Design of Voice based PC Control for Blind people

Mr. Vikrant A. Agaskar*, Mr. Dhruv Shah,**
Mr. Viral Shah, Mr. Amrut Shenoy**, Mr. Ankit Vora****

***(** Asst. Professor, Department of Computer Engineering, VCET, Vasai Road(West),
Thane- 401202. Email:vikrantagaskar@hotmail.com).
**(Department of Computer Engineering, VCET, Vasai road(West),
Thane-401202. Email:amrutshn@gmail.com).

ABSTRACT

Development of graphical interfaces and visual programming concepts has created serious problems for the blinds regarding usage of computer and visualization. Through this paper we would like to empower the blind people by providing them a tool to perform the basic task on their computer. The main aim behind this paper is to create voice enabled PC control software that will help them to use their voice. Screen reader is the approach that we intend to use for presenting graphical desktop applications in audio. The user listens to and interacts solely with the display, never with the underlined graphical interfaces. In today's world when everything is driven by technology, it is necessary that we use this technology for the less privileged people.

Keywords- **Screen Reading, Speech Recognition and Speech Synthesis.**

1. INTRODUCTION

Empowering the disabled is the current need of the hour. With technology rapidly expanding and the size of the computing devices shrinking day by day, it is absolutely necessary to ensure that the disabled sections of people are not left behind in the entire process. The solution to this lies in the form of our design that we aim to implement. This helps them to activate applications on their screen by using the microphone to give voice commands. Whatever is displayed on the screen would be read out to the person so that he gets the same feel of the screen as a person who views it with his eyes. By doing so we can achieve two goals: Firstly, eliminate the need to use the expensive Braille displays and secondly, make the entire experience engaging and comforting.

2. SCREEN READER

A screen reader is a software application that attempts to identify and interpret what is being displayed on the screen. This interpretation is then represented to the user with text-to-speech, sound icons or a Braille output device. Screen readers are a form of assistive technology (AT) potentially useful to people who are blind, visually impaired or learning disabled.

2.1 TRADITONAL TECHNOLOGY:

MSAA (Microsoft Active Accessibility) [1]

There are two main mechanisms to be aware of when it comes to MSAA, the MSAA event model and the eye accessible COM interface. The MSAA events are the mechanism by which the application/control notifies the system that an event has happened pertaining to the control. AT's can determine when these events have happened and respond by notifying the user.

MSAA DOM for an application is a tree structured group of elements that defines the accessible elements for an application. This means that each control that can be used by the end user should show up in the DOM.

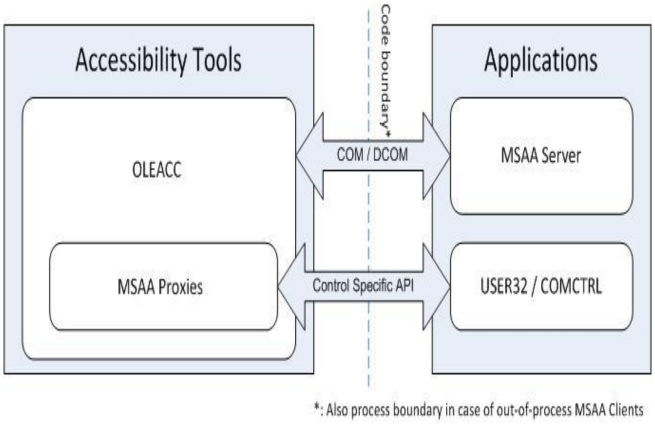

Fig. 1 : MSAA

2.2 CURRENT TECHNOLOGY: UI AUTOMATION [1]

MSAA represents individual UI elements as accessible objects and UI automation represents them as automation elements. Both refer to the accessibility tool or software automation program as the client.

MSAA offers a single COM interface with a fixed small set of properties. UI automation offers a richer set of properties as well as an extended set of interfaces called control patterns to manipulate automation elements.

2.2.1 UI AUTOMATION ARCHITECTURE

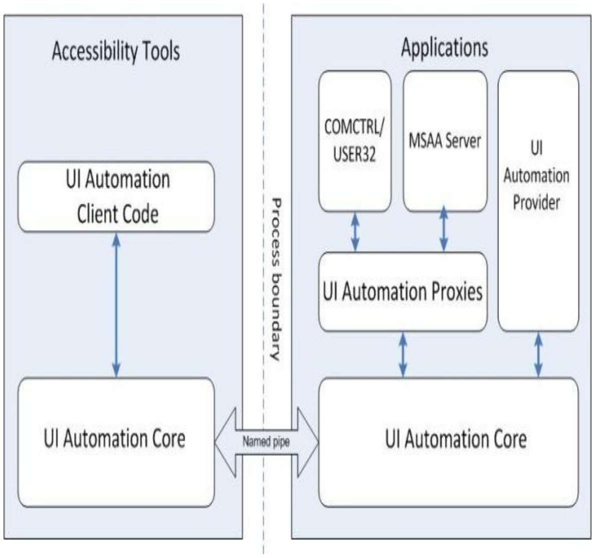

Fig. 2 UI AUTOMATION

As shown in Fig. 2, UI Automation loads the UI Automation code component into both the accessibility tools and application processes. This component manages cross-process communication and provides higher level services.

3. DESIGN

3.1. OBJECTIVE

Provide a means for the blind people to use the computer, since technology is rapidly on the move and they too need to be in synchronization with the technology.

3.2 BASIC DESIGN

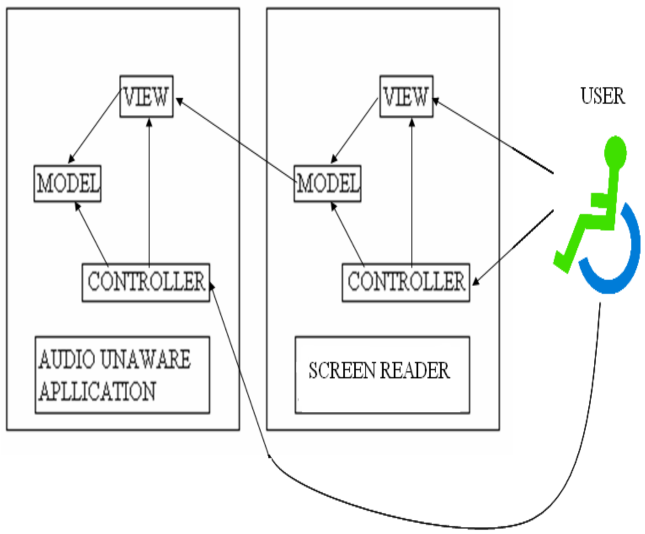

Fig. 3 Application Design

3.2.1 SPEECH RECOGNITION

Every speech recognition application consists of:

- An engine that translates waves into text
- A list of speech commands

A. Working

The initial state is in the "deactivate" state, which means that the program is in a sleepy state. After the command "activate" we will wake up the program and start recognizing other commands.

One of the important states in the program is the "menu" state, meaning that if a program is running one can say

"menu" to hook all the menu items and start using them. For example, if a person is running notepad he/she could open new file by saying *"menu"->"File"->"New File"*.

One of the problems with voice activated systems is what happens if one does not know exactly which commands the computer expects. No problem! If a person is unable to precede just say "commands list" and the program will give him/her what are the available commands.

B. Voice Training

To start talking one should do the following two steps:

- The first thing to do is adjust the microphone by clicking the right mouse button and choosing the "Mic Training Wizard"
- The second thing to do is training the engine to your voice by choosing "User training Wizard".

3.2.2 SCREEN READING

The working of screen reader can be broken down into five steps:

- Generating the content view UI Automation tree for the desktop with desktop as the root.
- Extracting the name, type and summary of all icons on the desktop from the UI Automation tree.
- Speech synthesizing of the text being generated.
- Introducing ambience sound, Ear cons to the synthesized speech.
- Pass the audio output to the output agent.

UI AUTOMATION [1]

- It comes under Windows API 3.0
- An object model called as the accessible object model (AOM) is created that makes it

easy for client applications to receive events, retrieve property values and manipulate UI elements.

- Elements are exposed as a tree structure, with the desktop as the root element.
- The common properties of the UI elements are name, id, state, type of the application on the desktop to be read.

A. TREE VIEWS

- Raw view: The full tree of automation element objects for which the desktop is the root.
- Control view: A subset of the raw view that closely maps to the UI structure as the user perceives it.
- Content view: A subset of the control view that contains the content most relevant to the user like the values in a dropdown combo box.

3.2.3 SPEECH SYNTHESIS

Speech synthesis is achieved using the following packages

A. *pyTTS* [6]

pyTTS is a Python wrapper for the Microsoft Speech API (SAPI) that enables speech synthesis. The output manager component pipes all output from pyTTS through pySonic a Python wrapper for the FMOD sound library to enable mixed, 3D sound and speech specialization.

B. *pyAA* [6]

The Microsoft Active Accessibility library provides programmers with runtime interface inspection and control capabilities on the Windows platform. pyAA wraps this library for use in python by developers who wish to create automated GUI tests or to derive information for assistive technologies from visual interfaces of other applications. This library is the conduit that is used to automate actions on graphical interfaces while maintaining a user facing auditory display composed of tasks.

C. *pyhook* [6]

It is a python wrapper for hooking mechanism in Windows which is used to capture all external device inputs and to communicate with various COM and DCOM threads.

D. *MinGW GCC 2.0.3* [2]

MinGW (Minimalist GNU for Windows), formerly mingw32, is a native software port of the GNU Compiler Collection(GCC) and GNU Binutils for use in the development of native Windows applications; MinGW can function either as a cross compiler targeting Windows or as a native tool chain run on Windows itself. It is developed along with a set of freely distributable import libraries and header files for the Windows API.

E. *SWIG 2.0.2* [2]

SWIG(Simplified wrapper and interface generator) is an open source software tool used to connect computer programs or libraries written in C or C++ with scripting languages such as Lua, Perl, PHP, Python, R, Ruby, Tcl and other languages like C#, JAVA, octave. Output can also be in the form of XML and Lisp S-expressions.

The aim is to allow calling native functions (C/C++) by interpreted code, passing complex data types, keeping memory from being freed, and inheriting classes across languages etc. The programmer writes an interface file containing a list of C/C++ functions to be visible to an interpreter. SWIG will compile the interface file into a regular C/C++. SWIG will generate conversion code for functions with simple arguments; conversion code for complex types of arguments has to be written by a programmer. SWIG tool creates source code which provides the glue between C/C++ and the target language. Depending on the language, this glue comes in two forms: a shared library that an extent interpreter can link to as some form of extension module, or a shared library that can be linked to other programs compiled in target language.

F. Py2exe [3]

Py2exe turns Python programs into packages that can be run on other Windows computer without installing Python. Python is needed on the computer where py2exe itself is run because py2exe is a Python program and it includes parts of Python in the package that is built.

4. CONCLUSION

Adapting GUI applications to an auditory display that describes user tasks via multiple streams of specialized speech and sound, provides a better user experience than a single stream narration of the screen. Such an approach can improve the awareness and effectiveness of people with visual impairments working in a computing environment.

REFERENCES

1. MSDN
2. Microsoft Foundation Class Library
3. www.python.org
4. www.speechcomputing.com
5. Microsoft Win32 DLL's
6. PySonic Library