EXECUTING COMMANDS OF OPERATING SYSTEM BY USING SPEECH RECOGNITION SYSTEM: AN EXPERIMENT WITH MICROSOFT WINDOWS

تنفيذ اوامر نظام التشغيل باستخدام نظام تميز الصوت : مجال التطبيق مايكروسوفت ويندوز

Assistant Lecture/ Meeras Salman Al-Shemarry
Information technology (IT)
Kerbala university/College of Sciences (Computer Department)

الخلاصـــة

الهدف الرئيسي من هذا البحث هو بناء وتطوير تطبيق بسيط لتنفيذ أوامر نظام التشغيل مايكروسوفت ويندوز MS.WINDOWS باصداراته المختلفة على أساس المدخلات الصوتية بدلا من استخدام الوسائل التقليدية للادخال (مثل لوحة مفاتيح او الماوس لجهاز الكمبيوتر). ويمكن تطبيق هذا النظام بحيث يمكن الاستفادة منه لتطبيقات الحاسب الآلي الاخرى. بمعنى اخر ، يهدف هذا البحث إلى تقديم صوت الإنسان كبديل لأجهزة الإدخال وهذا له اهمية كبيرة في التفاعل البشري بين الحاسوب وعمليات الاتصال، خصوصا للذين يعانون من مشاكل صحية. وقد تم الاستعانة بادوات نظام تميز الصوت Sphinx-4 لغرض تسهيل عمل التطبيق المقترح. كما وقد تم تنفيذ واختبار هذا التطبيق من خلال بيئة نظام MS.DOS كبداية اولى من خلال اختيار عينة بسيطة من اوامر نظام التشغيل MS.WINDOWS و تنفيذها .

ABSTRACT

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. By other words, this system would be useful for computer applications by presenting the human speech as a substantial substitute for input devices; 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. By using voice recognition system sphnis-4 which contains many tools to help and supports in order to execute this system. This research select simple prototype from commands of MS.WINDOWS as start to applicable this system by using Dos environment.

1. INTRODUCTION

Information technology take more roles on many aspects of our daily lives, however, the communication between human beings and Information processing machines become increasingly important. 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 [1]. This research is concerned with speech recognition technology, which is part of speech and signal processing, as well as human computer interaction (HCI). According to Yankelovich *et. al.* [2] applications with voice recognition can also be a very helpful tool for handicapped people who have difficulties with typing. The Sphinx-4 speech recognition system has been jointly developed by Carnegie Mellon University [3][4], Sun Microsystems Laboratories ,and Mitsubishi Electric Research Laboratories (MERL); it has been built entirely in the Java™ programming language, 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 [5]. Speech technology consists of speech synthesis and speech recognition. 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 [6].

Two modes of speech recognition are available [7][8]:

- 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.

2. SYSTEM REQUIREMENTS (HARDWARE AND SOFTWARE)

2.1 HARDWARE REQUIREMENTS

The system required a standard PC or laptop that contains at less simple hardware requirements such as:

• Pentium IV 3.00 GHz • Hard disk drive (80 GB) • 1 GB Random Access Memory (RAM) • Monitor • Sonic Microphone • Speaker

2.2 SOFTWARE REQUIREMENTS

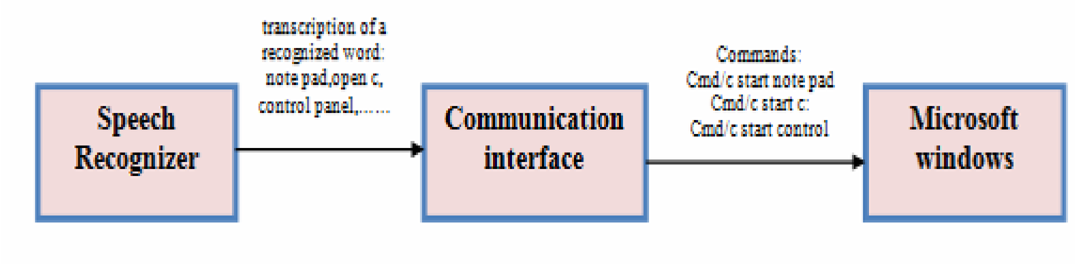
The system required the following software requirements:

1- This system operate on all versions of Microsoft Windows such as Me,2000,XP,7,…etc
2- Java SE 6 Development Kit or better. Go tojava.sun.com, a free software development package from Sun Microsystems that implements the basic set of tools needed to write, test and debug Java applications and applets [9].
3- Ant 1.8.0, available atant.apache.org. Ant is a Java library and command-line tool
4- Sphinx-4, available at sourceforge.net/projects/cmus phinx/files/sphinx4/ (must download the sphinx4-{version}-src.zip).

3. DESIGN SYSTEM PROTOTYPE

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 three commands such as (open c, note pad, control panel) as sample prototype to implement this application. The figure (1) explains the components of the simple prototype system.

hello.gram file *commandswords* file


Speech Recognizer
transcription of a recognized word: note pad,open c, control panel,……
Communication interface
Commands: Cmd/c start note pad Cmd/c start c: Cmd/c start control
Microsoft windows

figure(1): components of the simple prototype system

The perfect solution to this application by using advance programming way called Hash table technique that have two field key(word) and value(command) , 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 .

4. ARCHITECTURE DIAGRAM

Architecture of the system that shown in figure (2), 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? The system consists of two parts that work together and cannot be separated. The general information about each component of sphinx-4 system can known by visited these websites:

http://research.sun.com/sunlabsday/docs.2004/sphinx4.pdf
http://cmusphinx.sourceforge.net/sphinx4

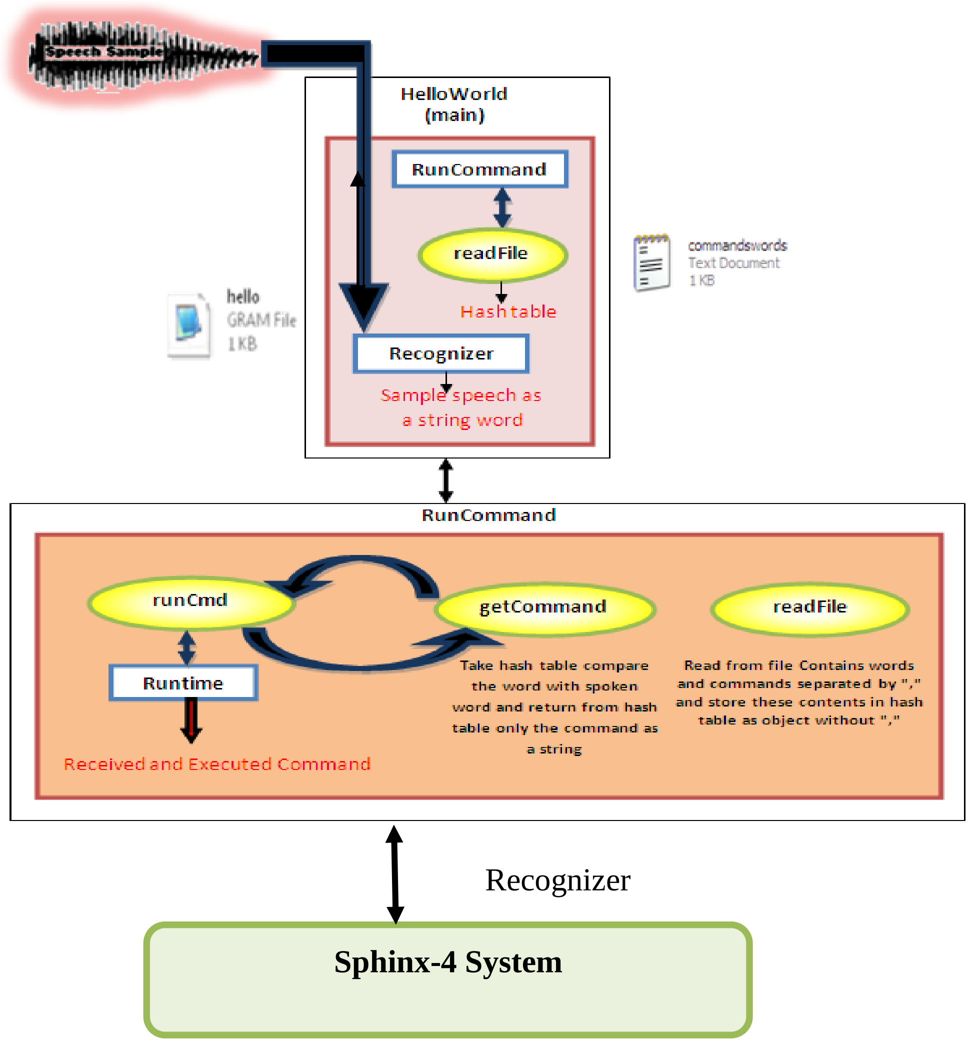

Speech Sample
HelloWorld
(main)
RunCommand
readFile
Hash table
Recognizer
Sample speech as
a string word
hello
GRAM File
1 KB
commandswords
Text Document
1 KB
RunCommand
runCmd
getCommand
readFile
Runtime
Received and Executed Command
Take hash table compare the word with spoken word and return from hash table only the command as a string
Read from file Contains words and commands separated by "," and store these contents in hash table as object without ","
Recognizer
Sphinx-4 System

Figure (2): Architecture diagram for the application system

The application architecture was implemented using sphinx-4 architecture in order to support re-usability and they are easy to maintain and enhance. each individual component of the architecture have a set of inputs and outputs and processes the data sequentially. The Sphinx-4 speech recognition system in these architectures reads a stream of data (speech sample) as input and produces a stream of data (words as a string) as outputs.

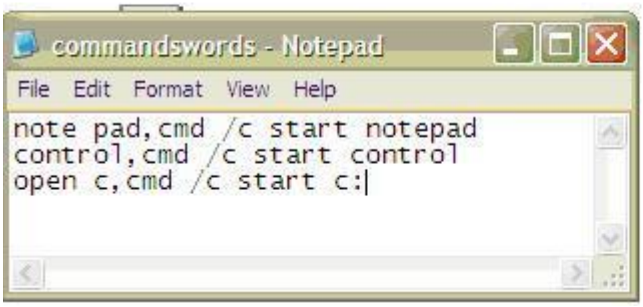
The application architecture consists of:

1. **Database:** the system use text files to save the data by using two text files names *(commandswords.txt)* file and grammar files name *(hello. gram).*
2. **HelloWorld**: 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.
3. **RunCommand:** this component have three methods and received speech sample after convert it to words as a string from *HelloWorld* component and give it to his methods. These methods are:
   a. **readinFile**: this method read information from the data base *commandswords.txt* file, store in *hash table*, and returns it to main method in order to received from it by *getCommand* method.
   b. **getCommand**: this method received hash table from *readinFile* method, and return from it only command as a string to *runCmd* method.

  c. ***runCmd***: this method received command as a string from *getCommand* method and executed it by call the both component *runtime* and *process*.

4. **Sphinx-4 system components**: consist of decoder, front end, knowledge base, grammar file (hello), and this application.

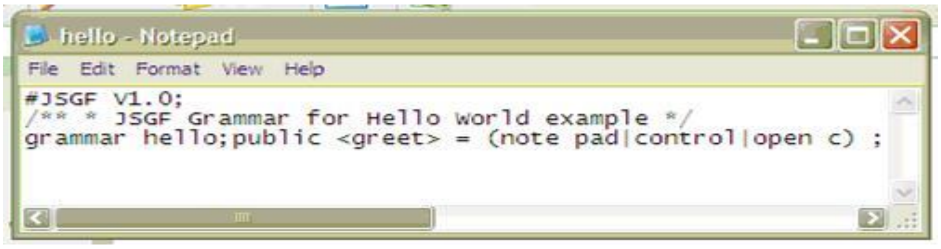
All these components design in java language. The figure (3) describes the *(commandswords.txt)* file contains on commands and words separated by comma (,).



```
note pad,cmd /c start notepad
control,cmd /c start control
open c,cmd /c start c:
```

Figure (3): the commandswords.txt file

The *(hello. gram)* file shown in figure (4) contains on words that the same words on *commandswords.txt* file and these words must found in *CMU.Dictionary*, can check the words if found or not by visit this website: http://www.speech.cs.cmu.edu/cgi-bin/cmudict



```
#JSGF V1.0;
/** * JSGF Grammar for Hello world example */
grammar hello;public <greet> = (note pad|control|open c) ;
```

Figure (4): the *hello. gram* file

A class diagram of this application shown in figure (5):

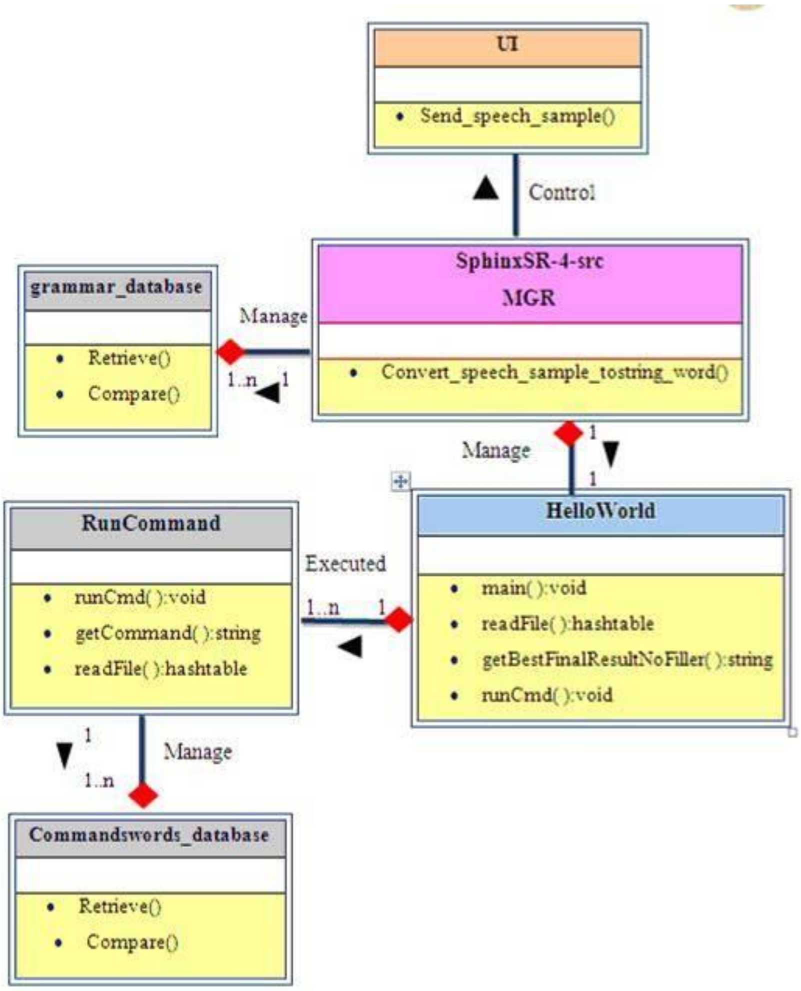


Figure (5): the class diagram of the system

The logic of the application is depicted by the flow chart diagram that shown in figure (6).

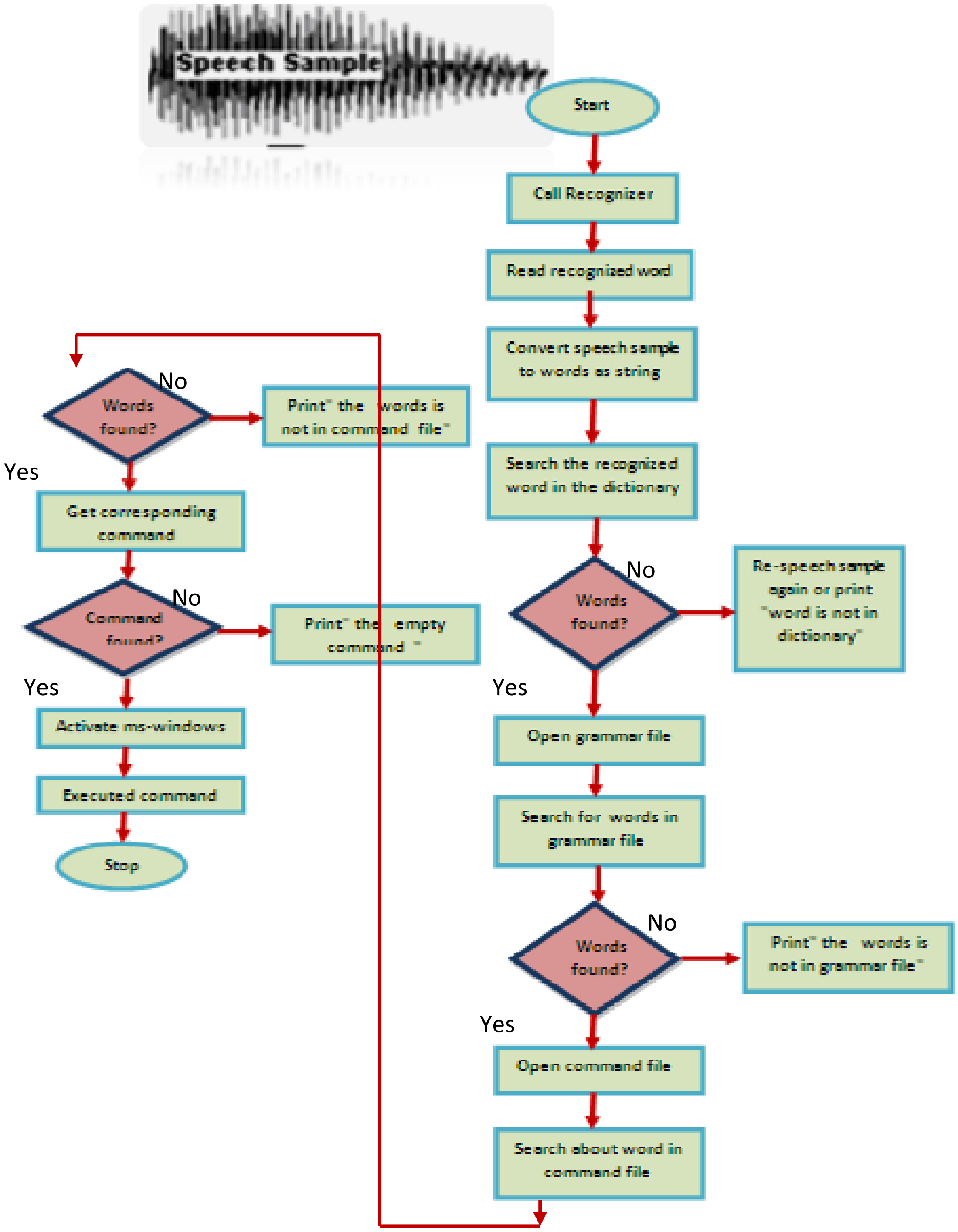

Speech Sample
Start
Call Recognizer
Read recognized word
Convert speech sample to words as string
Search the recognized word in the dictionary
Words found?
No
Re-speech sample again or print "word is not in dictionary"
Yes
Open grammar file
Search for words in grammar file
Words found?
No
Print" the words is not in grammar file"
Yes
Open command file
Search about word in command file
Words found?
No
Print" the words is not in command file"
Yes
Get corresponding command
Command found?
No
Print" the empty command "
Yes
Activate ms-windows
Executed command
Stop

Figure (6): the flowchart diagram for the system

5. IMPLEMENTATION AND TESTING USING DOS INTERFACE

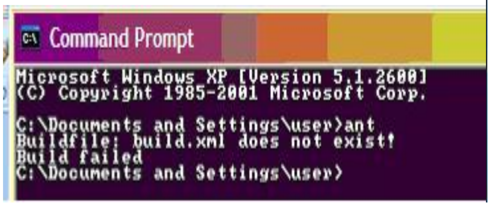
After having the acceptance of the system developed, the implementation phase begins. Implementation is the stage of a project during which theory is turned into practice. During this phase, all the programs of the system are loaded onto the user's computer. After loading the system, training of the users starts. Main topics of such type of training are:

- How to execute the package
- How to enter the data
- How to process the data (processing details)

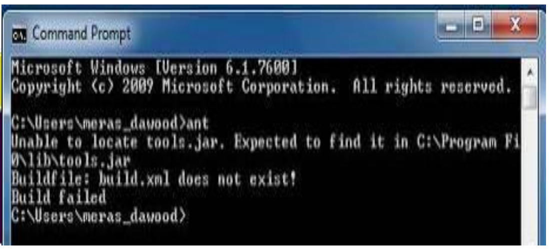
In order to make this system work on the rigth side must preaper the right eniverement to execluted it such as add the sphinx-4-src open sourse and apache-ant-1.8.0RC1-bin software in the local drive(c:) , the ant appliantion make or preaper the dos eneviroment in order to execluted java program,that is means work as a compiler to all contents of the folder project, before you build Sphinx-4, it is important to setup your environment to support the Java Speech API (JSAPI), because a number of tests and demos rely on having JSAPI found inside the sphnix4 library, then go to the Dos in order to check ant application by doing in Dos this command *(C:\>ant)*,notice the figures(7,8).

MS.WINXP

MS.WIN7



```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\user>ant
Buildfile: build.xml does not exist!
Build failed
C:\Documents and Settings\user>
```



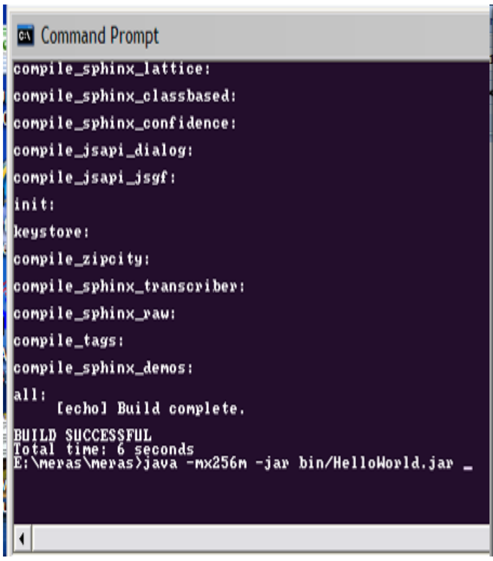
```
Microsoft Windows [Version 6.1.7600]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Users\meras_dawood>ant
Unable to locate tools.jar. Expected to find it in C:\Program Fi
0\lib\tools.jar
Buildfile: build.xml does not exist!
Build failed
C:\Users\meras_dawood>
```

Figure(7)

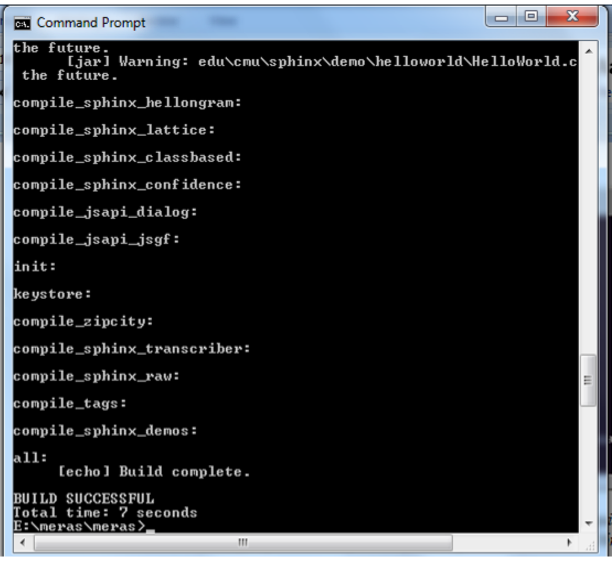
figure(8)

We in the right side because the Ms-Dos not appear error message about ant command just say to you the build.xml file not exist ,because this file not found in this path ,but found inside your project folder , the ant already doing compiler to all your system component by calling the bulid.xml configuration file . The (ant) command doing compiler and convert all java classes to jar files (executable files) and save these files in bin folder, write the name of the most important class to the system that have that main () method, notice the figures(9,10).

:



```
compile_sphinx_lattice:

compile_sphinx_classbased:

compile_sphinx_confidence:

compile_jsapi_dialog:

compile_jsapi_jsgf:

init:

keystore:

compile_zipcity:

compile_sphinx_transcriber:

compile_sphinx_raw:

compile_tags:

compile_sphinx_demos:

all:
     [echo] Build complete.

BUILD SUCCESSFUL
Total time: 6 seconds
E:\meras\meras>java -mx256m -jar bin/HelloWorld.jar _
```



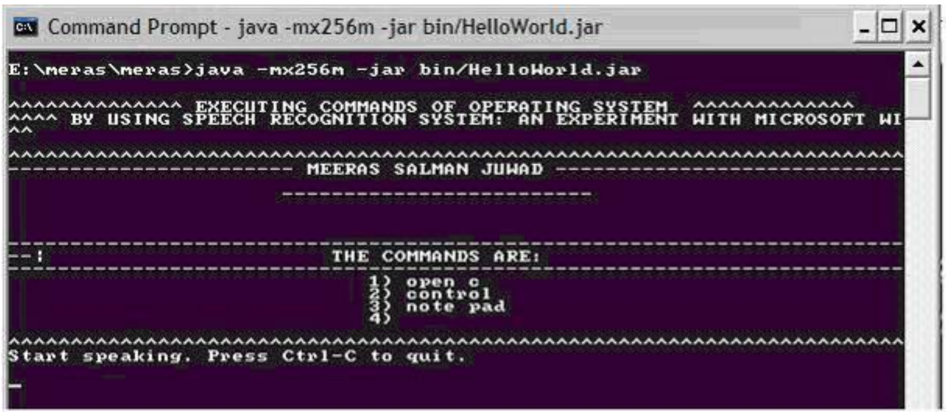
```
the future.
      [jar] Warning: edu\cmu\sphinx\demo\helloworld\HelloWorld.c
 the future.

compile_sphinx_hellongram:

compile_sphinx_lattice:

compile_sphinx_classbased:

compile_sphinx_confidence:

compile_jsapi_dialog:

compile_jsapi_jsgf:

init:

keystore:

compile_zipcity:

compile_sphinx_transcriber:

compile_sphinx_raw:

compile_tags:

compile_sphinx_demos:

all:
     [echo] Build complete.

BUILD SUCCESSFUL
Total time: 7 seconds
E:\meras\meras>_
```

Figure (9)

figure (10)

Press <<enter>> to implement the system, notice the figures (11, 12):

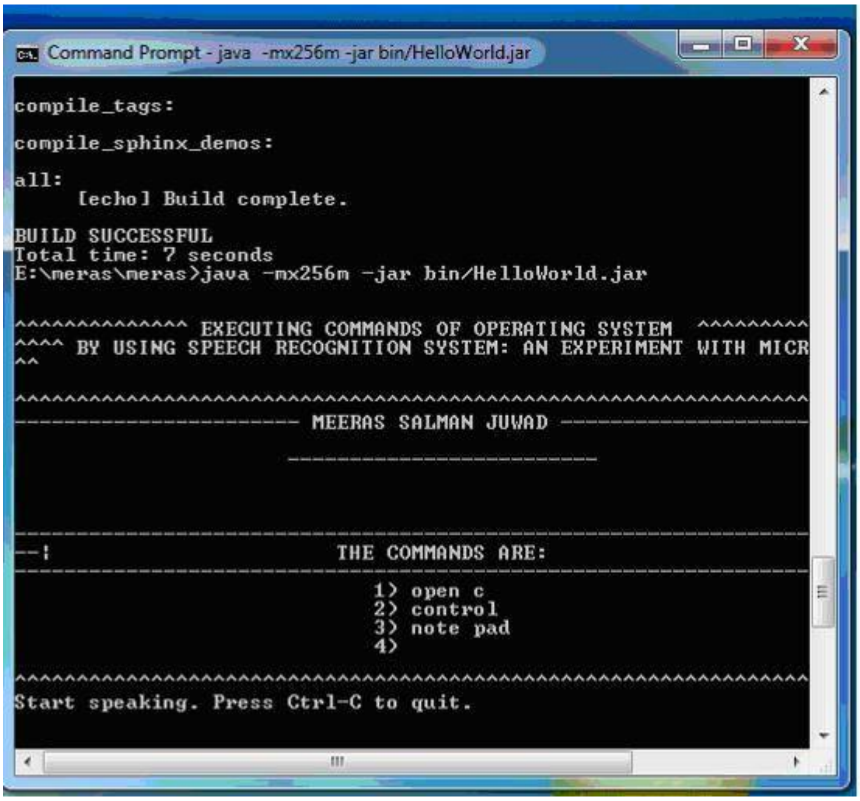
MS.WINXP



```
E:\meras\meras>java -mx256m -jar bin/HelloWorld.jar

^^^^^^^^^^^^^^^ EXECUTING COMMANDS OF OPERATING SYSTEM  ^^^^^^^^^^^^^^
^^^^ BY USING SPEECH RECOGNITION SYSTEM: AN EXPERIMENT WITH MICROSOFT WI
^^

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
----------------------- MEERAS SALMAN JUWAD ----------------------------
                        -------------------------


------------------------------------------------------------------------
--!                                THE COMMANDS ARE:
------------------------------------------------------------------------
                                      1) open c
                                      2) control
                                      3) note pad
                                      4)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Start speaking. Press Ctrl-C to quit.
_
```

Figure (11)

MS.WIN7

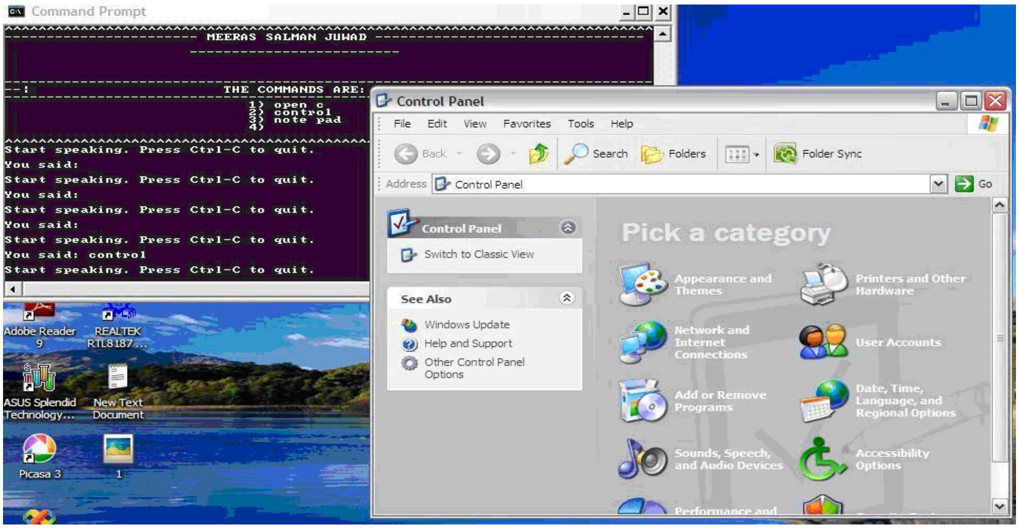


```
compile_tags:

compile_sphinx_demos:

all:
     [echo] Build complete.

BUILD SUCCESSFUL
Total time: 7 seconds
E:\meras\meras>java -mx256m -jar bin/HelloWorld.jar


^^^^^^^^^^^^^^ EXECUTING COMMANDS OF OPERATING SYSTEM  ^^^^^^^^^
^^^^ BY USING SPEECH RECOGNITION SYSTEM: AN EXPERIMENT WITH MICR
^^

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
----------------------- MEERAS SALMAN JUWAD --------------------
                      -------------------------



----------------------------------------------------------------
--!                          THE COMMANDS ARE:
----------------------------------------------------------------
                               1) open c
                               2) control
                               3) note pad
                               4)

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Start speaking. Press Ctrl-C to quit.
```

Figure (12)

If said: control panel ,notice the figures (13,14)

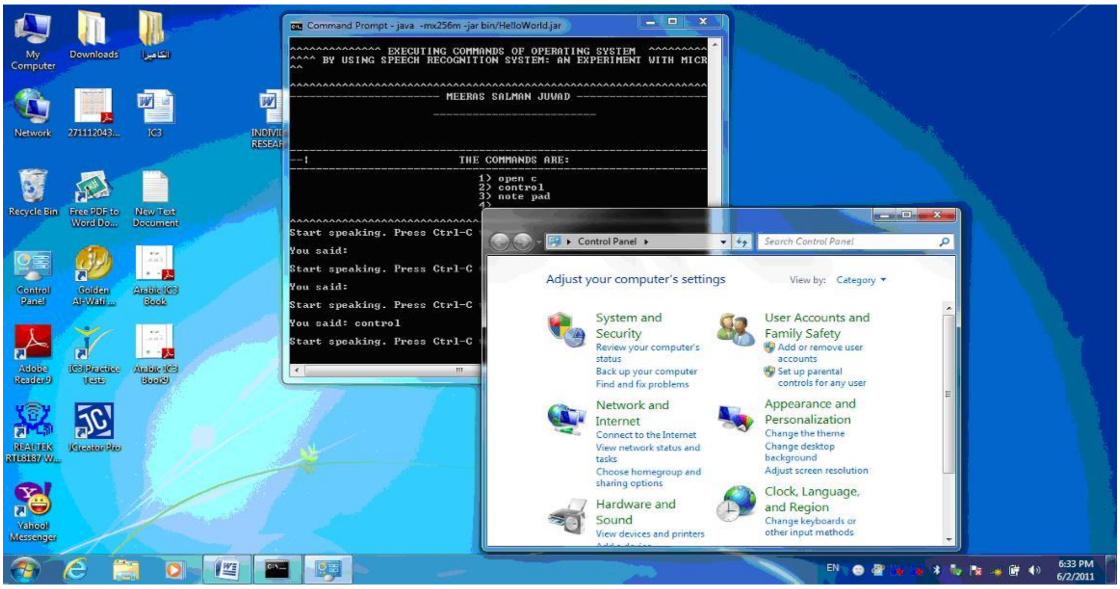
MS.WINXP



```
Command Prompt
---------------------- MEERAS SALMAN JUWAD ----------------------
                       ------------------------

--!                         THE COMMANDS ARE:
                                  1) open c
                                  2) control
                                  3) note pad
                                  4)
Start speaking. Press Ctrl-C to quit.
You said:
Start speaking. Press Ctrl-C to quit.
You said:
Start speaking. Press Ctrl-C to quit.
You said:
Start speaking. Press Ctrl-C to quit.
You said: control
Start speaking. Press Ctrl-C to quit.
```

Figure (13)

MS.WIN7

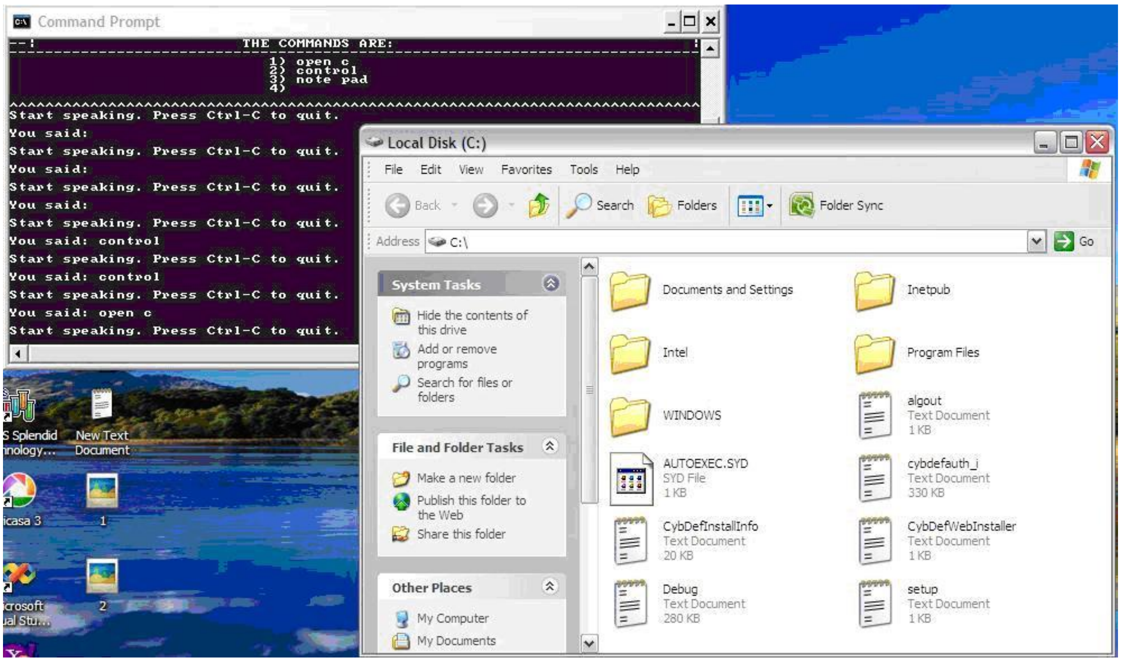


```
Command Prompt - java  -mx256m -jar bin/HelloWorld.jar
^^^^^^^^^^^^^^ EXECUTING COMMANDS OF OPERATING SYSTEM  ^^^^^^^^^
^^^^ BY USING SPEECH RECOGNITION SYSTEM: AN EXPERIMENT WITH MICR
^^
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
--------------------- MEERAS SALMAN JUWAD ----------------------
                    -------------------------

----------------------------------------------------------------
--!                      THE COMMANDS ARE:
----------------------------------------------------------------
                              1) open c
                              2) control
                              3) note pad
                              4)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Start speaking. Press Ctrl-C
You said:
Start speaking. Press Ctrl-C
You said:
Start speaking. Press Ctrl-C
You said: control
Start speaking. Press Ctrl-C
```

Figure (14)

If said: open c , notice figures(15,16)

MS.WINXP

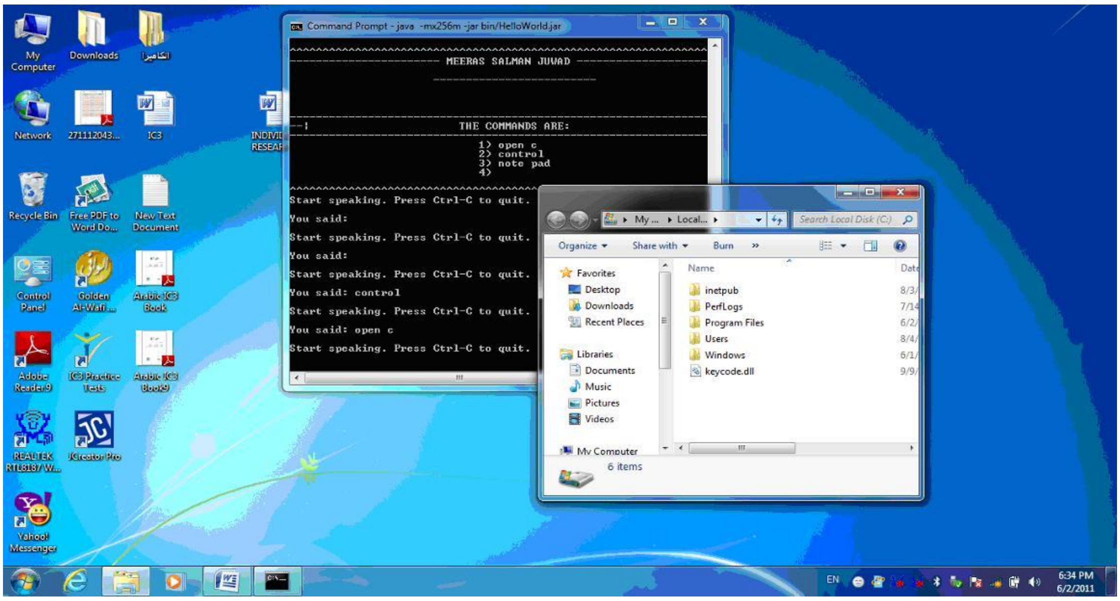


```
--!                          THE COMMANDS ARE:                          !
                                  1) open c
                                  2) control
                                  3) note pad
                                  4)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Start speaking. Press Ctrl-C to quit.
You said:
Start speaking. Press Ctrl-C to quit.
You said:
Start speaking. Press Ctrl-C to quit.
You said:
Start speaking. Press Ctrl-C to quit.
You said: control
Start speaking. Press Ctrl-C to quit.
You said: control
Start speaking. Press Ctrl-C to quit.
You said: open c
Start speaking. Press Ctrl-C to quit.
```

Command Prompt
Local Disk (C:)
File Edit View Favorites Tools Help
Back Search Folders Folder Sync
Address C:\ Go
System Tasks
Hide the contents of this drive
Add or remove programs
Search for files or folders
File and Folder Tasks
Make a new folder
Publish this folder to the Web
Share this folder
Other Places
My Computer
My Documents
Documents and Settings
Inetpub
Intel
Program Files
WINDOWS
algout Text Document 1 KB
AUTOEXEC.SYD SYD File 1 KB
cybdefauth_i Text Document 330 KB
CybDefInstallInfo Text Document 20 KB
CybDefWebInstaller Text Document 1 KB
Debug Text Document 280 KB
setup Text Document 1 KB

Figure (15)

MS.WIN7

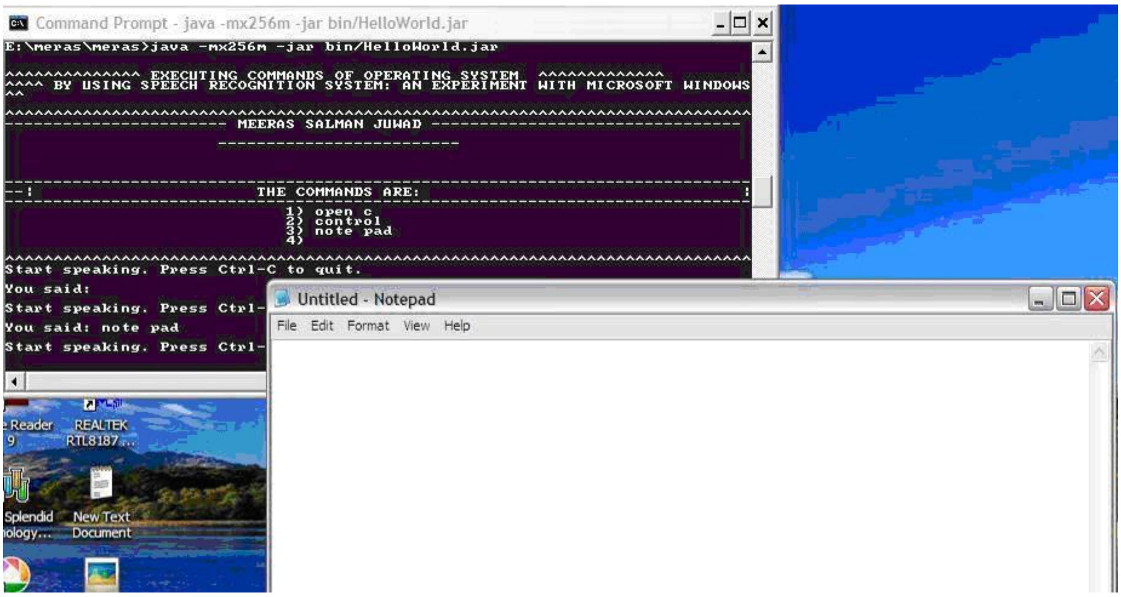


```
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
------------------------- MEERAS SALMAN JUWAD -------------------------
                        --------------------------

--!                          THE COMMANDS ARE:
                                  1) open c
                                  2) control
                                  3) note pad
                                  4)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Start speaking. Press Ctrl-C to quit.
You said:
Start speaking. Press Ctrl-C to quit.
You said:
Start speaking. Press Ctrl-C to quit.
You said: control
Start speaking. Press Ctrl-C to quit.
You said: open c
Start speaking. Press Ctrl-C to quit.
```

Figure (16)

If said: note pad , notice the figures (17,18)

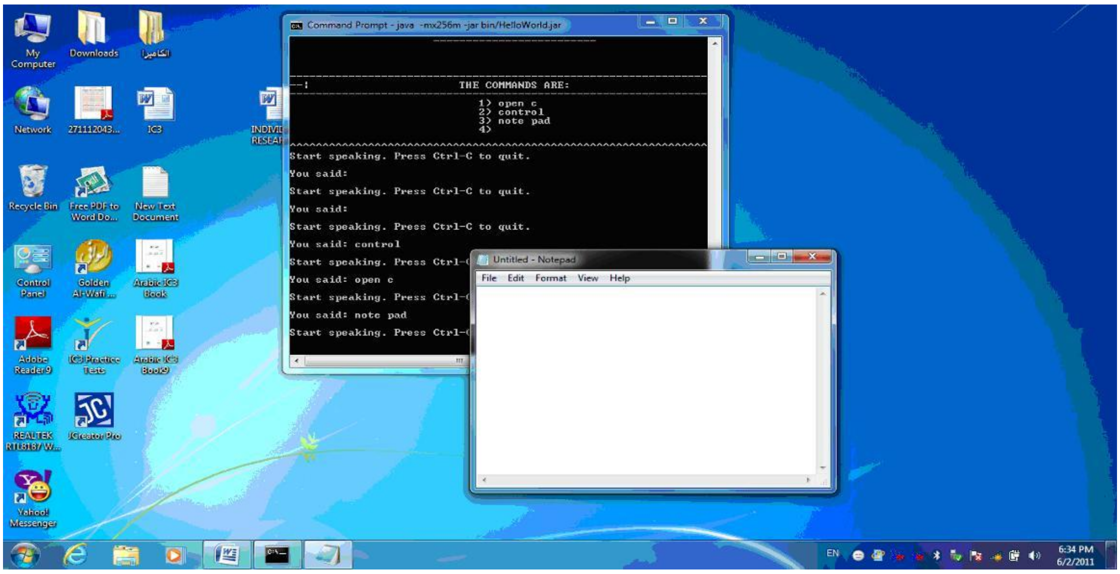
MS.WINXP



```
E:\meras\meras>java -mx256m -jar bin/HelloWorld.jar

^^^^^^^^^^^^^^^ EXECUTING COMMANDS OF OPERATING SYSTEM  ^^^^^^^^^^^^^
^^^^ BY USING SPEECH RECOGNITION SYSTEM: AN EXPERIMENT WITH MICROSOFT WINDOWS
^^

^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
---------------------- MEERAS SALMAN JUWAD -----------------------------------
                       -------------------------



------------------------------------------------------------------------------
--!                        THE COMMANDS ARE:                                 !
------------------------------------------------------------------------------
                               1) open c
                               2) control
                               3) note pad
                               4)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Start speaking. Press Ctrl-C to quit.
You said:
Start speaking. Press Ctrl-
You said: note pad
Start speaking. Press Ctrl-
```

Figure (17)

MS.WIN7



```
                    ---------------------------


------------------------------------------------------------------------------
--!                        THE COMMANDS ARE:
------------------------------------------------------------------------------
                               1) open c
                               2) control
                               3) note pad
                               4)
^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
Start speaking. Press Ctrl-C to quit.
You said:
Start speaking. Press Ctrl-C to quit.
You said:
Start speaking. Press Ctrl-C to quit.
You said: control
Start speaking. Press Ctrl-(
You said: open c
Start speaking. Press Ctrl-(
You said: note pad
Start speaking. Press Ctrl-(
```

Figure (18)

………….

……………………………………………………………….

6. CONCLUSION AND RECOMMENDATION

This research was successfully achieve the objectives of this study, furthermore, this application provide the greatest 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 aim of this research to construct and develop Commands Voice Recognition system of Microsoft Windows that is capable of recognizing and responding to speech inputs. It recommended that future study should include all commands of (MS.Win) 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 achieve this application but currently can overcome by using robotic reader application . It is also suggested that a complete requirement by the future researcher for this system that should be provided in order to measure the model functionalities (SW) and non functionalites (HW) such as performances for example, If it used less than these requirements in this research, the executing speed of this application become more difficult because the hardware environment unsuitable to sphinx system tools that used to implement this application. The future work may be able to carry out a full implementation and include other applications to capture the requirements in order to make it reliable and standardize.

7. RESEARCH SIGNIFICANCE

The Significance Of research includes how to use technology Speech Recognition which makes life easier and has much to offer for the future. The main advantages of this system include save time and avoid error's typing. Thus, this system makes communicating with the computer faster than manual interfaces do, like for example the keyboard. Instead of typing the command it is enough to say it. Finally, handicapped people can use this system to interact with computer's operating system.

REFERENCES

[1] P. Scott ," The Definitive Guide to Building Java Robots", 2006, New York, NY, USA: ISBN: 1-59059-556-4

[2] N. Yankelovich, G. Levow, and M. Marx, "Issues in speech user interfaces",1995.

[3] P. Lamere, P. Kwok, E. Gouvea, B. Raj, and P. Wolf , " The cmu sphinx-4 speech recognition system", 2004.

[4] Carnegie Mellon University, "Sphinx-4", http://cmusphinx. sourceforge.net/sphinx4/ , 2004.

[5] P. Lamere, P. Kwok, E. Gouvea, B. Raj, and P. Wolf, "Sphinx-4: A Flexible Open Source Framework for Speech Recognition", 2004c.

[6] S. Preston, "The Definitive Guide to Building Java Robots", 2005.

[7] V. Juraj, K. Juraj, R. Gregor," The Training of Slovak Speech recognition System based on Sphinx 4 for GSM Networks". In: *49th International Symposium* , Zadar, Croatia, 12-14 September 2007.

[8] Thesis or dissertation Computer Science, M. Jackson, "Automatic Speech Recognition: Human Computer Interface for Kinyarwanda Language", Master of Science in Computer Science of Makerere University, Makerere, 2005.

[9] R.G., P.G. Kuzmenko, I.V. Maltsev," Sphinx SD Tools User Guide: Sphinx SD Tools Developers Team",06 March 2008.