Personal Assistant for User Task Automation

Rasika Anerao[1], Utkarsh Mehta[2], Akash Suryawanshi[3]

[1]*(Computer Engineering, PES Modern College Of Engineering, Pune/ Savitribai Phule University,India)*
[2]*(Computer Engineering,PES Modern College Of Engineering, Pune/ Savitribai Phule University, India)*[3]*(Computer Engineering,PES Modern College Of Engineering, Pune/ Savitribai Phule University,India)*

ABSTRACT : *This paper presents the approach to develop a personal assistant that reduces the use of input devices like mouse and keyboard on our personal computer. Giving commands via speech makes it user friendly. This paper also describes the representation model, available personal assistants in the market, along with the implementation of this system. Details about the additional feature of remote access and addition of new commands that makes it different from others are also mentioned.*

Keywords – *addition of new commands, personal assistant, remote access, speech commands.*

I. INTRODUCTION

In today's world, all that we want is automation. May that be in our work or for our daily needs. So why not automation on our Personal Computer (PC) or our laptop?Thus, we came up with this concept of 'Personal Assistant'. The personal assistant is nothing but an implementation of assistance virtually on the user`s PC. The application is used with voice commands mainly. It is also used with textual input using keyboard and also using our mobile phones for remote access. This application runs on user commands. There are some predefined commands in the system and new commands may be added as and when necessary. System notifies the user about new emails, news reports, weather reports, etc. User can set alarms and reminders as well. System tasks like shutdown, lock system, sleep, file reading and writing can also be executed by the system. There are several modules for each task like time, alarm, search etc. The interface calls these predefined modules when specifiedin commands.

System initiates at startup and speech input to the system can be suspended with a command. All the tasks that require the use of keyboard and/or mouse can be handled by this system.

Now the features that make this application better are remote access and addition of new commands to the system at user's choice. Using assistant from anywhere is the newest feature as compared to other assistants. As we can use the assistant by remote access, it removes the limitation of location. We can give commands from any location and get the things done easily. Also user will be able to add new commands which will make the assistant better and user friendly.

II. MATHEMATICAL REPRESENTATION

Input Set:

The personal assistant takes voice and text input. Text input is also through remote signal. So the set of inputs will be,

I1 = {predefined command, fixed pattern sentential command, random sentences as command}
I2 = {text, voice, remote text}
Thus, I = I1 U I2

Input I = {all sentences in English via speech, all sentences in English as text, remote text input}

Output Set:

The output for the specified inputs above will be response determined by the system according to the input given and the database containing all the necessary inputs and their respective outputs.

O1 = {voice, display, text}
O2 = {GUI, application response}
Thus, O = O1 U O2

Output O = {Response for corresponding voice input, Response for corresponding input via GUI, application response}

There is a one to one relation between input and output as there is only one output for a particular input.

Functions:

Following are the operations performed on the input in the system:

1. Recognize (): This operation basically gets the input from the user. For speech input, it converts the voice into text and saves it. For text, the input is saved directly into the database while in case of remote signal, the signal is directly converted to the command.

2. Extract (): This operation analyses the input string saved and system gets an ideaabout which command is expected to be executed for the given input.

3. Search (): This operation searches the local database for the response of the command extracted by the previous operation.

4. Response (): This operation gives the output that we see on the screen or through speech for the given command.

III. APPROACH TO THE SYSTEM

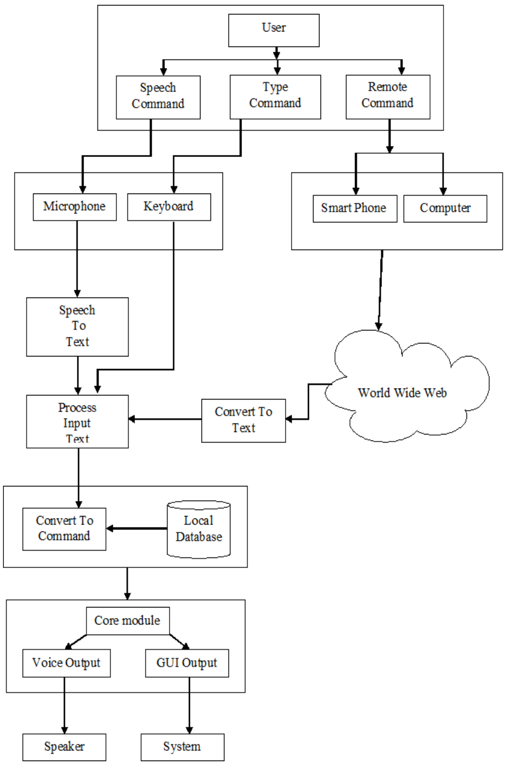

Fig 1: System architecture

The above architecture depicts the system. The basic flow of the system is as above in the figure. User gives text or voice input. Voice commands are converted to text through Google speech API and converted to text. Text input is simply stored in the database for further process. It is recognised and matched with the commands available in the database. Once the command is found, its respective task is executed as text, voice or through user interface as output. It can also be in the form of application execution.

IV. LITERATURE SURVEY

Various assistants available in the market include DRAGON – Naturally Speaking, Syn Virtual Assistant, Braina, etc. Also assistants popularly used in the smartphones are Cortana, Siri, and Google now etc.

Dragon – Naturally speaking assistant [1] that is available in the market has the functionalities of dictation, text to speech and command input. Although it needs identical configurations on the computers in the networked environment for assistance. Syn Virtual Assistant [2] available is mainly for voice commands that are used to connect to online services. Facility to add new commands is available but very few commands are predefined here. Braina [3] is an assistant that is developed especially for computers and its tasks. Here an android app is used to interact with the system over a Wi-Fi network. Other popular assistants available are, Google Now [4],Siri [5] and Cortana [6]. Now our goal is to add all these functionalities into a single system making it simpler and user- friendly. Also for the remote access, using a website, commands can be added to the system along with the execution of tasks. Let's see few in brief.

Dragon Naturally Speaking [1]:

Dragon NaturallySpeaking [1] (also known as Dragon for PC), is a speech recognition software package developed by Dragon Systems. NaturallySpeaking uses a minimal user interface. The software has three primary areas of functionality: dictation, text-to-speech and command input. The user is able to dictate and have speech transcribed as written text, have a document synthesized as an audio stream, or issue commands that are recognized as such by the program. In addition, voice profiles can be accessed through different computers in a networked environment, although the audio hardware and configuration must be identical on both machines. Dragon Dictate was first released for DOS, and utilized hidden Markov models, a probabilistic method for temporal pattern recognition.

Syn Virtual Assistant [2]:

Syn Virtual Assistant is a free and flexible Virtual Assistant Platform designed for users and developers to add Artificial Intelligence to their computer. It talks and takes voice commands and connects you to many online services. It can be even used to control your computer. Syn Virtual Assistant is a free AI software that exist in the market. Developers can easily extend its features using any .Net programming languages like C#, VB, C++, Net or even Iron Python.Users make their own customized AI Assistants for specific tasks including Smart House, Home Automation or even Task-Specific Virtual Assistants.

For investment opportunities -

- It remembers
- It understands
- It cares
- It speaks your language
- It's always at your service
- It learns

V. IMPLEMENTATION

Following is the list of features implemented:

1. Weather forecast
2. News alerts
3. Sets alarm
4. Displays date and time
5. Opens application
6. Locks the workstation.
7. Hibernates the system.
8. Power-off the system.
9. Sleep your machine.
10. Play typical song or video
11. Access specific file
12. Know definition of words
13. Search information on Google or Wikipedia
14. Dictate document and save
15. Email alert
16. Mail from any account to any other account
17. Use the assistant using site through text commands
18. Conversation with the assistant
19. Adding new commands (We can add new voice command by providing the action that is supposed to be executed when the command is given).

All of the features to be executed are given voice input as command. Voice command is recognized by Google Speech API.

Let's see the implementation of above tasks

1. Weather forecast:

For weather reports, we have used an API called W-underground [7]. By registering as a developer on the wunderground web site we get an API-key, which is used in the request that the executable sends to the API. The executable gets the weather information in xml format.

http://api.wunderground.com/api/"+wunderground_key + "/conditions/q/autoip.xml
This xml file detects your location and gives the weather information of your location only.
If we want to get the information of specific city we can use following link,
http://api.wunderground.com/api/"+wunderground_key + "/conditions/q/" + city_name + ".xml"

The xml from API contains information like-
Place
Observation time
Type of date like weather of forecast etc.
temperature_
Relative humidity
Wind
Pressure
Dewpoint
Visibility km
Latitude
Longitude and many more.

The xml is parsed and information is extracted and relevant information is given to user.

2. News headlines:

For this we use the same principle as that of weather forecast. We used Google RSS [8] to get the headlines and the xml file is parsed to get the headlines. The links are as follows,
http://news.google.com/?output=rss

3. Locking, hibernation, shutting down and rebooting the system:

For these different system calls are used directly from the application these calls are given and the calls are as follows,
1.Lockdown:"Rundll32.exe User32.dll, LockWorkStation"
2. Shutdown: shutdown -s -t 10
3. Reboot:
4.Hibernate:"rundll32.exePowrProf.dll, SetSuspendState"

4. Play typical song or video:

The .NET has a function called Directory.GetFiles (folder_path, filename_pattern); Using this function we search all the drives and the folders recursively. For songs we search the input song string from user with as,
Directory.GetFiles (folder, "*" + name + "*.exe"); and for videos, we use different extensions like mp4, avi, wmv, flv, mpg, mkv etc.

Once the file matching with the pattern is found, that file is executed and the respective song or video starts. This process continues until all the drives and folders are not searched.

5. Know definition of words:

A big database of words and their definitions is imported to client machine MySQL database.It contains various tables out of which we mostly used only two tables'- wn_gloss which contains all the definitions and wn_synset which contain all the words. An ID is given to each word called synset_id.
We use the following MySQL query to get all the available definitions of given-word.

SELECT m.gloss FROM `wn_gloss` AS m, wn_synset AS n WHERE n.word =\""+given-word+"\" AND n.`synset_id` = m.`synset_id`

Following is the example of a query with its output-
SELECT m.gloss FROM `wn_gloss` AS m, wn_synset AS n WHERE n.word ="computer" AND n.`synset_id` = m.`synset_id`

6. Dictate document and save:

The principle behind this is that, when user says “dictation on”, the assistant stores the input sentences on the MySQL without executing them as commands and when user input is “dictation stop”, an executable file is called which fetches all the inputs from database using the following query,

```
SELECT `command` FROM `speech_string` WHERE `srno`< (Select `srno` from `speech_string` where `command`="dictation stop" order by `srno` desc Limit 0,1) AND `srno`> (Select `srno` from `speech_string` where `command`="dictation on" order by `srno` desc Limit 0,1)"
```

Once the strings are collected, it is ready to create a word document and add all the strings to it. We have used an API called DocX which provides function to create word file and add text to it.

```
var doc = DocX.Create(fileName);
var paraFormat = new Formatting();
paraFormat.FontFamily = new
System.Drawing.FontFamily("Times New
Roman");
paraFormat.Size = 14D;
while (msqlReader.Read())
           {
              paraOne = paraOne + " " +
msqlReader.GetString(0);
           }
doc.InsertParagraph(paraOne, false, paraFormat);
doc.Save();
```

7. Conversation with the assistant:

A chatter bot is used to interact with the assistant. It uses AIML file to process the input set and give a set of outputs i.e.Responses.

Following code is used to initialise the bot,

```
Dim myBot As New Bot()
Dim myUser As New User("consoleUser", myBot)
myBot.loadSettings()
myBot.isAcceptingUserInput = False
myBot.loadAIMLFromFiles()
myBot.isAcceptingUserInput = True
```

After the initialisation, we have defined following function and passed all the strings to that function,

```
Public Function add_cmd_chat(ByVal cmd1) As Integer
Dim r As New Request(cmd1, myUser, myBot)
Dim res As New Result(myUser, myBot, r)
res = myBot.Chat(r)
If (cmd1.ToLower() <> "quit") Then
Label2.Text = res.ToString()
End If
SAPI.Speak(res.ToString())
Return 1
End Function
```

8. Remote input:

This is the main feature of this system. We can give input to the assistant by using remote module. Input to the assistant is through internet and LAN or Wi-Fi. For LAN or Wi-Fi, we have created a local Apache web server which contains a webpage which makes the user login into the portal.As the application on user’s computer starts, another more module is executed in background which sends requests to MySQL databases and check if some new row is inserted. If a new row is detected that command is executed accordingly.

Following query is executed to get the latest command,

```
SELECT * FROM `command1` WHERE `id`=" + remote_id + " order by `srno` desc LIMIT 0,1
```

The remote_id is given on every setup on client PC.

It is stored on the user as well as website MySQL databases.

9. Adding new commands:

We have used an Ms-Access database to store custom commands as well as some predefined commands.It contains regular expressions with which the input from user are matched and the executable file with arguments given in databaseonly are executed. To add command, user fills the fields like user input, speech response etc. then click on one of the following buttons open a file, open a folder, open a web page and custom speech response. On submitting information required, a simple insert query is performed on Ms-Access database which store the relevant information so that the user expected response is obtained for the given speech input.

VI. RESULT

When the application is executed it starts remote module in background. When user clicks on the centre of the gadget on the screen it starts the speech recognition and command runner module. When user gives speech input it is recognised and stored on the MySQL database which is fetched by the command runner module.When it gets the command it first tries to match the input with the commands stored in the Ms-Access database ifmatched it executes the command accordingly.If not then it check the command for some hardcoded flows of matched then the relevant output is given to user.If not the input is passed to chatter bot which gives output for sure. Thus, every input is handled and the system assures good performance. It’s the Google Speech API that assures accuracy for voice commands.

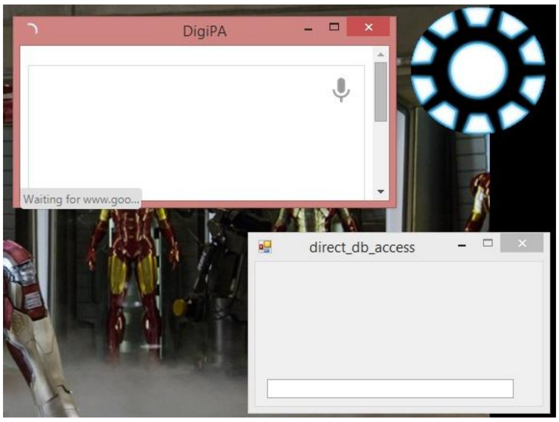

Fig. 2. Interface of the application

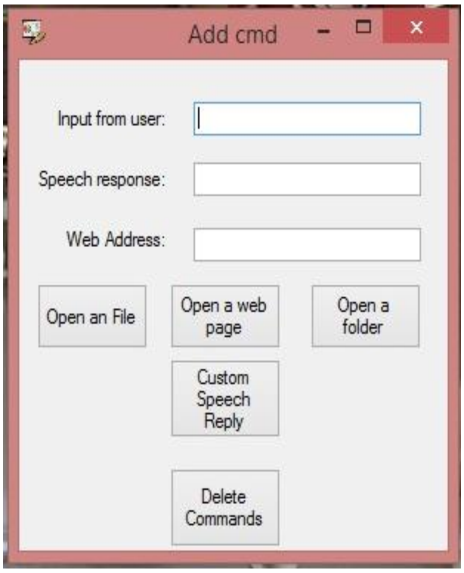

Fig. 3. Interface to add command feature

VII. CONCLUSION

This paper introduces with the approach that can used to develop a personal assistant. Though a number of assistants are available in the market for smartphones, personal assistants for PC's are not that popular in our country. Such system is mainly going to facilitate physically challenged people using computer. Moreover an additional feature of remote access eradicates the limitation of area within which the system can be accessed for normal users. This will help to develop an expert system of home automation which is an extended scope of this system. This can also be used professionally for certain tasks automation in a company and prove very helpful to users.

VIII. Acknowledgements

The authors of this paper would like to thank Ms. Manasi K. Kulkarni and Ms. Manisha C. Petare whose comments and suggestions helped in publishing the paper.

REFERENCES

[1]Nuance Dragon Assistant http://www.nuance.com/dragon/dragon-assistant/index.html
[2]Syn Virtual Assistanthttp://download.cnet.com/Syn-Virtual-Assistant/3000-2346_4-75910908.html
[3]Braina (Brain Artificial) is an intelligent personal assistant softwarehttp://www.brainasoft.com/braina/
[4]Google now features http://lifehacker.com/top-10-awesome-features-of-google-now-1577427243
[5]Sirihttp://en.wikipedia.org/wiki/Siri
[6]Cortana Windows phone 8.1 personal assistant http://www.windowsphone.com/en-us/features-8-1
[7]http://api.wunderground.com
[8]http://news.google.com