ISSN No. 0976-5697

Volume 3, No. 3, May-June 2012

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Speech Controlled Electronic Device System

Derrick Ntalasha
The Copperbelt University
Computer Science Department
Box 21692KitweZambia
derrick.ntalasha@cbu.ac.zm, dbntalasha@yahoo.com

Abstract: This paper presentsa computerapplication that is able to switch on and off, appliances that are connected to the computer using voice. The system performs its intended functions using speech recognition. The user of the system speaks through a microphone which is connected to the computer system. Once the system understands the speech signal,it performs a search process and performs the requested operation. The uttered words by the user consist of sentences or phrases that are recognizable to the system.

***Keywords**:* Speech Recognition, Natural Language and Grammar, Voice Interaction,

I. INTRODUCTION

Speech is an attractive channel for a range of communication tasks with computers. Speaking and listening are universal abilities, whereas typing remains a skill which must be learned and practiced. Voice interaction may be less intimidating to the computer naive population, allowing them to have a greater machine access. Voice is powerful in work environments where hands and eyes are busy or locations where a convenient terminal or keyboard may be impractical or the user is disabled. Voice input and output can allow database access via speech processing software to the extent that a computer interface can employ natural language. Voice is the obvious command channel; indeed, people often talk to computers even when they have no expectation of being understood. Speech synthesis and recognition allow the deployment of conversational systems [1]. Merely adding speech input/output to an application does not make a "user-friendly" interface [2]. Voice interaction involves systems that contain a dialog interface [3] for the user to access the application. With these issues taken in consideration a system called “Speech controlled electronic device System” was developed.

This Speech controlled electronic device System was developed to help the disabled and the able bodied to switch appliances on and off using the computer without difficulties, stop the inconveniences of physically moving to switches and prevent breakdowns of switches on appliances.

II. APPLICATION DEVELOPMENT

The system was designed to use a set of headsets or a microphone. The microphone connects to the computer systems sound card [4] through the microphone cables for input of the speech signal. The system takes the speech signal, converts it to text which is then used to control the appliances plugged into the electronic control box [5]. The whole application is depicted in figure 1 below.

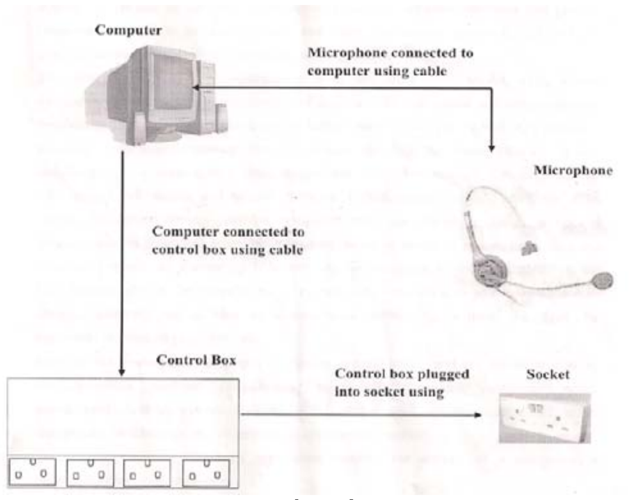
Computer
Microphone connected to computer using cable
Microphone
Computer connected to control box using cable
Control Box
Control box plugged into socket using
Socket

Figure 1: Physical Design

The application uses the existing capabilities of the Java platform [6]which makes it attractive for the development of a wide range of applications. With the addition of the Java Speech API, [7] Javaapplication developers can extend and complement existing user interfaces with speech input and output[8]. The system uses Java as the cross-platform interface to support command and control recognizers [9], through the Java Speech API. The Java Speech API is one of the Java Media APIs [10], a suite of software interfaces that provide cross-platform access to audio, video and other multimedia playback, 2D and 3D graphics, animation, telephony, advanced imaging, and more.

The Java Speech API, in combination with the other Java Media APIs, allows developers to enrich Java applications and applets with rich media and communication capabilities that meet the expectations of today's users. 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 [11].*

In this system though, emphasis is on speech recognition which 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. As an extension to the Java platform, the Java Speech API can be provided as an extension to

Personal Java and Embedded Java devices, allowing the devices to communicate with users without the need for keyboards or other large peripherals.

Most of the processes of a speech recognizer are automatic and are not controlled bythe application developer. For instance, microphone placement, background noise, sound card quality, system training, CPU power and speaker accent all affect recognition performance but are beyond an application's control.

The primary way in which an application controls the activity of a recognizer is through control of its *grammars [12]*.

A. Grammar:

A grammar is an object in the Java Speech API which indicates what words a user is expected to say and in what patterns those words may occur. Grammars are important to speech recognizers because they constrain the recognition process. These constraints make recognition faster and more accurate because the recognizer does not have to check for bizarre sentence. The Java Speech API supports two basic grammar types: rule grammars and dictation grammars. [13] These grammar types differ in the way in which applications set up the grammars, the types of sentences they allow, the way in which results are provided, the amount of computational resources required, and the way in which they are effectively used in application design.

B. Rule Grammars:

In a rule-based speech recognition system, an application provides the recognizer with rules that define what the user is expected to say. These rules constrain the recognitionprocess. Careful design of the rules, combined with careful user interface design, produce rules that allow users reasonable freedom of expression while still limiting the range of things that may be said so that the recognition process is as fast and accurate as possible. Any speech recognizer that supports the Java Speech API must support rule grammars.

The following is an example of a simple rule grammar. It is represented in the Java Speech Grammar Format (JSGF) which is defined in detail in the Java Speech Grammar Format Specification.

```
#JSGF V1.0;
// define the grammar name
grammarSimpleCommands;
// Define the rules
public<Command> = [<Polite>] <Action><Object> (and
<Object>)*;
<Action> = open | close | delete;
<Object> = the window | the file;
<Polite> = please;
```

Rule names are surrounded by angle brackets. Words that may be spoken are written as plain text. This grammar defines one *public* rule, <Command> that may be spoken by users. This rule is a combination of three sub-rules, <Action>, <Object> and <Polite>. The square brackets around the reference to <Polite> mean that it is optional. The parentheses around "and <Object>" group the word and the rule reference together. The asterisk following the group indicates that it may occur zero or more times.

Once a RuleGrammar has been loaded, or has been created with the newRule Grammar method, thefollowing methods of a RuleGrammar are used to create, modify and manage the rules of the grammar.

C. Rule Grammar methods for Rule Management:

Table 1: Rule Grammar methods for Rule Management

Name	Description
SetRule	Assign a rule object to a rulename.
GetRule	Return the rule object for a rulename
GetRuleInternal	Return a reference to the recognizer’s internal rule object for a rulename
ListRuleNames	List known rulenames
IsRulePublic	Test whether a rulename is public
DeleteRule	Delete a rule
SetEnabled	Enable and disable this rule grammar or rules of the grammar
IsEnabled	Test whether a rule grammar or a specified rule is enabled

D. Dictation Grammars:

Dictation grammars impose fewer restrictions on what can be said, making them closer to providing the ideal of free-form speech input. The cost of this greater freedom is that they require more substantial computing resources, require higher quality audio input and tend to make more errors.

A dictation grammar is typically larger and more complex than rule-based grammars.Dictation grammars are typically developed by statistical training on large collections of written text. Fortunately, developers don't need to know any of this because a speech recognition that supports a dictation grammar through the Java Speech API has a built-in dictation grammar. An application that needs to use that dictation grammar simply requests a reference to it and enables it when the user might say something matching the dictation grammar.

Dictation grammars may be optimized for particular kinds of text. Often a dictation recognizer may be available with dictation grammars for general purpose text, for legal text, or for various types of medical reporting. In these different domains, different words are used, and the patterns of words also differ.

A dictation recognizer in the Java Speech API supports a single dictation grammar for a specific domain. The application and/or user select an appropriate dictation grammar when the dictation recognizer is selected and created.

An example of a system that can be incorporated with Java speech API to achieve the objectives of the system is Sphinx-4 [12] which is a state-of-the-art speech recognition system written entirely in the Java programming language.

A dictation grammar is typically adaptive [14]. In an adaptive system, a recognizer improves its *Java Speech Application Programming Interface* performance (accuracy and possibly speed) by adapting to the style of language used by a speaker.

The recognizer may adapt to the specific sounds of a speaker (the way they say words). Equally importantly for dictation, a recognizer can adapt to a user’s normal vocabulary and to the patterns of those words. Such adaptation (technically known as language model adaptation) is a part of the recognizer’s implementation of the dictation grammar and does not affect an application. The adaptation data for a dictation grammar is maintained as part of a speaker profile.

The dictation grammar extends and specifies the Grammar interface by adding the following functionality:

¨ Indication of the current textual context,

¨ Control of word lists.

The following methods provided by the dictation grammar interface allowed an application to manage word lists and text context.

E. Dictation Grammar Interface Methods:

Table 2: Dictation grammar interface methods

Name	Description
SetContext	Provide the recognition engine with the preceding and following textual context
AddWord	Add a word to the dictation grammar
RemoveWord	Remove a word from the dictation grammar
listAddedWords	List the words that have been added to the dictation grammar
listRemovedWords	List the words that have been removed from the dictation grammar

F. Logical Design:

The design of the system is influenced by the nature of the underpinning principle, speech recognition. Speech recognition is a decode and search problem.Searching refers to the process of finding the sequence of words that best fit the features generated from the input speech.

At runtime the system will continuously listen for the incoming speech signal. Once the speech signal is detected, features are internally generated from it and the search process is performed. The recognizer is the class of the speech engine with which the application mostly interacts with. Using an incremental approach, more than one recognizer was instantiated with its grammar list.

The grammar is what guides the concatenation of sound units. Thus in short, the application is able to tab among recognizers and ratchet through grammar lists in order to transcribe the incoming speech as accurately as possible. Bearing in mind the fact that speech recognition is a search problem, here is cliché, to consolidate the basis of reasoning behind this approach: suppose one has to look up an arbitrary word in a dictionary, the word may not be in that dictionary. Therefore, it might be a wise idea to stack up two or three different dictionaries through which to search for a word.

Considering the complexity of the task of continuous speech recognition, the approach taken is bound to slow things down. But it is the sure way to increase the vocabulary size of the system and help meet the goal of ensuring that the system is speaker independent. Furthermore, having a list of grammars optimizes memory usage in that small lists can be loaded into memory as required as opposed to loading a single monolithic grammar.

III. RESULTS AND DISCUSSION

For testing purposes, the grammar which guides sentence formation is implemented in the scenarios below. These scenarios show the sentences that are recognizable. In the scenarios the emphasis is on what the user says as this is the input to the system.

First Scenario:Assume a situation where the Speech controlled electronic device System user wants to switch off the computer.

User: Computer off

Second Scenario:Assume a situation where the Speech Automated Speech controlled electronic device System User wants to switch on a fan.

User: Fan on / Switch fan on / Please switch fan on

Third Scenario:Assume a situation where the Speech controlled electronic device System wants to switch off the fan

User: Fan off / Switch fan off / Please switch fan off

Fourth Scenario:Assume asituation where the Speech controlled electronic device System User wants to switch on a charger.

User: Charger on / Switch charger on / Please switch charger on

Fifth Scenario:Assume asituationwhere the Speech controlled electronic device System User wants to switch off a charger.

User: Charger off / Switch charger off / Please switch charger off

Sixth Scenario:Assume asituationwhere the Speech controlled electronic device System User wants to switch on a fridge

User: Fridge on / Switch fridge on / Please switch fridge on

Seventh Scenario:Assume asituation where the Speech Automated Computer System Version Two User wants to switch off a fridge.

User: Fridge off / Switch fridge off / Please switch fridge off

Eighth Scenario:Assume asituationwhere the Speech controlled electronic device System User wants to switch on the light

User: Light on / Switch light on / Please switch light on

Ninth Scenario:Where the Speech controlled electronic device System User wants to switch off the light.

User: Light off / Switch light off / Please switch light off

IV. SYSTEM PERFORMANCE

The performance of the system is usually specified in terms of accuracy. Accuracy may be measured in terms of performance accuracy, which is usually rated with the number of sentence transcribed.

Since speech recognizers are not perfect listeners they make mistakes. This is one of the challenges of the system because it is working with imperfect speech recognition technology.

Unfortunately, recognition errors cause the user to form an incorrect model of how the system works. For example, if the user says “Socket fan on” and the recognizer hears “Socket light on”, the application will repeat the current message leading the user to believe that he/she did not speak properly.

Therefore, for optimal performance, it is assumed that the speech is coming from speakers that:

a. Have speech characteristics which match the enrolment data
b. Can achieve proper speaker adaptation
c. Work in a clean noise environment such as a quiet office or laboratory

A. Estimating the Accuracy of Speech Recognition:

The system has a maximum of 25 recognizable sentences. To estimate the accuracy, all the twenty five sentences are used. The figure below shows the tabulation of the test results used to estimate the accuracy of speech recognition. These conditions are obtained under the assumed perfect conditions.

USER		CORRECT TRANSCRIPTIONS OUT OF 25 SENTENCES
1		18
2		20
3		16
4		19
	AVERAGE	73%

After system testing, the speech recognition accuracy is estimated to be 73% on the average. The system is able to recognize continuous speech constrained by the grammar. Generally, the vocabulary size is inversely related to the accuracy.

Accuracy test results vary from time to time even with the same speaker. This could be attributed to how clear the speaker is in voice projection or word pronunciations and also speaking pattern.

V. CONCLUSION

The Speech Controlled Electronic Device System uses prompts to interact with its users. Each prompt asks the user to say what he intends to accomplish with the system. This system is able to capture speech spoken into the microphone and convert it into machine readable input.The input is then used to determine what signals to send to the parallel port, which is then able to control power supply of different devices connected to a control circuit in the control box. In the event that the user utters something that the system does not recognize, it does not proceed to the next prompt but tells him/her that it does not understand the utterance and then replays the prompt. It does this until the user says the correct word(s) or sentence. These words or sentences are defined in a grammar file. This file contains grammar rules and rule expansions on the expected answers to each prompt played.

VI. REFERENCES

[1]. Englund C, “Speech recognition in the JAS 39 Gripen aircraft -adaptation to speech at different G-loads”, Centre forSpeech Technology,Stockholm, 2004, pp. 2.

[2]. Breveon, “History of Speech Recognition Technology” [Online document]http://www.breveon.com/howitworks/SpeechRecognitionTechnology ,

[3]. Sphinx-4 Whitepaper, [Online document], Available HTTP:http://cmusphinx.sourceforge.net/sphinx4/#whitepaper ,

[4]. Electronic Publication, Microsoft Encarta [Speech Recognition] 2005

[5]. Muhirwe Jackson, “Automatic Speech Recognition: Human Computer Interface for Kinyarwanda language”, unpublished.

[6]. Reddy D.R.Speech Recognition by Machine: a Review. Proceeding of IEEE,64th ed.vol4 pp: 501-531, 1976

[7]. Tolba, H., and O’Shaughnessy, D., (2001). Speech Recognition by Intelligent Machines,IEEE Canadian Review

[8]. L. Rabiner and B. H. Juang, “Fundamentals of Speech Recognition,”Prentice Hall Inc., 1993.

[9]. Zue, V., Cole, R., Ward, W. “Speech Recognition.Survey of the State of the Artin Human Language Technology”. Kauii, Hawaii, USA, 1996.

[10]. L. Rabiner, “A Tutorial on Hidden Markov Models and SelectedApplications in Speech Recognition,” Proc. of the IEEE, Vol.PROC-77, No. 2, pp. 257-286, February, 1989.

[11]. Willie Walker, Paul Lamere, Philip Kwok, Bhiksha Raj, Rita Singh, EvandroGouvea, Peter Wolf, Joe Woelfel (2004), Sphinx-4: A Flexible Open Source Framework For SpeechRecognition, Sun Microsystems Inc.

[12]. Electronic PublicationSun Microsystems [FreeTTS], 2007

[13]. Jurafsky Daniel and Martin H. James, speech and Language Processing, Computational Linguistics, and Speech Recognition, UK: Prentice-Hall Inc, 2000pp 22 - 105

[14]. Charles Rich"Intelligent User Interfaces", Speech Technology in CE Task Guide, unpublished