Text – To – Speech Synthesis (TTS)

Article · May 2014

CITATIONS
5

READS
15,759

3 authors:

Ifeanyi Cosmas Nwakanma
Federal University of Technology Owerri
95 PUBLICATIONS **113** CITATIONS
SEE PROFILE

Ikenna Oluigbo
Claude Bernard University Lyon 1
8 PUBLICATIONS **14** CITATIONS
SEE PROFILE

Okpala Izunna
University of Cincinnati
13 PUBLICATIONS **17** CITATIONS
SEE PROFILE

International Journal of Research in Information Technology (IJRIT)

www.ijrit.com ISSN 2001-5569

Text – To – Speech Synthesis (TTS)

Nwakanma Ifeanyi[1], Oluigbo Ikenna[2] and Okpala Izunna[3]

[1]Assistant Lecturer, Department of Information Management Technology, Federal University of Technology Owerri, Imo State, Nigeria.
fraircos@yahoo.com

[2]Graduate Assistant, Department of Information Management Technology, Federal University of Technology Owerri, Imo State, Nigeria.
ikenna.oluigbo@gmail.com

[3]PG Scholar, Department of Information Management Technology, Federal University of Technology Owerri, Imo State, Nigeria.
izunna.okpala@yahoo.com

Abstract

Speech is one of the oldest and most natural means of information exchange between human. Over the years, Attempts have been made to develop vocally interactive computers to realise voice/speech synthesis. Obviously such an interface would yield great benefits. In this case a computer can synthesize text and give out a speech. Text-To-Speech Synthesis is a Technology that provides a means of converting written text from a descriptive form to a spoken language that is easily understandable by the end user (Basically in English Language). It runs on JAVA platform, and the methodology used was Object Oriented Analysis and Development Methodology; while Expert System was incorporated for the internal operations of the program. This design will be geared towards providing a one-way communication interface whereby the computer communicates with the user by reading out textual document for the purpose of quick assimilation and reading development.

Keywords: *Communication, Expert System, FreeTTS, JAVA, Speech, Text-To-Speech*

1. Introduction

Voice/speech synthesis is a field of computer science that deals with designing computer systems that synthesize written text. It is a technology that allows a computer to convert a written text into speech via a microphone or telephone. As an emerging technology, not all developers are familiar with speech technology. While the basic functions of both speech synthesis and speech recognition takes only minutes to understand, there are subtle and powerful capabilities provided by computerized speech that developers will want to understand and utilize.
Automatic speech synthesis is one of the fastest developing fields in the framework of speech science and engineering. As the new generation of computing technology, it comes as the next major innovation in man-machine interaction, after functionality of Speech recognition (TTS), supporting Interactive Voice Response (IVR) systems.
The basic idea of text-to-speech (TTS) technology is to convert written input to spoken output by generating synthetic speech. There are several ways of performing speech synthesis:

1. Simple voice recording and playing on demand;
2. Splitting of speech into 30-50 phonemes (basic linguistic units) and their re-assembly in a fluent speech pattern;

3. The use of approximately 400 diaphones (splitting of phrases at the centre of the phonemes and not at the transition).

The most important qualities of modern speech synthesis systems are its naturalness and intelligibility. By naturalness we mean how closely the synthesized speech resembles real human speech. Intelligibility, on the other hand, describes the ease with which the speech is understood. The maximization of these two criteria is the main development goal in the TTS field.

2. Objectives of the Study

The general objective of the project is to develop a Text-to-speech synthesizer for the physically impaired and the vocally disturbed individuals using English language. The specific objectives are:

1. To enable the deaf and dumb to communicate and contribute to the growth of an organization through synthesized voice.
2. To enable the blind and elderly people enjoy a User-friendly computer interface.
3. To create modern technology appreciation and awareness by computer operators.
4. To implement an isolated whole word speech synthesizer that is capable of converting text and responding with speech
5. To validate the automatic speech synthesizer developed during the study.

3. Scope of the Study

The study is focused on an ideal combination of a human-like behaviour with computer application to build a one-way interactive medium between the computer and the user. This application was customized using only one (1) word sentence consisting of the numeric digit 0 to 9 that could be used in operating a voice operated telephone system.

Human speech is inherently a multi modal process that involves the analysis of the uttered acoustic signal and includes higher level knowledge sources such as grammar semantics and pragmatics. This project intends to focus only on the acoustic signal processing without the incorporation of a visual input.

4. Significance of the Study

This project has theoretical, practical, and methodological significance:

The speech synthesizer will be very useful to any researcher who may wish to venture into the “Impact of using Computer speech program for brain enhancement and assimilation process in human beings”.

This text-to-speech synthesizing system will enable the semi-illiterates assess and read through electronic documents, thus bridging the digital divide. The technology will also find applicability in systems such as banking, telecommunications (Automatic system voice output), transport, Internet portals, accessing PC, emailing, administrative and public services, cultural centres and many others. The system will be very useful to computer manufacturers and software developers as they will have a speech synthesis engine in their applications.

5. Text – To - Speech Synthesis Defined

A speech synthesis system is by definition a system, which produces synthetic speech. It is implicitly clear, that this involves some sort of input. What is not clear is the type of this input. If the input is plain text, which does not contain additional phonetic and/or phonological information the system may be called a text-to-speech (TTS) system. A schematic of the text-to-speech process is shown in the figure 1 below. As shown, the synthesis starts from text input. Nowadays this may be plain text or marked-up text e.g. HTML or something similar like JSML (Java Synthesis Mark-up Language).

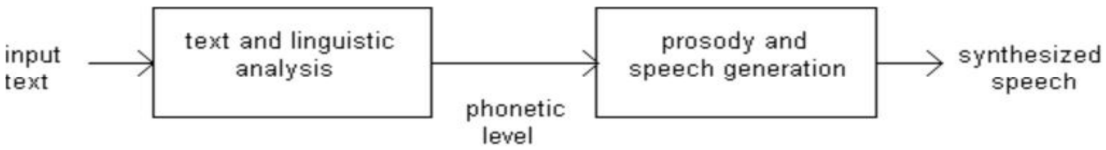

Figure 1: Schematic TTS

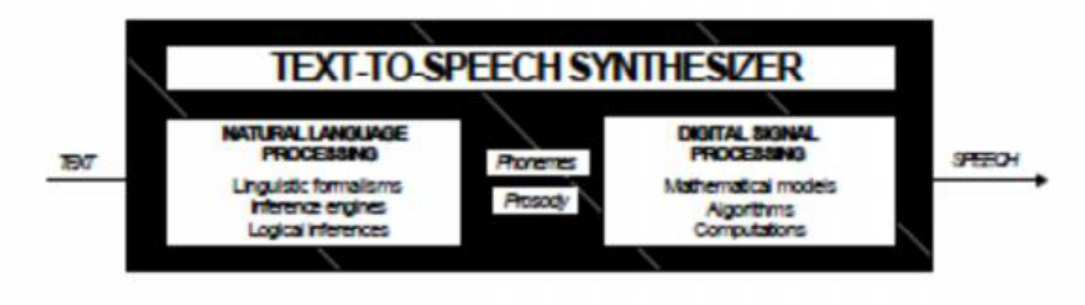

Figure 2: A general functional diagram of a TTS System.

5.1. Representation and Analysis of Speech Signals

Continuous speech is a set of complicated audio signals which makes producing them artificially difficult. Speech signals are usually considered as voiced or unvoiced, but in some cases they are something between these two. Voiced sounds consist of fundamental frequency (F0) and its harmonic components produced by vocal cords (vocal folds). The vocal tract modifies this excitation signal causing formant (pole) and sometimes anti-formant (zero) frequencies (Abedjieva et al., 1993). Each formant frequency has also amplitude and bandwidth and it may be sometimes difficult to define some of these parameters correctly. The fundamental frequency and formant frequencies are probably the most important concepts in speech synthesis and also in speech processing. With purely unvoiced sounds, there is no fundamental frequency in excitation signal and therefore no harmonic structure either and the excitation can be considered as white noise.
The airflow is forced through a vocal tract constriction which can occur in several places between glottis and mouth. Some sounds are produced with complete stoppage of airflow followed by a sudden release, producing an impulsive turbulent excitation often followed by a more protracted turbulent excitation (Allen et al., 1987). Unvoiced sounds are also usually more silent and less steady than voiced ones.
Speech signals of the three vowels (/a/ /i/ /u/) are presented in time-frequency domain in Figure 3. The fundamental frequency is about 100 Hz in all cases and the formant frequencies F1, F2, and F3 with vowel /a/ are approximately 600 Hz, 1000 Hz, and 2500 Hz respectively. With vowel /i/ the first three formants are 200 Hz, 2300 Hz, and 3000 Hz, and with /u/ 300 Hz, 600 Hz, and 2300 Hz.

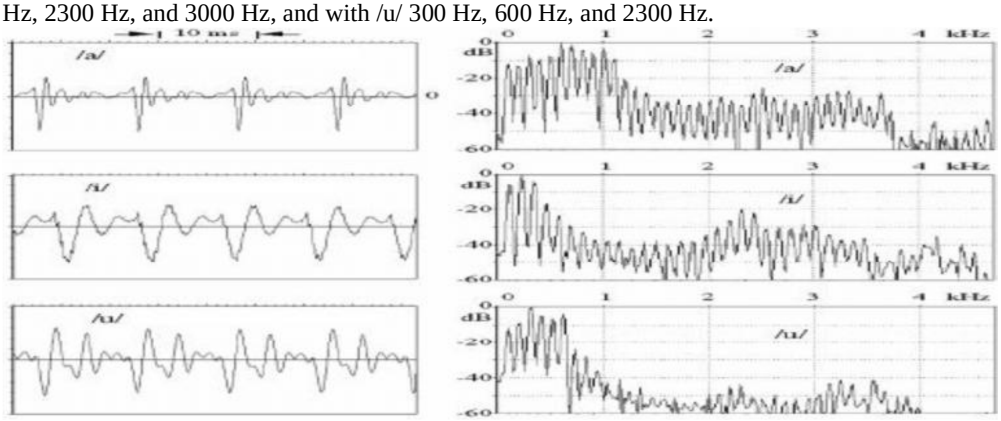

Figure 3: The time-frequency domain presentation of vowels /a/, /i/, and /u/.

6. Applications of Speech Synthesis

The application of synthetic speech is expanding fast whilst the quality of TTS systems is also increasing steadily. Speech synthesis systems are also becoming more affordable for common customers, making these systems suitable for everyday use. For example, better availability of TTS systems may increase employability for people with communication difficulties. Listed below are some of the applications of TTS system:
1. Applications for the Blind. 2. Applications for the Deafened and Vocally Handicapped
3. Educational Applications. 4. Applications for Telecommunications and Multimedia

5. Other Applications and Future Directions (e.g. Human-Machine Interaction)

7. Methodology and System Analysis

7.1 Analysis and Problems of Existing Systems

Existing systems algorithm is shown below in Figure 4. It shows that the system does not have an avenue to annotate text to the specification of the user rather it speaks plaintext.

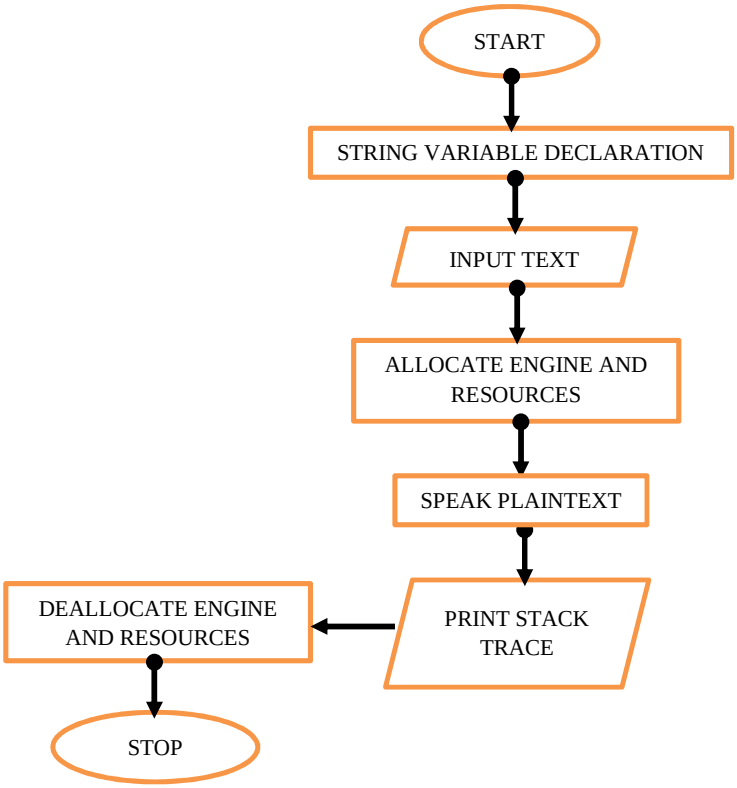

Figure 4: Algorithm of already existing systems

Due studies revealed the following inadequacies with already existing systems:

1. ***Structure analysis***: punctuation and formatting do not indicate where paragraphs and other structures start and end. For example, the final period in "P.D.P." might be misinterpreted as the end of a sentence.
2. ***Text pre-processing***: The system only produces the text that is fed into it without any pre-processing operation occurring.
3. ***Text-to-phoneme conversion***: existing synthesizer system can pronounce tens of thousands or even hundreds of thousands of words correctly if the word(s) is/are not found in the data dictionary.

7.2 Expectation of the New System

It is expected that the new system will reduce and improve on the problems encountered in the old system. The system is expected to among other things do the following;

1. The new system has a reasoning process. 2. The new system can do text structuring and annotation.
3. The new system's speech rate can be adjusted. 4. The Pitch of the voice can be adjusted.
5. You can select between different voices and can even combine or juxtapose them if you want to create a dialogue between them
6. It has a user friendly interface so that people with less computer knowledge can easily use it
7. It must be compatible with all the vocal engines 8. It complies with SSML specification.

8. Choice of Methodology for the New System

Two methodologies were chosen for the new system: The first methodology is Object Oriented Analysis and Development Methodology (OOADM). OOADM was selected because the system has to be represented to the user in a manner that is user-friendly and understandable by the user. Also since the project is to emulate human behaviour, Expert system had to be used for mapping of Knowledge into a Knowledge base with a reasoning procedure. Expert system was used in the internal operations of the program, following the algorithm of Rule-Based computation. The technique is derived from general principles described by researchers in knowledge engineering techniques (Murray et al., 1991; 1996).
The system is based on processes modelled in cognitive phonetics (Hallahan, 1996; Fagyal, 2001) which accesses several knowledge bases (e.g. Linguistic and phonetic knowledge bases, Knowledge bases about non-linguistic features, a predictive model of perceptual processes, and knowledge base about the environment).

9. Speech Synthesis Module

The TTS system converts an arbitrary ASCII text to speech. The first step involves extracting the phonetic components of the message, and we obtain a string of symbols representing sound-units (phonemes or allophones), boundaries between words, phrases and sentences along with a set of prosody markers (indicating the speed, the intonation etc.). The second step consists of finding the match between the sequence of symbols and appropriate items stored in the phonetic inventory and binding them together to form the acoustic signal for the voice output device.

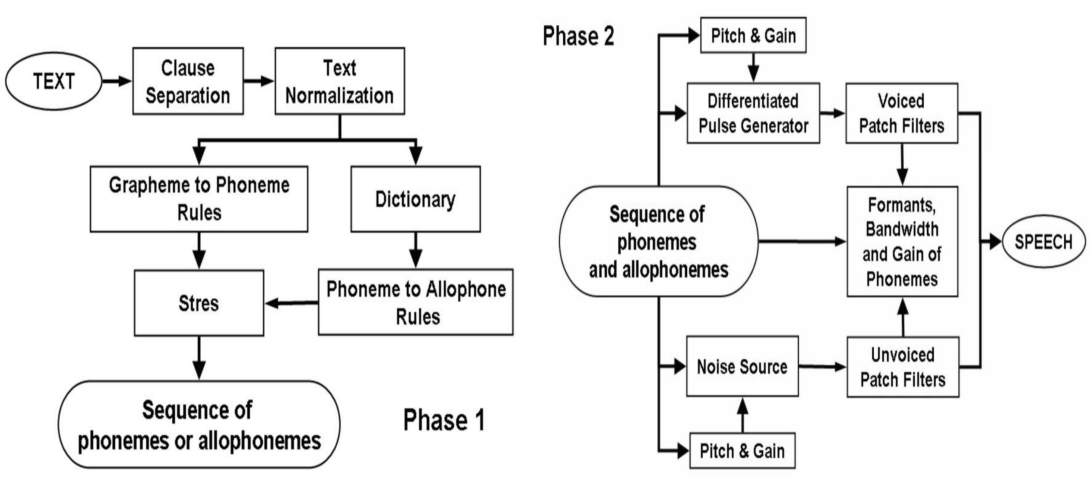

Figure 5: Phases of TTS synthesis process

To compute the output, the system consults
1. A database containing the parameter values for the sounds within the word,
2. A knowledge base enumerating the options for synthesizing the sounds.

Incorporating Expert system in the internal programs will enable the new TTS system exhibit these features:
1. The system performs at a level generally recognized as equivalent to that of a human expert
2. The system is highly domain specific.
3. The system can explain its reasoning process
4. If the information with which it is working is probabilistic or fuzzy, the system can correctly propagate uncertainties and provide a range of alternative solution with associated likelihood.

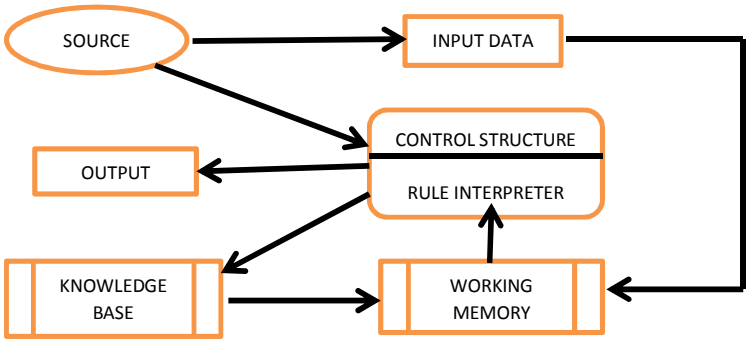

Figure 6: Data flow diagram of the Speech synthesis system Using Gane and Sarson Symbol

User Interface (Source): This can be Graphical User Interface (GUI), or the Command Line Interface (CLI).
Knowledge Base (Rule set): FreeTTS module/system/engine. This source of the knowledge includes domain specific facts and heuristics useful for solving problems in the domain. FreeTTS is an open source speech synthesis system written entirely in the Java programming language. It is based upon Flite. FreeTTS is an implementation of Sun's Java Speech API. FreeTTS supports end-of-speech markers.
Control Structures: This rule interpreter inference engine applies to the knowledge base information for solving the problem.
Short term memory: The working memory registers the current problem status and history of solution to date.

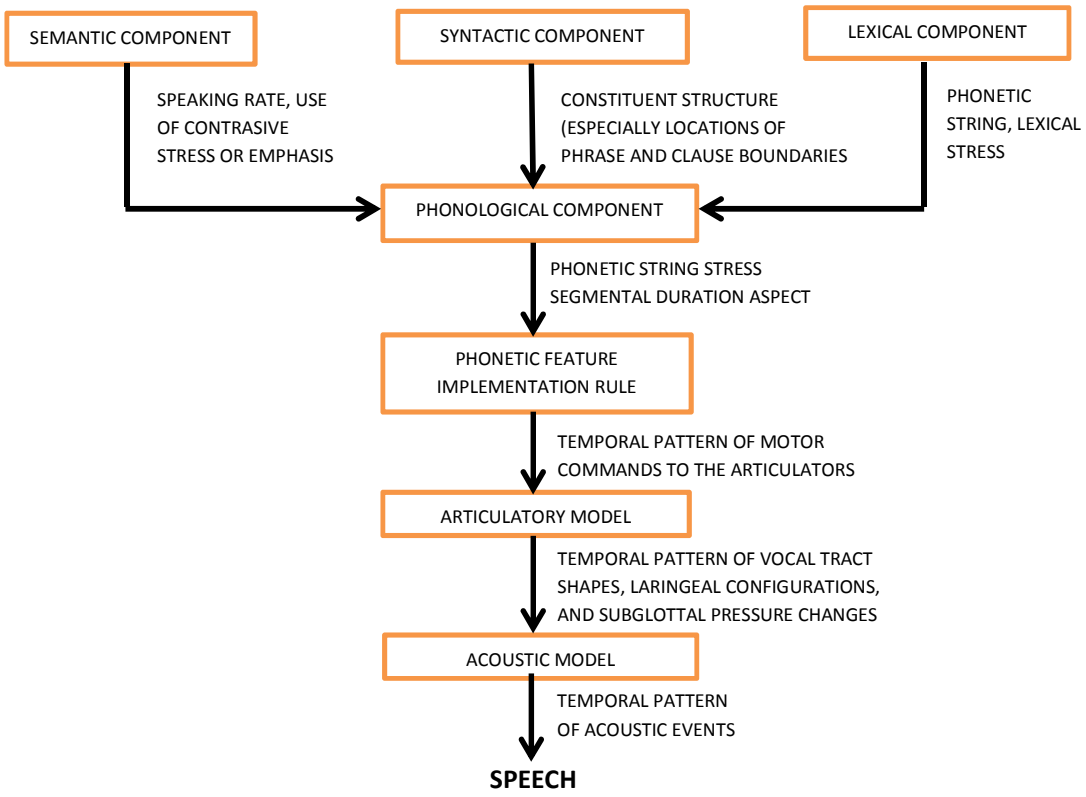

Figure 7: High Level Model of the Proposed System

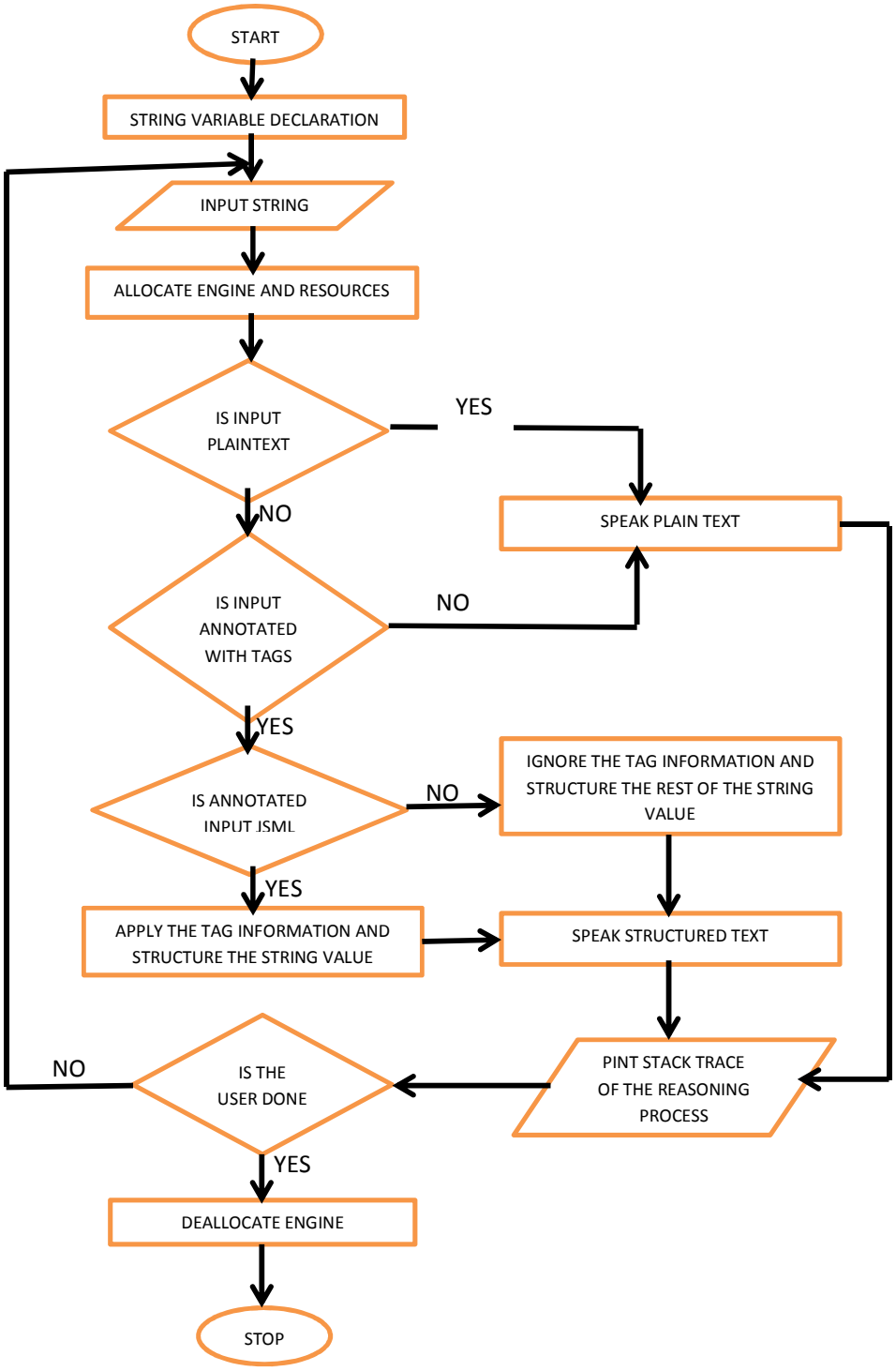

Figure 8: Flowchart representation of the program

8.1 Choice of Speech Engine and Programming Language

The speech engine used in this new system was the FreeTTS speech engine. FreeTTS was used because it is programmed using JAVA (the backbone programming language of this designed TTS system). It also supports SAPI (Speech Application Programming Interface) which is in synchronism with the JSAPI (Java Speech Application Programming Interface). JSAPI was also the standardized interface used in the new system.
FreeTTS includes an engine for the vocal synthesis that supports a certain number of voices (male and female) at different frequencies. It is recommended to use JSAPI to interface with FreeTTS because JSAPI interface provides the best methods of controlling and using FreeTTS. FreeTTS engine enable full control about the speech signal. This new designed system provides the possibility to choose a voice between three types of voices: an 8 kHz, diphone male English voice named *kevin*, a 16 kHz diphone male English voice named *kevin16* and a16khz limited domain, male US English voice named *alan*. The user could also set the properties of a chosen voice: the speaking rate, the volume and the pitch.

A determining factor in the choice of programming language is the special connotation (JSML) given to the program. This is a java specification mark-up language used to annotate spoken output to the preferred construct of the user. In addition to this, there is the need for a language that supports third party development of program libraries for use in a particular situation that is not amongst the specification of the original platform.
Considering these factors, the best choice of programming language was **JAVA.** Other factors that made JAVA suitable were its dual nature (i.e. implementing 2 methodologies with one language), its ability to Implements proper data hiding technique (Encapsulation), its supports for inner abstract class or object development, and its ability to provide the capability of polymorphism; which is a key property of the program in question.

9. Design of the New System

Some of the technologies involved in the design of this system includes the following:
Speech Application Programming Interface (SAPI): SAPI is an interface between applications and speech technology engines, both text-to-speech and speech recognition (Amundsen 1996). The interface allows multiple applications to share the available speech resources on a computer without having to program the speech engine itself. SAPI consists of three interfaces; The *voice text* interface which provides methods to start, pause, resume, fast forward, rewind, and stop the TTS engine during speech. The *attribute interface* allows access to control the basic behaviour of the TTS engine. Finally, the *dialog interface* can be used to set and retrieve information regarding the TTS engine.
Java Speech API (JSAPI): 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 supported through the Java Speech API are speech recognition and speech synthesis.
Speech recognition provides computers with the ability to listen to spoken language and to determine what has been said. Speech synthesis provides the reverse process of producing synthetic speech from text generated by an application, an applet or a user. It is often referred to as text-to-speech technology.

9.1 Design of Individual Objects of the Program

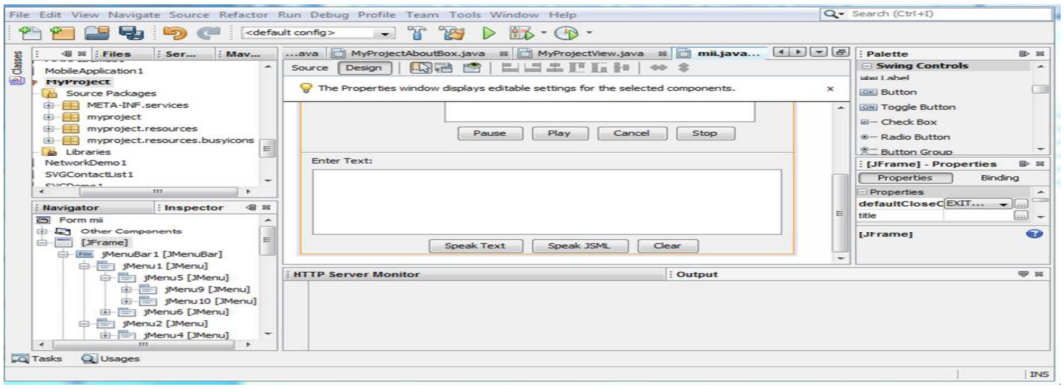

Figure 9: Netbeans Interface and program object manipulation

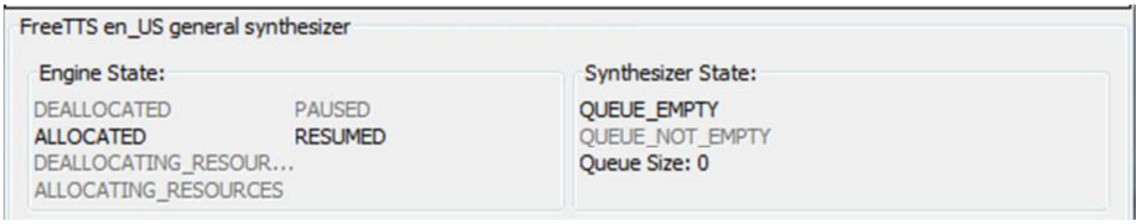

Figure 10: Panel containing the monitoring process of the system

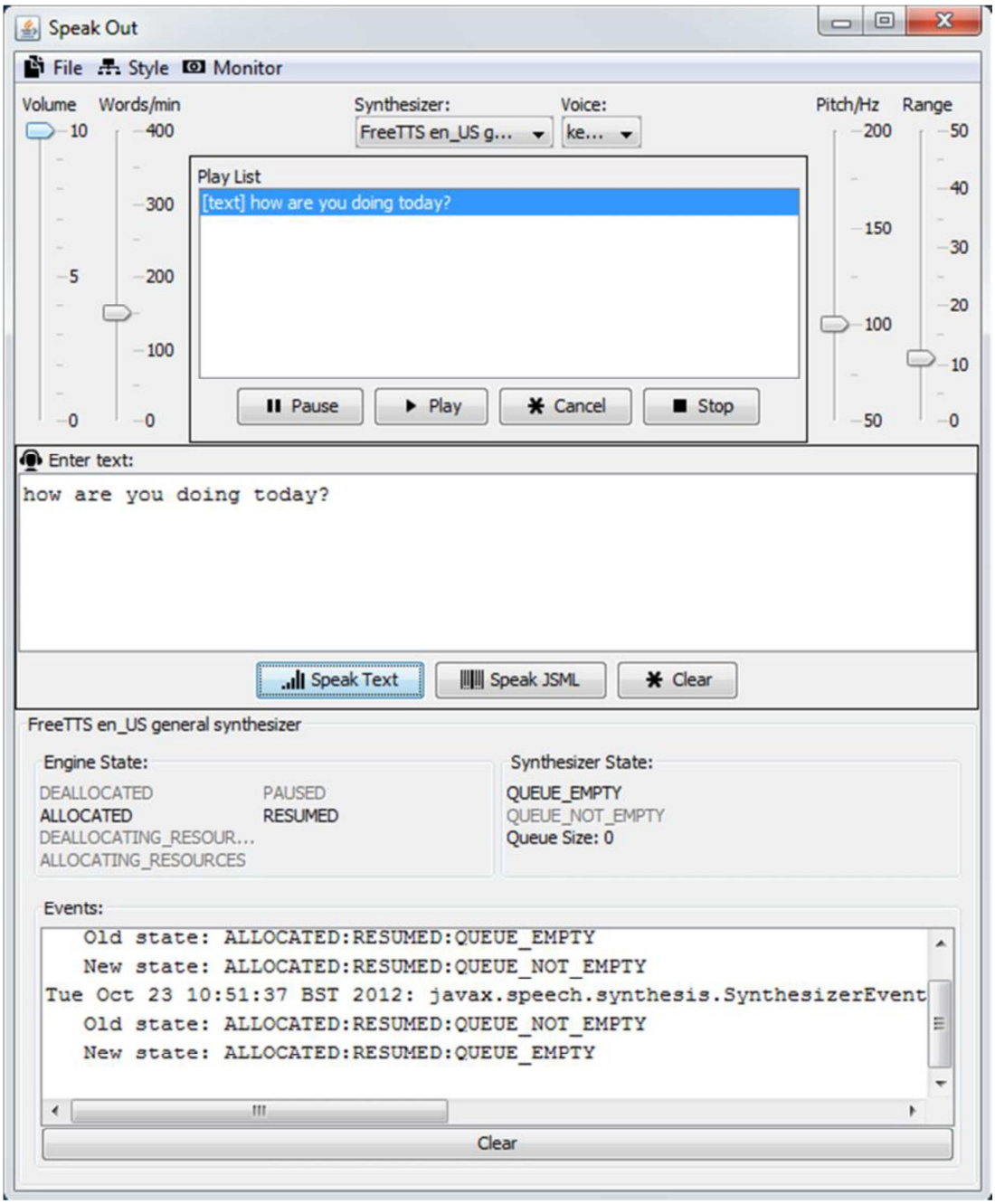

Figure 11: Overall view of the new system.

9.2 Functions of the Abstract Classes

1. Menu Bar: This will have the function of selecting through many variables and File chooser system.
2. Monitor: Monitors the reasoning process by specifying the allocation process and de-allocation state
3. Voice System: This shows the different voice option provided by the system
4. Playable session: This maintains the timing of the speech being given out as output, and produces a speech in synchronism with the rate specified.
5. Playable type: This specifies the type of text to be spoken, whether it is a text file or an annotated JSML file
6. Text-to-Speech activator: This plays the given text and produces an output
7. Player Model: This is a model of all the functioning parts and knowledge base representation in the program
8. Player Panel: This shows the panel and content pane of the basic objects in the program, and specifies where each object is placed in the system
9. Synthesizer Loader: This loads the Synthesizer engine, allocating and de-allocating resources appropriately

10. Conclusion and Recommendation

Synthesizing text is a high technology advancement and artificial formation of speech given a text to be spoken. With Text-to-Speech synthesis, we can actually mediate and fill in the lacuna provided by not fully exploiting the capabilities of some handicapped individuals. It's never been so easy to use a text-to-speech program, as just one click and your computer will speak any text aloud in a clear, natural sounding voice.
Therefore, there is need to use Information Technology to solve the problem for the
Before the use of the new system, proper training should be given to the users. This training can come in handy with proper tutor on how to handle JSML language and how to use it to annotate text for the proper output and emphasis.

References

[1] Abedjieva et al. (1993): Acoustics and audio signal processing. http://www.ufh.netd.ac.za/bitstream/10353/495/1/Mhlanathesis.pdf date: 21/07/12

[2] Allen, J., Hunnicutt, M.S., and Klatt, D. (1987). From text to speech – the MITalk system. MIT press, Cambridge, Massachusetts.

[3] Hallahan (1996): Phonetics and Theory of Speech Production. http://www.indiana.edu/~acoustic/s702/readfull.html date: 22/07/12

[4] I.R. Murray, J.L. Arnott, N. Alm and A.F. Newell (1991). A communication system for the disabled with emotional synthetic speech produced by rule. Proc. European Conference on Speech Communication and Technology 91.

[5] Murray et al. (1996): Application of an analysis of acted vocal emotions. http://www.dl.acm.org/citation.cfm?id=1314912 date: 22/07/12

[6] Fagyal, Z., Douglas, K. and Fred J. (2006) A Linguistic Introduction, Cambridge University Press, the Edinburgh Building, Cambridge CB2 2RU, UK

[7] Amundsen (1996): Review of Speech Synthesis Technology. http://www.koti.welho.com/slemmett/dippa/dref.html date: 23/08/12

[8] Wikipedia, the free encyclopedia: http://en.wikipedia.org/wiki.