ISSN:2348-2079

Volume-6 Issue-2

International Journal of Intellectual Advancements and Research in Engineering Computations

Human Computer Interaction for Visually impaired People

[1]R.Deepan [2]S.Ramya [3]S.Praveen Raj [4]V.Manimaran

[1-3] UG Scholars, Department of Computer Science and Engineering, Nandha Engineering College, Erode

[4]Assistant Professor, Department of Computer Science and Engineering, Nandha Engineering College, Erode

***Abstract*— This project handles two core speech technologies "Speech Synthesis" and "Speech Recognition" that are supported through the Java Speech API. Our project requires a "Speech engine" designed to deal with either speech input or speech output. Speech synthesizer and speech recognizer are both speech engine instances. The Java Speech API defines a standard, easy-to-use, cross-platform software interface to state-of-the-art speech technology. Speech recognition provides computers with the ability to listen to spoken language and to determine what has been said. In other words, Speech synthesis provides the reverse process of producing synthetic speech from text generated by an application, an applet or a user. It is often referred to as text-to-speech technology. More time and effort has been given to increase user interactivity with computer using mouse and speech synthesis. Operation like media player, text reader, file search etc. have been implemented to give the application a look as of an O.S. for blinds. A GUI has been developed, that works same as windows GUI. Each window will have some buttons, each for a unique task. As mouse comes over the button, the synthesizer will speak that mouse is over this button, and what will happen if he performs some action with mouse.Full use of all the functionalities of mouse has been incorporated to make the application easier to use, just using one or two peripherals.**

Index Terms: **Usability, Text -To-Speech, Open Source, Graphical User Interfaces, Computer Interfaces.**

I. INTRODUCTION

The Computer usage today is predominantly based on graphical interaction, where the visual presentation of information is essential both for input and output. This type of interaction can present problems if the user for some reason has difficulties accessing these modalities. These difficulties include mobility impairments, making hand movements hard, and visual impairments making visual presentation of information limited or impossible, Many blind and visually impaired people use a *screen reader* to access the contents of the computer screen. This is a piece of software that reads the contents of the screen and presents it using synthetic speech or Braille. For example, when browsing a desktop with files and folders, the user steps through the objects line by line, getting their names read out by a speech. In order to access commands, menus and make simple mouse emulation, the keyboard is used. The major difference between a screen reader and a graphical user interface is the way the information is presented. Plain text is a linear or one-dimensional way of presenting information, information that in the graphical user interface is two or three-dimensional. For example in a desktop file system, like Mac OS or Windows, the files and folders are placed on a two dimensional surface, and different windows could be placed on this surface also, both next to each other and partly or totally covering each other. When presenting this information using speech synthesis or Braille, this two or three-dimensional information is projected into one dimension. Important spatial information such as grouping and spatial relationships gets lost in this transfer when everything has to be presented in a linear fashion. This is the data structure that the screen reader has to maintain keeping all the information the screen contains. It contains information about what characters are written where on the screen, the different controls that are used and the grouping and the spatial relationship between the different objects. The problem with maintaining this model is that it could become outdated if the screen updates without updating the OSM. Furthermore, the information in the OSM could be insufficient if the application uses non-standardized controls like a picture instead of a button or presents text that is included in an image instead of using the built in functions in the operating system for writing text. In this case, there is no

way for the screen reader to recognize that the image is a button or even a string of characters, and the user will miss it entirely. Even though the present development of accessibility in modern operating systems has made it easier for screen readers to access the screen contents, there are still problems in representing what is presented on the screen. The project is partly speech command based.

II. HUMAN-COMPUTER INTERACTION FOR Blind Computer Users: Starting Point

To overcome the difficulties in existing system here are implementing the following modules to provide speech interactive Operating System that allows two-way speech interaction. Here we have implemented speech based interface in MS Windows by using speech engine. Speech synthesizer and speech recognizer are both speech engine instances. The project is partly speech command based (limited in scope). More time and effort has been given to increase user interactivity with computer using mouse and speech synthesis. Utilities like media player, text reader, file search etc. have been implemented to give the application a look as of an O.S. for blinds. A GUI has been developed, that works same as windows GUI. Each window will have some buttons, each for a unique task. As mouse comes over the button, the synthesizer will speak that mouse is over this button, and what will happen if he performs some action with mouse. The idea of GUI is same as Braille code. The user will come to know about the context he is in, and the state in which the system is. This will be achieved by notifying user, via speech, about the state of machine, user, and context. Full use of all the functionalities of mouse has been incorporated to make the application easier to use, just using one or two peripherals. Left click performs action associated with each button and scrolling scroll button will give the current position of mouse pointer.

III. OBJECTIVES

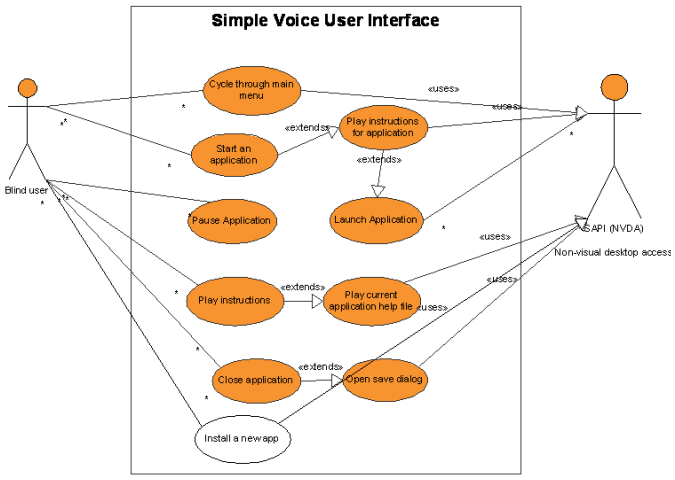
1. The paper is partly speech command based operating system. More time and effort has been given to increase user interactivity with computer using mouse and speech synthesis.Utilities like media player, text reader, file search etc. have been implemented to give the application a look as of an O.S. for blinds.
2. A GUI has been developed, that works same as windows GUI. Each window will have some buttons, each for a unique task. As mouse comes over the button, the synthesizer will speak that mouse is over this button, and what will happen if he performs some action with mouse.
3. The idea of GUI is same as Braille code. The user will come to know about the context he is in, and the state in which the system is. This will be achieved by notifying user, via speech, about the state of machine, user, and context.
4. Full use of all the functionalities of mouse has been incorporated to make the application easier to use, just using one or two peripherals.
5. Left click performs action associated with each button and scrolling scroll button will give the current position of mouse pointer.

IV. SOFTWARE DEVELOPMENT PROCESS

This application gives a new idea for a desktop interface for blind users using very simple techniques. Introduction of mouse, touchpad, touch screen and GUI based operating systems made the human-computer interaction a lot easier. Now even a kid could work with computers. Introduction of graphics, multimedia and sound in computers made it very much interactive. A new generation of computers had started. But in all these stage, no concern to those people were given for whom the GUI, colored screens, monitor, mouse and keyboard doesn't matter. Because they can't either see them or work on them. Until fourth stage, no such interface to computers was developed that helped visually challenged people and handicapped persons to interact with computer by themselves. So now comes the idea of a speech based Operating System that is fully operated via speech, and that can be operated easily by any blind person.

The javax.speech package of the Java Speech API defines an abstract software representation of a speech engine. "Speech engine" is the generic term for a system designed to deal with either speech input or speech output. Speech synthesizers and speech recognizers are both speech engine instances.

The javax.speech.synthesis package and javax.speech.recognition package extend and augment the basic functionality to define the specific capabilities of speech synthesizers and speech recognizers. The Java Speech API



makes only one assumption about the implementation of a JSAPI engine: that it provides a true implementation of the Java classes and interfaces defined by the API. The Java Speech API defines a standard, easy-to-use, cross-platform software interface to state-of-the-art speech technology. Two core speech technologies are supported through the Java Speech API: speech recognition and speech synthesis. Speech recognition provides computers with the ability to listen to spoken language and to determine what has been said. In other words, it processes audio input containing speech by converting it to text. Speech synthesis provides the reverse process of producing synthetic speech from text generated by an application, an applet or a user. It is often referred to as text-to-speech technology.

The trainer then looks into a dictionary which maps every word to a sequence of sound units, to derive the sequence of sound units associated with each signal. Thus, in addition to the speech signals, you will also be given a set of transcripts for the database (in a single file) and two dictionaries, one in which legitimate words in the language are mapped sequences of sound units (or sub-word units), and another in which non-speech sounds are mapped to corresponding non-speech or speech-like sound units. It will refer to the former as the language dictionary and the latter as the filler dictionary.

The incremental and iterative development type of development process was chosen since this type of project of interface design is based on usability and heuristic process that cannot be measured precisely with only decision matrixes, only by constant feedback and reevaluation can good results be obtained. Each process must be developed, tested and improved. The basic idea behind iterative enhancement is to develop a software system incrementally, allowing the developer to take advantage of what was being learned during the development of earlier, incremental, deliverable versions of the system. Learning comes from both the development and use of the system, where possible. Key steps in the process were to start with a simple implementation of a subset of the software requirements and iteratively enhance the evolving sequence of versions until the full system is implemented.

V. System Analysis

A. Study On Existing System

In Existing system there are many Speech based technologies which are based on one-way speech interaction, with which a blind user cannot be sure whether the required task has been performed or not. And hence this application is of no use as the Operating System is not speech interactive. Even there is no way for the blind users to perform major file based operations.

Disadvantages:

1. This project deals with the problem of interaction between blind people and computer.

2. Deals with the problem of absence of a platform (Operating System) through which a blind person can work on computer.

3. Speech technology has been introduced recently but speech based applications are of no use because the platform (Operating System) is not speech interactive.

4. The main problem is lack of a speech based file management system by which a blind person can at least work on file system.

5. By this application, it want to generate an efficient interface between a blind user and computer.

B. Study On Proposed System

To overcome the difficulties in existing system here are implementing the following modules to provide speech interactive Operating System that allows two-way speech interaction.

1. Windows incorporated in the GUI.
2. Explorer window.
3. File operation window.
4. Media player.
5. System utilities.
6. Speech recognizer.

we have implemented speech based interface in MS Windows by using speech engine. Speech synthesizer and speech recognizer are both speech engine instances. The project is partly speech command based (limited in scope). More time and effort has been given to increase user interactivity with computer using mouse and speech synthesis. Utilities like media player, text reader, file search etc. have been implemented to give the application a look as of an O.S. for blinds. A GUI has been developed, that works same as windows GUI. Each window will have some buttons, each for a unique task. As mouse comes over the button, the synthesizer will speak that mouse is over this button, and what will happen if he performs some action with mouse. The idea of GUI is same as Braille code. The user will come to know about the context he is in, and the state in which the system is. This will be achieved by notifying user, via speech, about the state of machine, user, and context. Full use of all the functionalities of mouse has been incorporated to make the application easier to use, just using one or two peripherals. Left click performs action associated with each button and scrolling scroll button will give the current position of mouse pointer.

VI. CONCLUSION

With a good cooperation and kind response of our professors and our team mates here have come to the end of this project successfully. With our immense efforts and dedication it completed this project in a period of six months. As all the software's used in our project are open source it is less cost effective. Our project deals with the introduction of speech based Operating System with which the user can interact with the system by the easiest way of communication, mainly the "speech". As the major concern of our project is about blind users it favors the people who have no concern about the recent technological developments like GUI, color screens and monitors.

REFERENCES

[1] Bach-y-Rita P, Collins CC, Saunders F, White B, Scadden L. Vision substitution by tactile image projection. Nature 221, (1969) 963-964

[2] Boyd, L.H., Boyd, W.L. and Vanderheiden, G.C. The graphical user interface: Crisis, danger and opportunity. In J. of Visual Impairment and Blindness, Dec. (1990) 496-502

[3] Hiroaki Takahashi, et al. Electro-Tactile Display with Localized High-Speed Switching. In Proc. of ICAT 2002, Tokyo, Japan (2002) 10-16

[4] International Community for Auditory Display. Website 2006 http://www.icad. org/

[5] Hiroyuki Kajimoto , Masahiko Inami , Naoki Kawakami , Susumu Tachi, SmartTouch - Augmentation of Skin Sensation with Electrocutaneous Display, Proceedings of the 11th Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems , March 22-23, 2003

6] Vincent Lévesque , Jérôme Pasquero , Vincent Hayward , Maryse Legault, Display of virtual braille dots by lateral skin deformation: feasibility study, ACM Transactions on Applied Perception , v.2 n.2, p.132-149, April 2005

[7] Sharon Oviatt, Ten myths of multimodal interaction, Communications of the ACM, v.42 n.11, p.74-81, Nov. 1999

[8] Richter, K. and Enge, M. Multi-modal Framework to support users with special needs in interaction with public information systems. In Proc. of Mobile HCI, Udine, Italy, Springer, LNCS, 2795, (2003) 286-301

[9] Anthony Savidis , Constantine Stephanidis, Developing dual user interfaces for integrating blind and sighted users: the HOMER UIMS, Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, p.106-113, May 07-11, 1995, Denver, Colorado, USA

[10] Savidis, A., Stephanidis, C. Unified user interface design: designing universally accessible interactions. Interacting with Computers 16(2): 243-270 (2004)

[11] Tang, H. and Beebe, D.J. An Oral Tactile Interface for Blind Navigation. IEEE Trans. on Neural Systems and Rehab. Engineering, 14(1), (2006)

[12] Charles Wiecha , Pedro Szekely, Transforming the UI for anyone. anywhere: enabling an increased variety of users, devices, and tasks through interface transformations, CHI '01 Extended Abstracts on Human Factors in Computing Systems, March 31-April 05, 2001, Seattle, Washington

[13] W3C, WAI-ARIA Overview. W3C-WAI 09 2010: http://www.w3.org/WAI/intro/aria.

[14] [ARIA-PRACTICES] WAI-ARIA Authoring Practices. J. Scheuhammer, M. Cooper, L. Pappas, R. Schwerdtfeger, Editors, W3C Working Draft.

[15] Prof Dr. Helmut Krueger, Clausiusstr, Swiss Federal Institute of Technology. Zurich "Introduction to Human-Computer Interaction" 1998

[16] WAI-ARIA 1.0 Primer. W3C Working Draft 16 September 2010: http://www.w3.org/TR/wai-aria-primer/