option is chosen a VK is shown which allows the user selects the new value of the actual option.

Otherwise, two log files are generated by the application. These files can be used in research tasks. The user events are registered by one of them. The prediction words are stored by the other log file. This saves the selected prediction word and its different lists of prediction, the time, the number of automatic reset of system and the number interaction by rows, columns and letters. The files allow study the advantages of a kind of VK and/or a prediction system.

3.2 User Control Interface

Three kinds of communication ports were designed: parallel/serial port and TCP/IP. In the future usb port will be added. These ports allow a lot different types of interface. At least an event has to be generated by the interface. Two different events are allowed. One is the selection event to select current option (mandatory) and the other is the change event to advance to next option (optional). These events are evaluated in [11, 12]. Events are distinguished based on time. The parallel port was tested by a stick switch based on the select event. The serial port was used by a button switch based on selection and change events and a microphone based on selection event. The TCP/IP was tested by EMG interface. In addition the system could be handled by control and/or shift button of conventional keyboard.

Any interface that generates input events based on user actions can be used to interact with this software. At the moment, it has been tested using different switches and an electromyography based device. The last one uses three electrodes placed on the user's arm. Hence, a movement of his/her hand can be detected and an adequate event can be generated. The signal processing procedure splits the electromyography signal on non-overlapping windows that are 100 samples width. An absolute value operation is applied on each window and the result is integrated. The signal is interpreted as a selection when obtained value is above a set threshold.

3.3 Procedure

Some trials focused on text input system have been done. Five users without disabilities between 25 and 35 years old participated in these trials. They had not used the application before. Results using conventional text entry, K-Word and NT prediction system are compared between each in order to obtain time savings. Also penalization time due to user errors is calculated. Maximum prediction list length is set to 14 words. The list of suggestions in K-Word predictor is displayed when user has selected three first characters of the wished word. The main objective of these trials is to entry text without mistakes. The application implements a functionality that let user to fix errors. Some tests with handicapped people must be done in the future.

4 Work and Results

The trial results were stored in two log files. Time measurements included increments due to penalization sources. These sources were: wrong character selections (in this case, time was required to set them); wrong word selections from prediction list; selections using more than one user input (users was not able to choose the desired row, column, letter or word); reset of system due to time out, etc. Besides time measurements, information about different types of mistakes can be found in the aforementioned log files.

Trial results are shown in tables 1 and 2. The table 2 shows the gain using the predictor systems. The K-Word predictor gain about text entry conventional is 27.58%, which means a time saving of 192.537 seconds. So, the gain using NT predictor is 52.02% about text entry conventional, which is 353.04 seconds. On the other hand, NT predictor is 34.36% faster than K-Word.

Table 1. Times in text entry systems

User	KWord	NT	Conventional
1	*475.05*	*318.79*	*572.68*
2	*501.21*	*356.38*	*580.48*
3	*357.15*	*238.14*	*522.87*
4	*512.76*	*335.20*	*868.18*
5	*501.02*	*296.15*	*765.65*

Table 2. Gains of text entry systems

User	KWord vs. Conventional (%)	NT vs. KWord(%)	NT vs. Conventional (%)
1	*17.05*	*33.06*	*44.33*
2	*13.66*	*29.90*	*38.61*
3	*31.70*	*33.32*	*54.46*
4	*40.94*	*34.63*	*61.39*
5	*34.56*	*40.89*	*61.32*

5 Conclusion and Planned Lines of Research

It has been presented an application that is easy to use and it does not require any training. The operative system can be handled, and text can be written. It must be mentioned that users must have knowledge in handling computer using control keys of a standard keyboard. So, it is not operating system dependent, because it is based on Java.

All its possibilities are included in graphical interface that is shown permanently on the computer screen. The design of this application is flexible in respect of main keyboard displayed on screen and the used input signal. It is possible to change among several configurations of keyboard during execution time. Also, three different prediction systems have been implemented. Furthermore, the inputs which handle the software come from several devices (parallel port, serial port or TCP/IP).

On the other hand, the application is a research tool. Two log files are made. They both store information about a using session. Therefore, different kinds of results can be extracted, such as penalize due to wrong character selections, wrong word selections from prediction list, selections using more than one user input (users was not able to choose the desired row, column, letter or word), reset of system due to time out, etc.

Acknowledgements

This project has been carried out within the framework of a research program: (p08-TIC-3631) – Multimodal Wireless interface funded by the Regional Government of Andalusia.

References

1. Eurostat. Health Statistics. Luxembrourg: Office for Official Publications of the European communities (2002) ISBN: 92-894-3730-8, http://epp.eurostat.ec.europa.eu (Retrieved on December 2009)
2. Dhillon, H.S., Singla, R., Rekhi, N.S., Rameshwar, J.: EOG and EMG Based Virtual Keyboard: A Brain-Computer Interface. In: 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2009 (2009)
3. Nam, C.S., Jeon, Y., Li, Y., Kim, Y.-J., Yoon, H.-Y.: Usability of the P300 Speller: Towards a More Sustainable Brain-Computer Interface. International Journal on Human-Computer Interaction 1(5) (2009)
4. Wivik on-screen keyboard software, http://www.wivik.com/ (Retrieve on December 2009)
5. Merlin, B., Raynal, M.: Increasing Software Keyboard Key by Recycling Needless Ones. In: Assistive Technology from Adapted Equipment to Inclusive Environments - AAATE 2009, vol. 25, pp. 139–143 (2009) ISBN: 978-1-60750-042-1
6. Ward, D.J., Blackwell, A.F., MacKay, D.J.C.: Dasher - a Data Entry Interface Using Continuous Gestures and Language Models. In: UIST 2000 (2000), http://www.inference.phy.cam.ac.uk/dasher (Retrieve on December 2009)
7. Gómez González, I.M., Molina Cantero, A.J., Romero, O.R.: Multi Purpose Interface for Handling General Computer Applications. In: 9th European Conference for the Advancement of Assistive Technology in Europe (AAATE 2007), San Sebastián, España, pp. 7–8 (2007)
8. Garay-Vitoria, N., Abascal, J.: Text prediction: a survey. Univ. Access Inf. Soc. 4, 188–203 (2006)
9. Rivera, O., Molina, A.J., Gómez, I., Sánchez, G.: Evaluation of Unambiguous Virtual Keyboards with Character Prediction. In: Conference of the Association for the Advance of Assistive Technology in Europe (AAATE 2009), Florence, Italy, vol. 25, pp. 144–149 (2009) ISBN: 978-1-60750-042-1
10. Sendín, M., López Juan, M.: Contributions of Dochtomic View of plasticity to seamlessly embed accessibility and adaptivity support in user interfaces. In: Advances in Engineering Software, pp. 1261–1270 (2009)
11. Rivera, O., Molina, A.J., Gómez, I., Merino, M.: A Study of Two-inputs Scanning Methods to Enhance the Communication Rate. In: Conference of the Association for the Advance of Assistive Technology in Europe (AAATE 2009), Florence, Italy, vol. 25, pp. 132–137 (2009) ISBN: 978-1-60750-042-1
12. Rivera, O., Molina, A.J., Gómez, I.: Two-Input Based Procedures Using a Single Device. In: Conference of the Association for the Advance of Assistive Technology in Europe (AAATE 2009), Florence, Italy, vol. 25, p. 881 (2009) ISBN: 978-1-60750-042-1