



Controlling Device through Speech Recognition System

Vishakha Karpe, Shilpa Kabadi, Asmita Mutgekar, Manisha Pokharkar

Department of Computer Engg, Pune
India

Abstract— *In the modern era, mouse control has become an important part of human computer interaction. Speech Recognition is the process of automatically recognizing a certain word spoken by a particular speaker based on individual information included in speech waves. This technique makes it possible to use the speaker's voice to verify his/her identity and provide controlled access to various services. Speech recognition provides computers with the ability to listen to spoken language and to determine what has been said. Using speech recognition we can give commands to computer and the computer will perform the given task. The main objective of this research is to construct and develop system to execute commands of operating system by using speech recognition system that is capable of recognizing and responding to speech inputs rather than using traditional means of input (e.g. computer keyboard, mouse), this will lead to save time and reduce effort by the user. This is of great importance to increase the interaction between people and computers by using speech recognition; especially for whom suffer from health problems, for example, persons with disabilities from the movement, this technology helps physically challenged skilled persons. This application will help in reduction in hardware requirement and can be implemented in other electronic devices also.*

Keywords— *Speech Recognition, Speech Synthesis, Phoneme recognition, JSAPI, JSGF*

I. INTRODUCTION

Speech technology is a very popular term right now. It can be divided into two categories: speech recognition and speech synthesis. They have both been popular research subjects for over four decades. Speech recognition is highly demanded and has many useful applications. The speed of typing and handwriting is usually one word per second, so speaking may be the fastest communication form with a computer. This research is concerned with speech recognition technology, which is part of speech and signal processing, as well as human computer interaction (HCI).

Voice recognition is used to map a voice command with its corresponding action. This is brought about by converting speech to text. The program matches the input voice with the voice on which it is trained and maps it to the best possible result. The speech recognition engines are responsible for converting acoustical signals to digital signals, and then to text. The speech synthesizer engines are responsible for converting text to a spoken language. This process first breaks the words into phonemes, which are then transformed to a digital audio signal for playback.

The applications with voice recognition can also be a very helpful tool for handicapped people who have difficulties with typing. The Sphinx-4 architecture has been designed for modularity; any module in the system can be smoothly exchanged for another without requiring any modification of the other modules.

Two modes of speech recognition are available:

- Dictation: Users read data directly into a microphone. The range of words the engine can recognize is limited to the recognizers, grammar, or dictionary of recognizable words.
- Command and control: Users speak commands or ask questions. The range of words the engine can recognize in this case is usually defined by a limited grammar. This mode often eliminates the need to “train” the recognizers.

In this research we will concern with the second mode of speech recognition SR to convert an acoustic signal, captured by a microphone to a set of words. The recognized words will be executed as commands of operating system.

II. BASIC PROTOTYPE OF SPEECH RECOGNITION SYSTEM

This application utilizes the “Command & Control” aspect of speech recognition, that means must specify a grammar file that contains all the words or names of commands. The format of this file called Java Speech Grammar Format (JSGF). After the grammar file need to create a configuration file for the recognizer. The configuration file already exists with sphinx4 system is used to define names and types of all the components of sphinx4 system and to the connectivity of these components and configuration of each of them. In this research can achieve various commands such as (open, note pad, control panel.....) as sample prototype to implement this application.

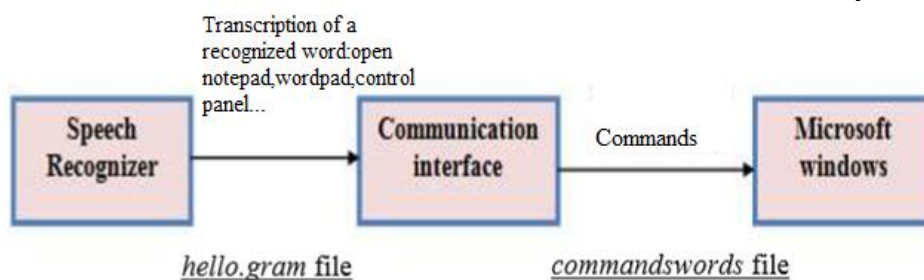


Fig. 1 Components of Speech Recognition System

The perfect solution to this application by using advance technology sphinx4 that is a speech recognition system written entirely in the Java(TM) programming language to add new module of command by create text file named commandswords.txt that contains all the words and commands that want to say to the computer, thus any new module can added it by added the word and command separated by “,” to commandswords file and add the word associated with it in to hello.gram file.

III. SYSTEM REQUIREMENTS (HARDWARE AND SOFTWARE)

A. Hardware Requirements

- Processor Intel core i3
- RAM 4GB
- Microphone i-ball

B. Software requirements

- JDK.
- This system operate on all versions of Microsoft Windows such as 2000, XP, 7 ...etc
- Eclipse europa
- Sphinx-4 API
- JSAPI, JSGF

IV. TERMINOLOGY

A. Speech Engine

The Speech Engine loads a list of words to be recognized. This list of words is called a grammar.

B. Speech Recognition

Speech recognition provides computers with the ability to listen to spoken language and to determine what has been said, i.e. it processes audio input containing speech by converting it to text. Supports grammar definitions using JSGF.

C. Java Speech API/JSAPI

The Java Speech API enables speech applications to interact with speech engines in a common, standardized, and implementation-independent manner. Two core speech technologies are supported through the Java Speech API: speech recognition and speech synthesis.

D. Java Speech Grammar Format

Java Speech Grammar Format (JSGF) defines a standard for writing rule-based grammars. All rule grammars in Java Speech applications must follow the JSGF standard. Grammars are used by speech recognizers to determine what the recognizer should listen for, and so describe the utterances a user may say.

E. Sphinx Speech Recognition System

Sphinx-4 is an open source speech recognition system written in the java programming language. The Sphinx-4 architecture has been designed for modularity; any module in the system can be smoothly exchanged for another without requiring any modification of the other modules. Sphinx-4 the ability to dynamically load and configure modules at run time, yielding a flexible and pluggable system.

V. SYSTEM DESIGN

Architecture of the system that shown in figure includes certain components those are important for its formulation. It explains more details about how the system works? Or what the most requirement that must be available to the system in order to operate.

A. User

The voice command input given to the system by the user through microphone. Desktop speech recognition systems get audio input through a microphone. It must be one of the commands displayed on the screen.

B. Recognizer

A speech recognizer is a speech engine that converts speech to text. The basic functionality provided by a Recognizer includes grammar management and the introduction of results when a user says things that match active grammars. The Recognizer interface extends the Engine interface to provide this functionality.

The major steps of a typical speech recognizer are as follows:

- **Grammar design:** Defines the words that may be spoken by a user and the patterns in which they may be spoken.
- **Signal processing:** Analyzes the spectrum (the frequency) characteristics of the incoming audio.

- **Phoneme recognition:** Compares the spectrum patterns to the patterns of the phonemes of the language being recognized.
- **Word recognition:** Compares the sequence of likely phonemes against the words and patterns of words specified by the active grammars.
- **Result generation:** Provides the application with information about the words the recognizer has detected in the incoming audio.

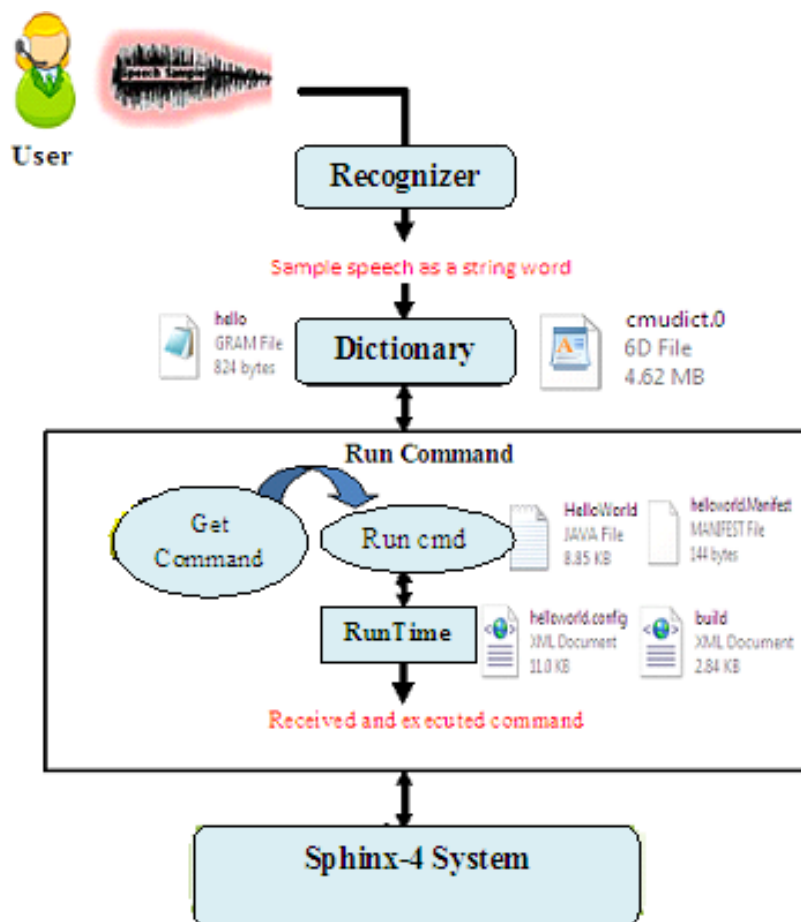


Fig. 2 Architecture of Speech Recognition System.

C. Dictionary

Dictionary file (cmudict.6D or .dict extension) is a list of words with a sequence of phones. Responsible for determining how a word is pronounced. According to grammar (hello.gram), phonetic representation and Dictionary, the system returns an n-best list (i.e.:a word plus a confidence score). Dictionary means the mapping table of phonetic representation and word, for example “thu, thee” is mapping to “the”.

D. Run Command

This component received speech sample after convert it to words as a string from recognizer.

1) **HelloWorld.java:** this is very important component, it is the heart of the structure because in this component can get the final result about the work of the system.

2) **getCommand:** this method received word as command and send it to runCmd.

3) **runCmd:** this method received command as a string from getCommand method and executed it by call the both component runtime and process. Later on command is executed at runtime. The files helloworld.config, helloworld.manifest, build.xml are supporting files at runtime.

E. Sphinx-4 system:

Consist of decoder, front end, knowledge base, grammar file (hello), and this application. Sphinx-4 having ability to dynamically load and configure modules at run time, yielding a flexible and pluggable system.

VI. IMPLEMENTATION AND TESTING USING WINDOWS 7

Implementation is the practical execution of the system.

Process of implementation:

1. In order to make this system work add the sphinx-4 API to the library as the reference to the program.
2. Adding the supporting files such as helloworld.config, helloworld.manifest, build.xml to the package containing java file as shown in Fig.3.

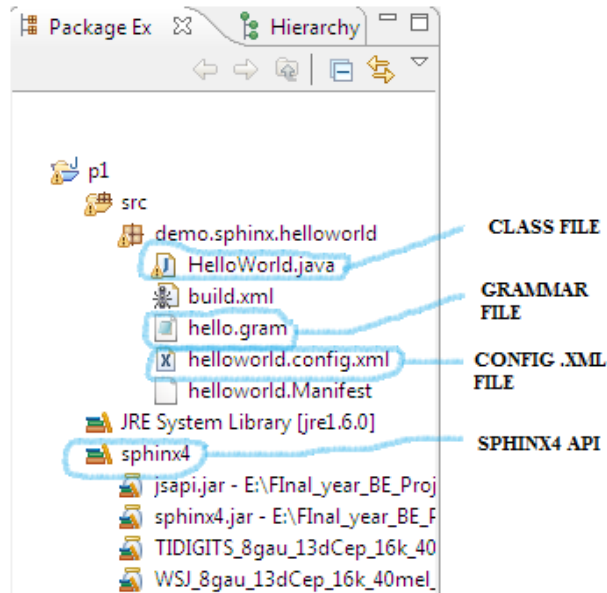


Fig.3 Supporting files in the system.

- The java file contains code that takes input voice and the .xml file matches the input with commands defined in the grammar file and it also maps with dictionary file. Grammar file tells what to say and Dictionary tells how to say. Fig.4 shown below is grammar file.

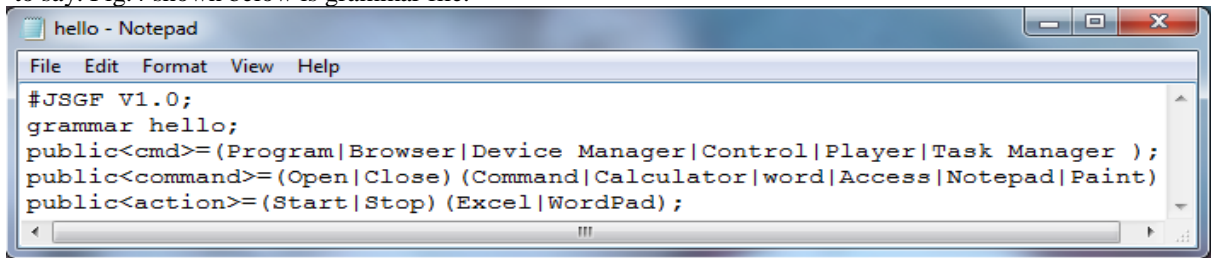


Fig.4 Grammar File required

- Run the java file
 - In Fig.5 below INPUT: Open Paint, notice the figure

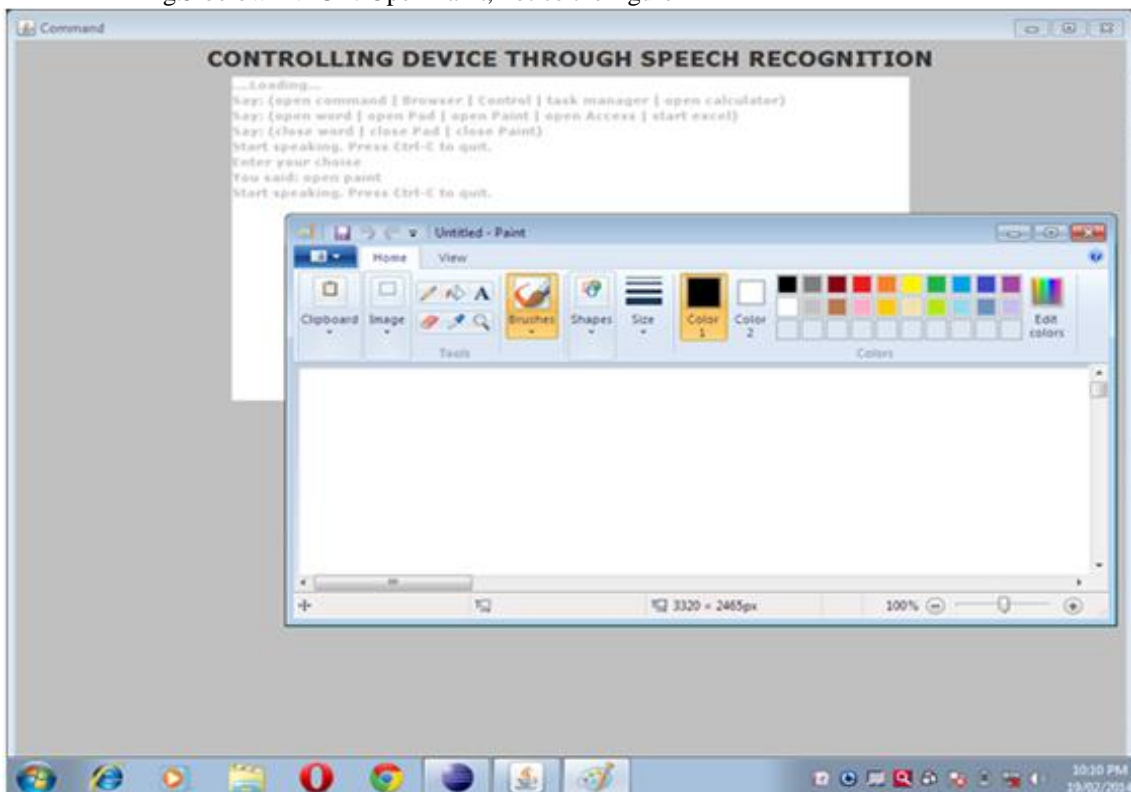


Fig. 5

- Next in Fig.6 INPUT: Close Paint , notice the figure

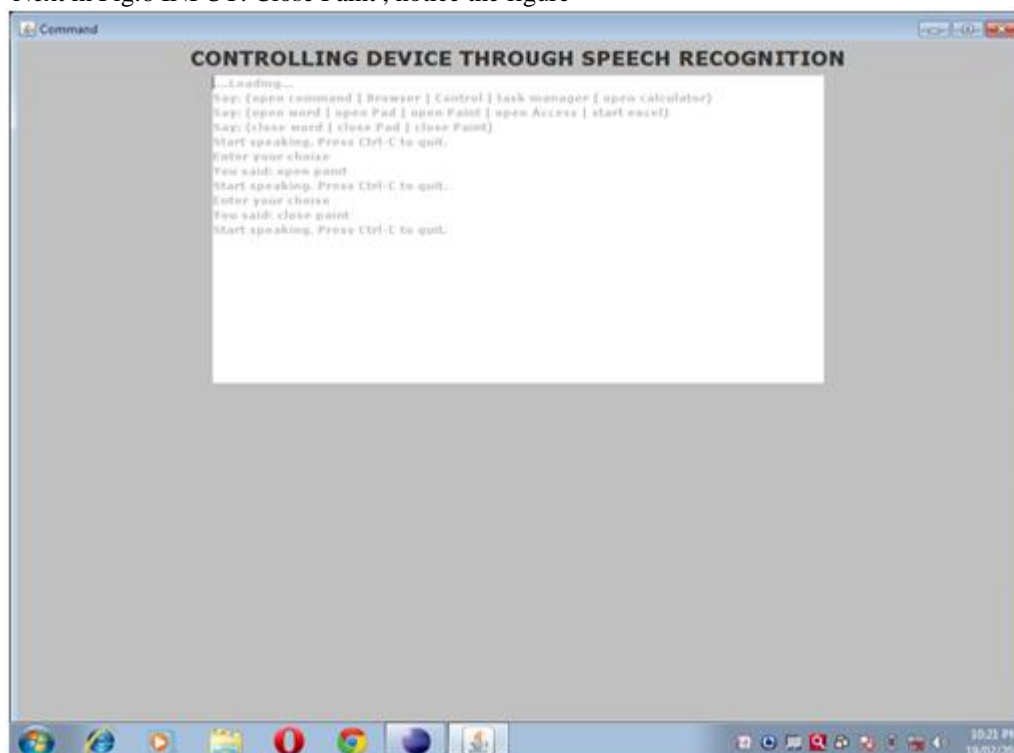


Fig. 6

5. Similarly we can give other commands also.

VII. CONCLUSION AND RECOMMENDATION

This research has successfully achieved the objectives of this study, furthermore, this application provide the great service and useful for computer applications, which aims at representing windows operating system commands based on speech inputs rather than using traditional means of input (e.g. computer keyboard or mouse). The Java Speech Application Programming Interface (JSAPI) will allow applications and applets to use speech technology for the development of advanced user interfaces, it's easy to use and maintain. With the Java Speech API, developers will be able to develop applications with dynamic and compelling user interfaces. It recommended that future study should include all commands of (MS Windows) and solve some error that found in speech inputs because the noise that happened outside the computer or incorrect pronunciation for some commands, this is one of difficulties that face the researcher during achieving this application. This application with some added features can be the future of the computing system.

REFERENCES

- [1] Sonam Kumari, Kavita Arya, Komal Saxena, "Controlling of device through speech recognition"paper , March2012.
- [2] Anupam Choudhary, Ravi KshirsagarS., "Process Speech Recognition System using Artificial Intelligence Technique", paper, Nov. 2012.
- [3] Sandeep Kaur, "Mouse Movement using Speech and Non-Speech Characteristics of Human Voice", international journal, JUNE2012.
- [4] Raghvendra Priyam, Rashmi Kumari, Dr. Prof. Vindeh Kishori, "AI application for Speech Recognition".
- [5] Michael S. Fulkerson and Alan W. Biermann, "Javox: A Toolkit for Building Speech-Enabled Applications", Department of Computer Science, Duke University.
- [6] Sun Microsystems, Inc. Business, "Java™ Speech API Programmer's Guide", Version1.0 October 26, 1998.
- [7] Information about JSAPI. Available: <http://java.sun.com/products/java-media/speech>.
- [8] Willie Walker, Paul Lamere, Philip Kwok, Rita Singh, "Sphinx-4: A Flexible Open Source Framework for Speech Recognition", white paper, SUN MICROSYSTEMS INC.2004.
- [9] StephenS.Wang and JustinStarrent, "A Java Speech Implementation of the Mini Mental Status Exam", ColumbiaUniversity, NewYork.
- [10] Bharat Singhvi, Vishnu Kanth T "ADAPTIVE VOICE RESPONSE SYSTEM", Nov2010.
- [11] Meeras Salman, "EXECUTING COMMANDS OF OPERATING SYSTEM BY USING SPEECH RECOGNITION SYSTEM: AN EXPERIMENT WITH MICROSOFT WINDOWS", journal of Kerbala university, 2012.