DeskAss – An Intelligent Desktop Assistant for Windows Application Users

Adewale Abdulrasaq SHOMOPE [1], Jerry Abayomi SARUMI [2]
1,2 Computer Science Department, Lagos State University of Science & Technology
shomopelaspocomputer@gmail.com; 08033581398

I. ABSTRACT

User experience, the very concept overseen by Human Computer Interface field of Computer Science has become the most critical aspect of any system design, be it hardware or software or the combination of the two. Windows, Linux, Apple, Android, Blackberry, etc. are various famous and commonly used Operating Systems across the globe today.

Today, Microsoft's Cortana, Apple's Siri, and Google Assistant by Google are competing in the process of delivering virtual assistant services for desktop and/or mobile users. As much awesome as the features of these systems are, there are limitations, the most significant one being their uselessness when there is no internet connectivity. This project aims to bring these features to Windows operating system users while enabling the users to as well enjoy the system even when there is no internet connection.

DeskAss – the final system of this project provides eminent control over all applications and software installed on the computer and automates all the required tasks through voice on a single go. It takes the client contribution to type of voice or content and procedure it and returns the yield in different structures like activity to be performed or the query item is directed to the end client. What's more, this proposed framework can change the method for associations between the end client and the work area. The framework is being planned so that every one of the administrations given by the desktop is open by the end client on the client's voice directions.

Keywords: SIRI, Google Voice Search, Virtual Assistant, Artificial Intelligence, Smart Assistant, Desktop Assistant, Voice Recognition, Text To Speech, Python, Python AI Modules

II. INTRODUCTION

Over the years, technology has evolved in helping us get things done in ways that surpass our previous imaginations. We have seen a variety of technological advancements take over the tasks that are naturally done by man. In the center of this is the innovation of the era in computing – Artificial Intelligence. AI or Artificial Intelligence has gone on to prove to be useful in wide areas of applications.

Among the leading use-cases of Artificial Intelligence are robotics and virtual assistants. With these two, the way

business is run in the world has changed completely. The very application of virtual assistant began on a low, very basic - starting from Chabot and other online applications that offer an exceptional virtual experience to the users to heavy software that is almost as intelligent as a man such as Androids. Today, the application of Artificial Intelligence has been identified in almost all areas of life. And among these is having a companion that helps you get things done on your personal computer, minimizing your use of a keyboard and mouse as much as possible.

In this work, I look forward to bringing to existence, an intelligent virtual assistant that aids us in using our personal computer. This project is hence termed "Intelligent Desktop Assistant".

The idea behind this project – Intelligent Desktop Assistant, with its software end product named *DeskAss* is to bring about a free, yet effective, reliable, and easy-to-use virtual assistant for Windows operating system users.

III. EXISTING RELATED APPLICATIONS

Siri:

Siri is Apple Inc.'s virtual assistant that uses voice queries and a natural-language user interface to answer questions, make recommendations, and perform actions by delegating requests to a set of internet services. The software adapts to users' language usages, searches, and preferences, with continuing use. Returned results are individualized.

Siri can make calls or send texts for you whether you are driving, have your hands full, or are simply on the go. [1] It can even announce your messages on your AirPods. It also offers proactive suggestions — like texting someone that you're running late for a meeting — so you can stay in touch effortlessly.

Cortana:

[3] Cortana is a virtual assistant developed by Microsoft, which uses the Bing search engine to perform tasks such as setting reminders and answering questions for the user. It is a desktop assistant that can set reminders, recognize natural voice without the requirement for keyboard input, and answer questions using information from the Bing search engine that is then displayed on Microsoft Edge, both of which are owned by the same company. The information that can be gotten with the help of Cortana includes weather and traffic conditions, sports scores, biographies).

Google Assistant:

Google Assistant is an artificial intelligence–powered virtual assistant developed by Google that is primarily available on mobile and smart home devices. It was first launched as an extension of Google Now, designed to be personal while expanding on Google's existing "OK Google" voice controls.

Google Assistant can run timers, reminders, and other apps on mobile phones, connect you and control your smart home, connect you to the internet and fetch information for you as needed anytime.

As well, the Assistant can engage in a two-way conversation, using Google's natural language processing algorithm. Search results are presented in a card format that users can tap to open the page. In February 2017, Google announced that users of Google Home would be able to shop entirely by voice for products through its Google Express shopping service, with products available from Whole Foods Market, Costco, Walgreens, PetSmart, and Bed Bath & Beyond at launch, and other retailers added in the following months as new partnerships were formed.

IV. OBSERVATION AND INTERVIEW CARRIED OUT

Since I am an addicted user of the computer system; Windows operating system to be precise, and as well, I am a budding programmer, gathering data becomes a very easy task for me. I made researches on the operating system-oriented functionalities that the users of iOS have access to but that are not available on Windows in the case of Siri, see how Cortana works on Windows10 and studied the features of Google Assistant – all of which have some features intended to be improved by this project.

I then had some serious discussions with experienced programmers, especially those that are into Artificial Intelligence software development. I learned of the most suitable programming language and easy-to-use Independent Development Environment for this project purpose. As well, I have seen some tutorial videos on YouTube on Python programming, using Python libraries, the available frameworks, and calling the methods all of which are essential to the success of this project.

V. ANALYSIS OF THE STUDY

Intelligent Desktop Assistant is among the software applications that utilize the use-cases of Artificial Intelligence [4] – the branch of software development that proves to be the answer for future technological needs. Verily, the application can become available for use for Windows operating system users in the country.

The DeskAss software uses the “Pyttsx3” and “voice recognition” methods from the Python libraries to listen to the command passed by the user, recognize what it means in text and analyze it. If the instruction is to do an offline task like play certain music or open a file, the conditional statement that checks for the task requested has within its block, what to do as I have defined for the method. The condition is checked and the one that

is met is carried out within a blink of an eye.

For online related tasks such as search the profile of a person from Wikipedia, or other web-based tasks, the "Wikipedia" and "web browser" methods are called upon. These then fetch the needed information from the internet using the defined browser and speak out the answer immediately.

DeskAss can set a reminder, tell you the time of the day, greet you, open a file, send an email, read an email, access search engines like Google for information, and give you directions using the GoogleMap method. There are a lot of responsibilities on DeskAss's shoulder and it gets things done for you as quickly as possible.

VI. THE NEEDED TOOLS FOR THIS APPLICATION

1. Python Programming

[3] Despite being a general-purpose language, Python has made its way into the most complex technologies such as Artificial Intelligence, Machine Learning, Deep Learning, and so on. While this language is not necessarily the only one capable of being used to develop Artificial Intelligence software, there are reasons why a lot of developers out there prefer it. Some of these reasons were studied by me, and in the end, I found out that it is indeed the best that I could use to achieve the goals and objectives of this project.

Artificial Intelligence applications like our Intelligent Desktop Assistant (DeskAss). However, thanks to Python's support for pre-defined packages, we don't have to code algorithms. And to make things easier, Python provides a "check as you code" methodology that reduces the burden of testing the code.

What is more, there are 100s of pre-built libraries to implement various Machine Learning and Deep Learning algorithms. All I have to do is to install and load the necessary packages with a few commands. Examples of pre-built libraries include NumPy, Keras, Tensorflow, Pytorch, and so on.

2. Python Virtual Assistant Libraries

As mentioned above, Python has some readymade libraries from which we can get the methods needed in our application program. Some of them are:

Speech Recognition Package: There is a need for something that can capture the instructions being given by the user, hence, the need for the SpeechRecognition package. This package allows Python to access audio from your machine's microphone, transcribe audio, save the audio to an audio file, and other similar tasks.

Text to Speech Package: After receiving the instructions through voice command, the Desktop Assistant will need to convert the voiced question to a text to process the output. Once the assistant looks up an answer online (for instructions that involve searching information from the

internet) it will need to convert the response into a voiceable phrase. For this purpose, we'll use the TTS package (Google Text-to-Speech).

Audio Playback Package: All that's left is to give voice to the answer. The mpyg321 package allows for Python to play MP3 files. This is very handful as the Assistant will often need to access the music files on the computer and play them as instructed.

Pyttsx3: This is a Python voice engine module that has some preinstalled sample voices in a form of a list that can be used by any programmer for AI programs.

VII. UNDERSTANDING VIRTUAL ASSISTANT SYSTEM

Virtual Assistants based on Artificial Intelligence connect the bridge between humans and computers, giving us the ability to talk to machines with our natural language and getting them to perform tasks for us.

Here is a working scenario. Through an immediate and deep understanding of what customers mean (and not just what they say) in their own words, chatbots enable a human-like dialogue, much like the one that you would have with a live agent. The customer asks a question, and the virtual agent understands the question and responds with the answer. And it all happens from the comfort of the user's home or wherever they happen to be, whenever they need it. Naturally, this is a major advantage for the customer.

There are many advantages for the companies offering this service to customers, beyond the obvious customer experience advantages. Companies can go a step further to offer personalized, tailored services based on customer preferences, to respond to needs that may be the customer hasn't yet asked for.

They also reduce operational costs. By handling the majority of customer interactions (which would have traditionally been managed by an agent in a call center), operators are freed up to manage more complex tasks and to extract strategic information from the conversations, to pinpoint targeted and specific customer requirements (not just general trends). In addition, a virtual assistant is enough smart to perceive when the customer needs support from a human agent and can make this transition happen in a seamless way.

VIII. SYSTEM ARCHITECTURE

1. User Application: The user accesses the application through the Graphics User Interface created for user interactions with the application. In the case of our DeskAss, there is an icon, in form of a listening microphone that will be clicked by the user for the program to initialize. Once the program is initialized, it will welcome the user with a greeting message and asks what the user needs it

to do. Hence, the speech support comes in.

2. Speech Recognition and Support: The Asynchronous Speech Recognition (ASR) receives an audio stream with a spoken sentence, and the name of the grammar to be used to recognize the speech. There are two kinds of grammars: GRXML, which is a W3C standard to specify the words or sentences to be recognized by the ASR, and ARPA, a statistical language model. The first type is more limited regarding the number of sentences that can be recognized and are manually defined but can return tags identifying the sentence's meaning. For ARPA, the creation of the grammar is automatic, since it is a statistical language model, but it requires large amounts of text to create the model, as well as the mechanisms to extract the meaning of the sentences.

3. Speech Synthesis: For this part of the service, called Text-to-Speech (TTS), the application sends a message with the information to be read to the user, the method to use to synthesize it to speech, using the Microsoft Speech Platform (MSP), and the chosen voice. The service accepts other parameters such as speech volume and rate. The rate parameter defines the speed of the speech. Based on recent experiments in our group the default value chosen for the speed parameter makes the speech understandable for the elderly, and if the value increases, elderly people may have more difficulty in understanding it. The service returns an audio stream containing the spoken sentence.

4. Integration with the framework: As the speech instruction passed to the multimodal framework for recognition and analysis, the integration framework handles the tasks to be performed and calls the necessary methods imported and defined for the actual command passed with the predefined keywords.

IX. SYSTEM DESIGN AND IMPLEMENTATION

Following the design of the proposed system, the implementation phase comes in. Implementation brings the design to life. The user, having used the related existing systems such as Microsoft Cortana, Apple's Siri, and Google Assistant, hence tests the newly designed system; DeskAss to see the improvements that the new system proposes. Two activities are faced by me as the designer of the new system, just as it is for other developers. These are the testing phase and the debugging. During testing, I, as the developer test run the program on multiple occasions to experience both foreseeable and unexpected bugs that usually come with new designs. The debugging, on the other hand, allows me to remove the bugs, thereby giving room for the development of a reliable, efficient, and robust system that can meet users' requirements or expectations.

The application has a single user interface. This window displays an

Artificial Intelligence image placed on row 1 of the rows-pan, a property of Python;'s Tkinter module for user interface design. Following it is a button that has label properties. The default text is "Initializing…" which then changes to "Click to start"; a command that starts program execution as soon as the user clicks the button. The text changes to "Recognizing" which then tries to take command from the computer microphone.

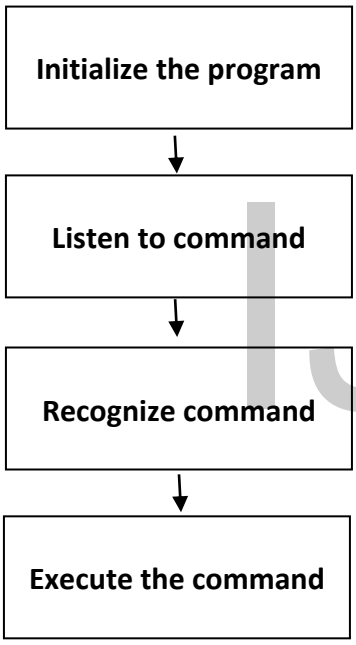

Fig. 1: The program flow

X. RESULT AND CONCLUSION

The result of this system is a very intelligent desktop assistant application that allows Windows operating system users to make access to a virtual assistant and get things done by using voice command for both offline and online services. This application is built to offer a wide range of compatibility for all Windows operating systems although might not work well enough for Windows XP and other previous versions. It allows Windows operating system users to access the voice recognition functionalities enjoyed for online services as seen on Apple operating system while offering the offline services as well.

This application allows full utilization of voice recognition module and Pyttsx3 voice engine from Python libraries to access virtual assistant system for Windows operating system users. With these, functionalities such as the opening of Windows system programs such as calculator, calendar, email, etc., launching of user's installed program such as Microsoft Office applications, Coreldraw, internet browsers, etc., fetching information from Wikipedia just by mentioning the world "Wikipedia" in the command, sending of email by voice command, searching for information on Google, opening social media website and many more can be done by commanding the application. The application, when launched is always in a ready state, listening to the command, recognizing it, and performing the task immediately.

While the system meets the aim of the developer, there is always room for improvement. Just like existing systems have their limitations, there are things that this system cannot do yet. I plan to make this bigger than it already is. More functionality will be added later in the future.

In using this system, it is recommended that you have a good microphone gadget. This could be a wired or wireless device. As well, you can use your mobile phone as a microphone. Learn how to connect your phone as a PC microphone if you need to use it. As well, the system only works perfectly in a silent environment. Be sure to move away from the noise to use it.

REFERENCES

[1] https://www.apple.com/siri/

[2] Mehrotra, R., Awadallah, A. H., Kholy, A. E., & Zitouni, I. (2017, August). Hey Cortana! Exploring the use cases of a Desktop based Digital Assistant. In Sigir 1st international workshop on conversational approaches to information retrieval (cair'17) (Vol. 4).

[3] Zulaikha L. (2020) A Comprehensive Guide To Artificial Intelligence With Python.

[4] Ahmed H. A., Ahmed E. K., and Imed Z. (2015). Exploring the use cases of a Desktop based Digital Assistant. Microsoft, Microsoft Research Redmond, WA, USA