Voice Assistant Using Python

[1]Rakshit Verma, [2]Rudraksh Goyal, [3]Mansi

Department of Computer Science & Application
School of Engineering & Technology
Sharda University
Greater Noida, India

***Abstract* - The advent and widespread use of the Internet of Things has eased the active adoption of artificial intelligence technology in human life (IOT). The way autonomous machines communicate with one another and with humans is evolving. As a result of new capabilities, numerous solutions for integrating intelligent objects into social networks of the Internet of Things have been developed. The ability of artificial intelligence to recognise human language naturally is one of the important trends in the field. New understandings in this area may result in new techniques for organic human-machine communication, where the latter would train the former to comprehend human speech and interact accordingly. Voice assistant is one of these tools, and it can be incorporated into many other intelligent systems. The basic operating principles of voice assistants are discussed in this essay, along with some of their key drawbacks and restrictions. It is explained how to create a local voice assistant without needing cloud services, thus expanding the range of uses for such gadgets in the future. Software agents known as voice assistants can understand spoken language and respond with synthetic voices. The most well-known voice assistants are Google Assistant, Apple's Siri, Amazon's Alexa, Microsoft's Crotona, and Apple's Siri, and they are all built into smartphones or specific home speakers. Users can use voice commands to manage other common chores like email, to-do lists, and calendars in addition to asking their assistants questions, controlling home automation devices, and controlling media playback. This essay will examine the fundamental operations and prevalent features of voice assistants in use today.**

***Keywords* - Deep Learning, Artificial Intelligence, Virtual Assistant, Voice Assistant, Speech Recognition**

I.INTRODUCTION

Gone are the days when humans depended on other humans for help or services. The digitalization of the world made sure that humans no need to contact anyone else to seek help, they could depend on a far more efficient and reliable device which can take care of their everyday needs. The computers, mobiles, laptops, etc., became a part of us and our daily life, It could carry out simple calculations to complex programs to reduce monotonous work and waste of manpower. Virtual Personal Assistant has almost become a basic necessity in all electronic devices so as to execute the required problems easily. More than just being a bot , VPA can make life easier for the user in various ways. Speech recognition is one of the relatively new integration into the VPA. But, though its moderately efficient, it is not very helpful and are not used by the user due to its high amount of error. Though the error percentage of the upcoming VPAs is around 5 percent, it still is not quite up to the mark to where it becomes a basic part of the user life. Thus the projects aim is to build a VPA with speech recognition which has a very minimal error percentage. The intricate process of voice recognition makes use of cutting-edge ideas like neural networks and machine learning. A neural network with vectors for each letter and syllable is constructed once the aural input has been analysed. The data set refers to this. When someone speaks, the machine compares it to this vector and selects the syllables that have the highest correspondence.

A voice assistant can be a computer assistant that listens for specific speech instructions and responds by using synthesis, language processing algorithms, and human voice. Voice assistants will

provide relevant information by listening for specific phrases and removing background noise. They may also carry out specific tasks as requested by the user supported commands, also known as intents. While voice assistants may be entirely a software system on which all devices can be combined, other assistants are specifically designed for each individual device use, such as the Amazon Alexa clock. Voice assistants are now integrated into some of the gadgets we use every day, such as smartphones, PCs, and high-quality speakers.[1] The widespread use of AI in consumers' daily lives is also reviving the shift toward voice. Siri from Apple, Alexa from Amazon, Cortana from Microsoft, Google Assistant from Google, and the recently released intelligent assistant known as AIVA are some of the most well-known voice assistants. The architecture and design of voice assistants are briefly discussed in this paper. To create a smart assistant that can handle IoT applications and even resolve user inquiries utilizing online searches, voice controlled personal assistant systems will employ natural language processing and can be coupled with artificial intelligence techniques. It may be made to require less work from people when interacting with numerous different subsystems than would otherwise be necessary if done manually. The system will improve the quality of human existence by achieving this.[2] More precisely, this system is made to communicate intelligently and operate other subsystems, such as Internet of Things (IoT) devices or gadgets that receive news from the Internet, provide other information, or receive previously saved personal data. The Android app should allow users to add information like calendar entries, alarms, or even reminders.

The system will consist of the following phases: voice data collection; voice analysis and text conversion; data storage and processing; and speech production from the output of the text processing.[3] Every phase generates data, which may be utilized later to identify trends and provide user suggestions. Artificial intelligence devices that learn and comprehend people may use this as a primary foundation. Thus, based on a review of the literature and an analysis of the current system, we have determined that the suggested system would not only make it easier to interface with other systems and modules but will also keep us organized.

II. PROPOSED PLAN OF WORK

Analyzing the audio commands issued by the user through the microphone was the first step in the process. This could involve anything, including obtaining information or using a computer's internal storage. This study is based on reading the aforementioned material and putting its examples to the test. With the express purpose of discovering best practises and a deeper understanding of voice assistant, tests are made by programming in accordance with books and internet resources.[4] We require a voice assistant that can not only accept voice commands but also carry out the appropriate instructions and output them in speech or another format. This article introduces a personal voice assistant that accepts commands in a discrete manner. This is accomplished through the use of a synchronous procedure. [5] The primary cost in this project is the price of paperwork. Microphones and speakers must also be purchased by users. They are, once again, inexpensive and widely available. It is relatively inexpensive.

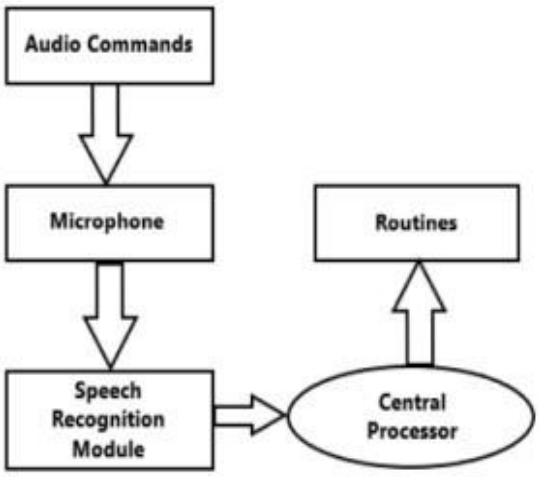


Fig. 1. Proposed plan of work

Figure depicts the voice assistant's core procedure workflow. The speech input is converted to text using speech recognition. The central processor then receives this text and determines the nature of the command before calling the appropriate script to run. The difficulties don't end there, though.[6] The software may not be able to understand you even after hundreds of hours of input because of other reasons. A speech recognition system might easily become confused by background noise. This is because it lacks the ability to tell the difference between your voice and the background noises it "hears," such as a dog barking or a helicopter buzzing overhead. Engineers must programme the device with such capability; they collect data on these background noises and "teach" the device to filter them out.[7] Speech recognition systems can be sensitive to these pitch shifts since humans typically adjust their voice pitch to compensate for noisy situations.

III. SYSTEM ARCHITECTURE

This project's system design demonstrates how control moves throughout the system. Additionally, it displays the software and hardware needed for the program's execution. The following is the architecture diagram:

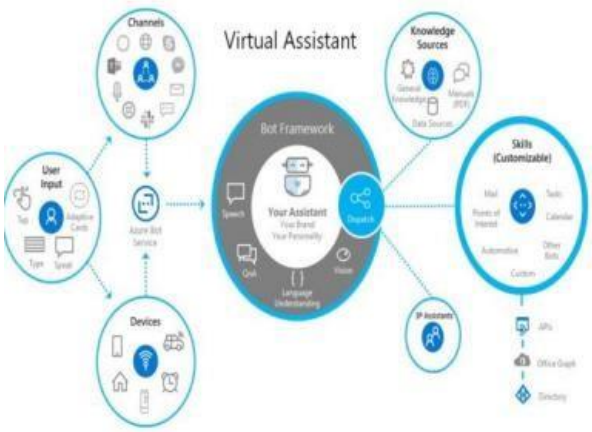


Fig. 2. System Architecture

The steps of the overall system design are as follows:

1. Voice-activated data collecting.

2. Voice analysis and text conversion
3. Data processing and storage
4. Creating voice from the result of text processing [8]

IV. METHODOLOGY

With the aid of sapi5 and pyttsx3, we first enable system voice for our program. Natural language processing (NLP) is used by virtual assistants to convert text or voice input into executable commands. When a user asks her personal assistant a question in order to finish a task, the natural language impulses are translated into executable instructions or digital data that software can examine [9]. Then, to determine the correct solution, compare this data with the information from the software. Virtual assistants are programs that run machines using their own commands.[10] Using Python installation packages such as Voice Recognition, gTTS, and Pipwin, you can create virtual assistants. The process of transforming speech to text is known as speech recognition. Voice assistants like Alexa and Siri frequently use this [11]. Python includes a Speech Recognition API. It is capable of converting voice or audio instructions to text for any further processing. The user issues a command to that of an interactive device, such as a laptop or PC, in the diagram above. These dialogue devices are capable of listening and recognizing commands. Combine this command with the cloud storage that you already have to further evaluate the process [12]. When a match request matches cloud data, it generates text and audio output. Find a function or logic to execute based on the request and transmit the backend process's result as the response. [13] A Python textto-speech conversion library is called pyttsx3. It is compatible with Python 2 and 3 and works offline, unlike competing libraries. Microsoft created the Speech Application Programming Interface, or SAPI, as an API to enable the use of speech synthesis and recognition within Windows programs.

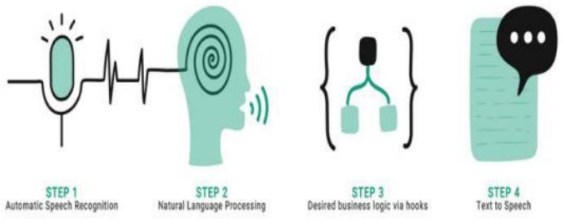


Fig. 3. Methodology

Then, all of the program's capabilities are defined in the main function. The following features are intended for the proposed system:

1. The assistant solicits input from the user and continues to watch for commands. According to the needs of the user, the listening period can be customized.
2. If the assistant doesn't understand the command clearly, it will keep asking the user to repeat it.
3. The user can choose between a male or female voice for this assistant, depending on their preferences.[14]
4. The assistant's current version offers functions like monitoring weather updates, sending and checking emails, searching Wikipedia, opening applications, checking the time, taking notes, showing notes, opening and closing applications, and opening and closing YouTube and Google. [15]

A. IMPORTED MODULES

1. Module for Speech Recognition - The ability for your assistant to recognise your voice is one of the most important aspects because we're developing a voice assistant app. To install this module, type the following command into the console.[16]
2. DateTime component - Date and time are displayed using the Datetime package. Python already includes a built-in datetime module.
3. Wikipedia - We have utilised the Wikipedia module in our project to get more information from Wikipedia or to do a Wikipedia search because, as we all know, Wikipedia is a great and enormous source of knowledge, just like GeeksforGeeks or any other sources. Use pip install wikipedia to install this Wikipedia module.[17]
4. Webbrowser - To conduct an online search. Python already includes this module.
5. OS - Python's OS module offers tools for communicating with the operating system. OS is included in the basic utility modules for Python. Utilizing operating system-dependent functionality is made possible by this module.[18]
6. Pyaudio - Pyaudio is a cross-platform C++ library that interfaces with audio drivers. PyAudio is a set of Python bindings for PortAudio.[19]
7. PyQt5 - A complete set of Python bindings for Qt v5 is available as PyQt5.On all platforms that are supported, including iOS and Android, it enables Python to be used as an alternative to C++ for application development. It is implemented as more than 35 extension modules. Apps built using C++ may also incorporate PyQt5 so that users can customise or improve the functionality of those applications.[20]
8. Python Backend - The voice output is obtained by the Python backend, which then determines whether the command or the speech output is an API call and context extraction. The output is then returned to the Python backend so that the user can receive the desired output.
9. Text-to-speech programme - The term "text-to-speech" (TTS) describes a feature that allows computers to read text aloud. Written text is converted by a TTS Engine into a phonemic representation, which is subsequently translated into waveforms

that can be output as sound. Through third-party publishers, TTS engines with various languages, dialects, and specialised vocabularies are accessible.[21]

10. Text to speech conversion - Speech recognition software is employed to translate spoken input into text output. It decodes the voice and translates it into text that the computer can easily understand.[22]
11. Content Removal - The automatic extraction of structured data from unstructured and/or semi-structured machine-readable materials is known as context extraction (CE). This activity typically involves employing natural language processing to process documents written in human languages (NLP). Recent developments in the processing of multimedia documents, such as automatic annotation and content extraction from photos, audio, and video, could be viewed as the results of context extraction tests.[23]
12. Production of text - It conducts the action, decodes the voice command, and then displays the voice command as textual output in the terminal.[24]

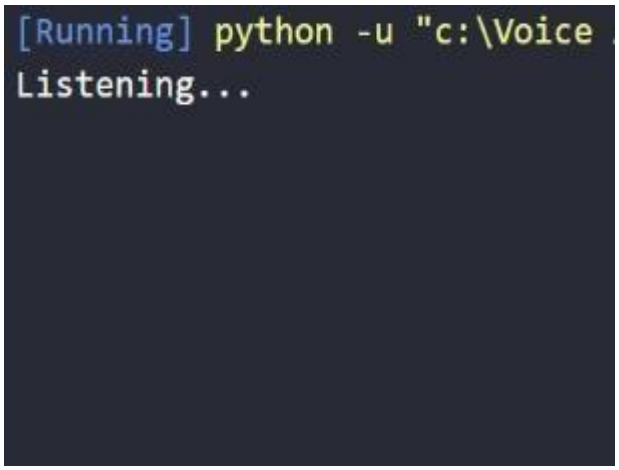
B. DATA FLOW SEQUENCE

1. Device Initialize: Device initialization is accomplished by invoking the device's name.
2. Task manager: Task Manager handles Text-to- Speech and Speech-to-Text conversion
3. Service Manager: Command analysis and cloud server and web service adapter matching.
4. Execute Command: After determining whether the command is a match, launch the corresponding Python script [25]

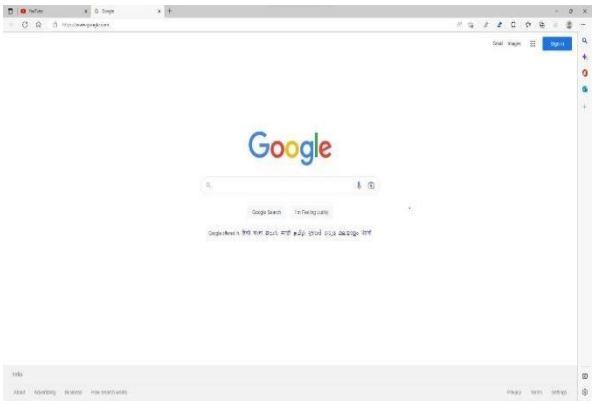
V. EXECUTION

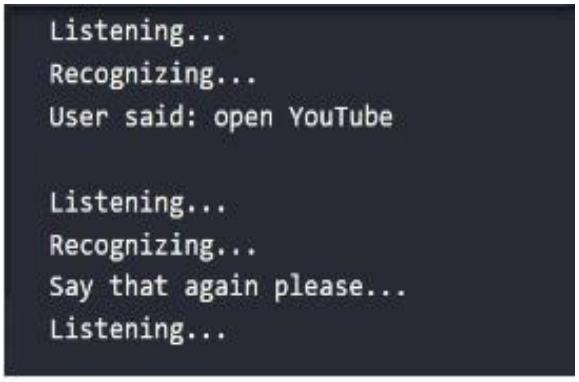
When the assistant first starts, it will wait for the user to provide input. When a voice command is given to the assistant, it will be recorded and searched for using the keyword that was included in the command. If the assistant was successful in locating a keyword that was associated with the given command, it will carry out the task in accordance with the input and return the output to the user in both voice and text form in the terminal window. If not, the assistant will once more begin to wait for the user to provide accurate information. Each of these functions plays a unique role in how well the entire system functions.

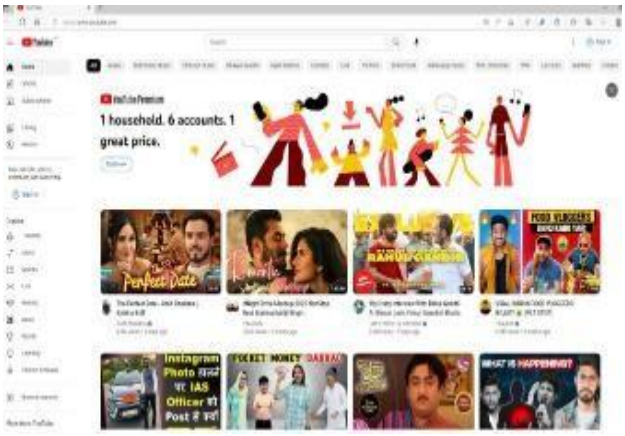
```
import pyttsx3 #pip install pyttsx3
import speech_recognition as sr #pip install speechRecognition
import datetime
import wikipedia #pip install wikipedia
import webbrowser
import os
import smtplib

engine = pyttsx3.init('sapi5')
voices = engine.getProperty('voices')
# print(voices[1].id)
engine.setProperty('voice', voices[0].id)


def speak(audio):
    engine.say(audio)
    engine.runAndWait()


def wishMe():
    hour = int(datetime.datetime.now().hour)
    if hour>=0 and hour<12:
        speak("Good Morning!")

    elif hour>=12 and hour<18:
        speak("Good Afternoon!")

    else:
        speak("Good Evening!")

    speak("Hey Buddy. Please tell me how may I help you")

def takeCommand():
```


[Running] python -u "c:\Voice
Listening...


Listening...
Recognizing...
User said: open Google
Listening...
Recognizing...
Say that again please...
Listening...


Google


Listening...
Recognizing...
User said: open YouTube
Listening...
Recognizing...
Say that again please...
Listening...



VI.CONCLUSION

This paper provides a thorough description of the design and implementation of a Static Voice-enabled personal computer assistant in Python. In comparison to earlier times, the voiceactivated personal assistant of today will be more efficient at saving time and being supportive of people with disabilities. This Assistant effectively completes some tasks that the user has assigned. Additionally, this assistant can perform a lot of things with just one voice command, including sending messages to the user's mobile device, automating YouTube, and obtaining data from Google and Wikipedia. It not only responds to human orders but also to questions posed by the user or words said by the user, such as opening operations and tasks. It welcomes the user in a way that makes them feel at ease and allows them to communicate with the voice assistant freely. The programme should also get rid of any extra physical labour that is required for the user to carry out each and every operation. With the help of this voice assistant, we were able to automate several services with just one line of code. It makes most user tasks, including web searches and other activities, easier. Our goal is to create this project a fully functional server assistant that is intelligent enough to take the place of a traditional server administrator. The PyCharm community has supported the project's use of open source software components, which can quickly accommodate any modifications. This project's modular design gives it greater flexibility and makes it simple to add new features without impairing existing system functionality. The system will consist of the following phases: voice data collection; voice analysis and text conversion; data storage and processing; and speech production from the output of the text processing. Every phase generates data, which may be utilized later to identify trends and provide user suggestions. Artificial intelligence devices that learn and comprehend people may use this as a primary foundation. Thus, based on a review of the literature and an analysis of the current system, we have determined that the suggested system would not only make it easier to interface with other systems and modules but will also keep us organized.

REFERENCES

1. DOUGLAS O'SHAUGHNESSY, SENIOR MEMBER, IEEE, "Interacting With Computers by Voice: Automatic Speech Recognition and Synthesis" proceedings of THE IEEE, VOL. 9
2. Kei Hashimoto1, Junichi Yamagishi2, William Byrne3,Simon King2, Keiichi Tokuda, "An analysis of machine translation and speech synthesis in speech-to-speech translation system" proceedings of 5108978-1-4577-0539-7/11/$26.00 ©2011 IEEE.
3. Nil Goksel-Canbek2 Mehmet Emin Mutlu, "On the track of Artificial Intelligence: Learning with Intelligent Personal Assistant" International Journal of Human Sciences, 13(1), 592601. doi:10.14687/ijhs.v13i1.3549.
4. Harshita Phatnani, Mr. Jyotiprakash Patra and Ankit Sharma' "CHATBOT ASSISTING: SIRI" Proceedings of BITCON-2015 Innovations For National Development Natinal Conference on : Research and Development in Computer Science and Applications, E-ISSN2249–8974.
5. Sutar Shekhar, Pophali Sameer, Kamad Neha, Prof. Devkate Laxman, " An Intelligent Voice Assistant Using Android Platform" , March 2015, IJARCSMS, ISSN: 232 7782
6. VINAY SAGAR, KUSUMA SM, "Home Automation Using Internet of Things", June-2015, IRJET, e-ISSN: 2395 -0056
7. DOUGLAS O'SHAUGHNESSY, SENIOR MEMBER, IEEE, "Interacting With Computers by Voice: Automatic Speech Recognition and Synthesis" proceedings of THE IEEE, VOL. 91, NO. 9, SEPTEMBER 2003
8. Nil Goksel-Canbek2 Mehmet EminMutlu, "On the track of Artificial Intelligence: Learning with Intelligent Personal Assistant" International Journal of Human Sciences, 13(1), 592601. doi:10.14687/ijhs.v13i1.3549.
9. Easwara Moorthy, A., Vu, K.-P.L.: Privacy Concerns for Use of Voice Activated Personal Assistant in the Public Space. International Journal of Human- Computer Interaction 31, 307–335 (2015)
10. Tsiao, J.C.-S., Tong, P.P., Chao, D.Y.: Natural- Language Voice-Activated Personal Assistant, United States Patent (10), Patent No.: US 7,216,080 B2 (45), 8 May 2007.
11. Lopez, G., Quesada, L., Guerrero, L.A.: Alexa vs Siri vs Cortana vs Google assistant: a comparison of speech-based natural user interfaces. Conference Paper, January 2018
12. Kepuska, V., Bohouta, G.: Next generation of virtual personal assistants (Microsoft Cortana, Apple Siri, Amazon Alexa and Google Home). In: IEEE Conference (2018)
13. Gong, L.: San Francisco, CA (US) United States US 2003.01671.67A1 (12) Patent Application Publication c (10) Pub. No.: US 2003/0167167 A1 Gong (43) Pub. Date: 4 September 2003 for Intelligent Virtual Assistant

14. Sumitkumar Sarda, Yash Shah, Monika Das, Nikita Saibewar, ShivprasadPatil,"VPA: Virtual Personal Assistant" Published in 2017 International Journal of Computer Applications (0975 – 8887) Volume 165 – No 1.
15. Alexa Blogs, "The Story of the Capital One Alexa Skill," Zoey Collier, December 16, 2016, https://developer.amazon.com/blogs/alexa/post/c70e 3a9b-405c-4fe1-bc20-bc0519d48c97/the-story-of-the-capital-onealexa-skill.
16. BBVA website, "Efma recognizes Garanti Bank's mobile voice assistant," Luz Fernández Espinosa, February 17, 2017, recognizes-garanti-banks-mobile-voice-assistant/
17. Bill Selak. 2017. (July 2017). Amazon Alexa in the Classroom. Retrieved February 14, 2019 from http://www.coolcatteacher.com/amazon-alexa-classroom/
18. awar, N., & Bendle, N. (2018). Marketing in the Age of Alexa. Harvard Business Review, 96(3), 80-86.
19. Han, S., & Yang, H. (2018). Understanding adoption of intelligent personal assistants: a parasocial relationship perspective. Industrial Management & Data Systems, 118(3), 618636.
20. Jiang, J., Hassan Awadallah, A., Jones, R., Ozertem, U., Zitouni, I., Gurunath Kulkarni, R., & Khan, O. Z. (2015, May). Automatic online evaluation of intelligent assistants. In Proceedings of the 24th International Conference on World Wide Web (pp. 506-516). International World Wide Web Conferences Steering Committee.
21. Milhorat, P., Schlogl, S., Chollet, G., Boudy, J., Esposito, A., & Pelosi, G. (2014). Building the Next Generation of Personal Digital
22. Neelay Trivedi. 2018. Problem Pal: Generating Autonomous Practice Content in Real-Time with 6 Voice Commands and Amazon Alexa. In Proceedings of E-Learn: World Conference on E- Learning. Las Vegas, 80-82.
23. Schweitzer, F., Belk, R., Jordan, W., & Ortner, M. (2019). Servant, friend, or master? The relationships users build with voice-controlled smart devices. Journal of Marketing Management, 1-23.
24. Tearsheet, "Inside the development of Erica, Bank of merica's AI-powered bot," Tanaya Macheel, July 28, 2017, https://www.tearsheet.co/artificial- intelligence/inside-thedevelopment-of-erica-bank- of-americas-ai- powered-bot [25] Kei Hashimoto1, Junichi Yamagishi2, William Byrne3,Simon King2, Keiichi Tokuda, "An analysis of machine translation and speech synthesis in speech-to-speech translation system" proceedings of 5108978-1-4577-0539-7/11/$26.00 ©2011 IEEE