An Interactive Intelligent Web-Based Text-to-speech System for the Visually Impaired

Dr. N. P. Essien
Computer Education, University of Uyo, Nigeria
Mrs V. A. Uwah
Department of Computer Science, Akwa Ibom State College of Education, Afaha Nsit
&
Mr. E. P. Ododo
Computer Education, University of Uyo, Nigeria

Abstract

Text-to-speech is an AI-based system that converts text into speech. It uses Natural Language Processing and Digital Signal Processing to achieve its goal. The paper describes a web-based system that converts text to speech. It was developed using JavaScript and is compatible with a variety of speech synthesis libraries. The programming language used in this project is light-weight and can support development of web apps. It is also able to convert plain text files into an audio format. The system is able to convert plain text files to audio. This software can be used by users all around the world at the same time and its supported on major desktop and hand-held devices once it is linked to the web. This system will be of great help to lecturers and students in the classroom, e-libraries and computer aided learning for visually impaired users.

Keywords: Text to Speech system, Natural Language Processing, Digital Signal Processing, Web based system

Introduction

The society is more and more progressively focused on data handling, processing, storage and dissemination, using electronics based technologies, today's computers is able stimulate several human capabilities like reading, grasping, calculating, speaking, memory recall, comparison numbers, drawing, passing judgments, and even interactive learning. Researchers are working to expand these capabilities and, therefore the power of computers by developing hardware and software that can initiate intelligent human behavior (Raiyetunbi and Ayeh, 2020). There are researchers working on the systems that have the ability to reason, to learn or accumulate knowledge to strive for self-improvement, and to stimulate human sensory and mechanical capabilities. This general area of research is known as Artificial Intelligence.

Artificial intelligence (AI) is a wide-ranging branch of computer science concerned with building smart machines capable of performing tasks that typically require human intelligence. Artificial intelligence (AI) is a major influence in our society today and greatly on the state of education today, and the implications are huge. AI has the potential to transform education and learning in particular by empowering learners with different abilities. It is a technology that enable the physically challenged person and semi-illiterates assess and read through electronic documents, thus bridging the digital divide with the aid of an interactive intelligent system.

An interactive intelligent system is an intelligent system that people interact with. An intelligent system embodies one or more capabilities that have traditionally been associated more strongly with humans than with computers, such as the abilities to perceive, interpret, learn, use language, reason, plan, and decide. Systems that exhibit these capabilities mostly use techniques that originated within the field of artificial intelligence. One of such intelligent system is text-to-speech synthesizer.

Speech synthesis is the artificial production of human speech. A computer system used for this purpose is called a speech computer or speech synthesizer, and can be implemented in software or hardware products. A text-to-speech (TTS) system (Artificial Interpreter, AI) converts normal language text into speech; other systems render symbolic linguistic representations like phonetic transcriptions into speech, (Allen, 1987 in Raiyetunbi and Ayeh, 2020).

Text-to-speech synthesis -TTS - is the automatic conversion of a text into speech that resembles, as closely as possible, a native speaker of the language reading that text. Text-to-speech synthesizer (TTS) is the technology which lets computer speak to you. The TTS system gets the text as the input and then a computer algorithm which called TTS engine analyses the text, pre-processes the text and synthesizes the speech with some mathematical models (algorithms). The

TTS engine usually generates sound data in an audio format as the output (Isewon, Oyelade and Oladipupo, 2012). TTS systems are capable of "reading" any string of text characters to form original sentences.

In special education, that is education for the physically disabled, reading is essential in learning. The act of reading permits students to learn vocabulary and concepts and to access different types of reading materials (Serafini 2004). This is why the able body persons are required to read textbooks, lecture materials and other text to the hearing of the visually impaired person so as to enable the physically challenge to learn. If students fall behind in reading comprehension for their age/grade level, then students struggle to process new vocabulary and concepts presented in textbooks and other literature. Difficulty in reading may translate into poor school performance due to the inability to process new vocabulary and concepts in a meaningful manner. These difficulties can evolve into students especially the physically challenged in losing interest in education. This cycle can lead to students‟ dropping out of high school and possessing below-average reading comprehension skills as adults. Hence the manual or traditional reading approach that are usually employed in schools, lecture rooms , Libraries and workshop, could be faced with certain challenges to the physically challenged. It is through reading that students expand their vocabulary and learn about the world.

Researchers have demonstrated that the use of technology exposes struggling readers to different types of literature and assists them with vocabulary acquisition (Stone-Harris 2008). The significance of the study also exhibits that the use of web based text to speech synthesizer apps can lead to an improvement in struggling readers‟ skills and attitudes. If use of such apps can be proven to benefit struggling readers, then educators will possess another instructional technique to assist struggling readers improve their reading skills and attitudes.

Some of them which inspired the development of our software are as follows: Isewon, Oyelade and Oladipupo (2012) developed the Text To Speech Robot, a

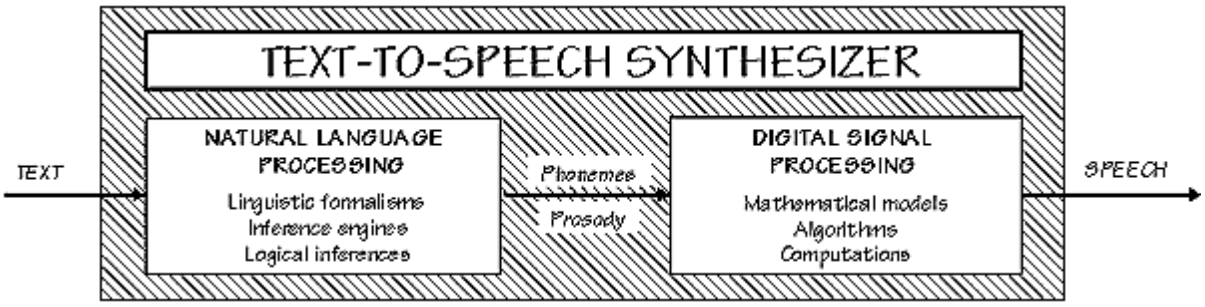
simple application with the text to speech functionality. The system was developed using Java programming language. As a result, the software could only function offline when installed on a computer because it was not cloud enhanced. Kaladharan (2015) worked on An English Text to Speech Conversion System; a system which he developed using Microsoft .Net framework. Although the system was successful, it was not cloud enhanced or enabled. (Serafini 2004) has explained that much research validates the importance of reading aloud to students, positing that the act of reading aloud introduces new vocabulary and concepts, provides a fluent model, and allows students access to literature they are unable to read independently. He adds that audiobooks, sometimes called Text to speech system (TTS) or Artificial Interpreter is an important component of a comprehensive reading program. (Kylene, 1998) has said that audiobooks, when used with reluctant, struggling, or second language learners, serve as a scaffold that allows students to read beyond their reading level. Since the reading process develops through oral language experiences, audiobooks benefit struggling readers by increasing comprehension and appreciation of written text (Wolfson 2008). This benefit has long been seen by classroom teachers. 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: To enable the deaf and dumb to communicate and contribute to the growth of an organization through synthesized voice and to enable the blind and elderly people enjoy a User-friendly computer interface.

Overview of Speech Synthesis

Speech synthesis can be described as artificial production of human speech (Suendermann, Höge, and Black, 2010). The computer system used for this is called a speech synthesizer and can be implemented in software or hardware. The text-to-speech system (TTS) converts natural language into text-to-speech (Allen, Hunnicutt

& Klatt, 1987 in Isewon, Oyelade & Oladipupo, 2012). Text-to-speech intelligence allows the visually impaired or visually impaired to listen to what is written on a personal computer. A text to speech system is by definition a system that produces synthetic speech.

This is implicitly clear, which means some kind of contribution. What is not clear is this type of entry. If the input is text only, then only the text means that it has not added any additional phonetic and / or phonological information that the system can call a speech synthesis (TTS) system. The text-to-speech system has a front and back section. In the front, two question tasks were performed. With the first application, you can change the text of text from symbols, such as numbers and abbreviations, to equivalent words. This process is called text normalization, pre-processing, or tagging. The front part then presents the phonetic transcriptions of each word and divides and marks the text into prosodic entities such as sentences, clauses and questions. This assignment process is called text-to-phoneme or grapheme-to-phoneme transformation. Phonetic transcriptions and prosodic information together form symbolic linguistic representations created at the front end. The background - often called a synthesizer - then turns the symbolic linguistic representation into sound. In some systems, this involves the use of prosody calculation (sound contour, phoneme duration) applied to the speech output (Van-Santen, 1994 in Isewon, Oyelade and Oladipupo, 2012). A simplified formulation of this tolerance is shown in Figure 1 below. The input text can be, for example, a word processing program data, a standard ASCII e-mail message, a mobile text message, or a newspaper scanned text. The speech sound is finally generated by a lower level information synthesizer.



Structure of A Text-To-Speech Synthesizer System

Text-to-speech synthesis takes place in several steps. The TTS systems get a text as input, which it first must analyze and then transform into a phonetic description. Then in a further step it generates the prosody. From the information now available, it can produce a speech signal. The structure of the text-to-speech synthesizer can be broken down into major modules:

Natural Language Processing (NLP) module: It produces a phonetic transcription of the text read, together with prosody.

Digital Signal Processing (DSP) module: It transforms the symbolic information it receives from NLP into audible and intelligible speech.

The major operations of the NLP module are as follows:

Text Analysis: First the text is segmented into tokens. The token-to-word conversion creates the orthographic form of the token. For the token "Mr" the orthographic form "Mister" is formed by expansion, the token "12" gets the orthographic form "twelve" and "1997" is transformed to "nineteen ninety seven".

Application of Pronunciation Rules: After the text analysis has been completed, pronunciation rules can be applied. Letters cannot be transformed 1:1 into phonemes because correspondence is not always parallel. In certain environments, a single letter can correspond to either no phoneme (for example, "h" in "caught") or several phoneme ("m" in "Maximum"). In addition, several letters can correspond to a single phoneme ("ch" in "rich"). There are two strategies to determine pronunciation:

In dictionary-based solution with morphological components, as many morphemes (words) as possible are stored in a dictionary. Full forms are generated by means of inflection, derivation and composition rules. Alternatively, a full form dictionary is used in which all possible word forms are stored. Pronunciation rules determine the pronunciation of words not found in the dictionary.

In a rule based solution, pronunciation rules are generated from the phonological knowledge of dictionaries. Only words whose pronunciation is a complete exception are included in the dictionary.

The two applications differ significantly in the size of their dictionaries. The dictionary-based solution is many times larger than the rules-based solution's dictionary of exception. However, dictionary-based solutions can be more exact than rule-based solution if they have a large enough phonetic dictionary available.

Prosody Generation: after the pronunciation has been determined, the prosody is generated. The degree of naturalness of a TTS system is dependent on prosodic factors like intonation modelling (phrasing and accentuation), amplitude modelling and duration modelling (including the duration of sound and the duration of pauses, which determines the length of the syllable and the tempos of the speech) (Text-to-Speech Technology, 2014).

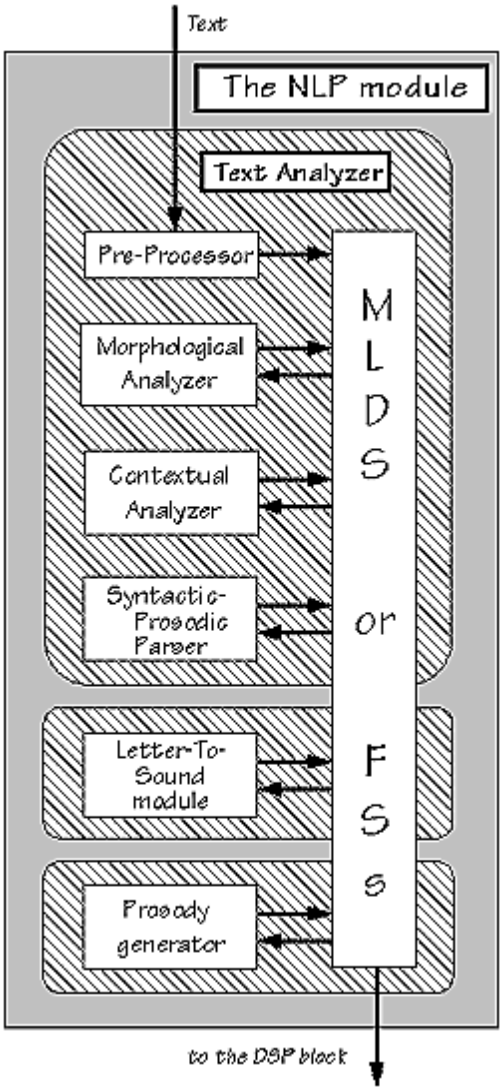


Fig 1: Operations of the natural Language processing module of a TTS synthesizer (Isewon, Oyelade and Oladipupo, 2012).

The output of the NLP module is passed to the DSP module. This is where the actual synthesis of the speech signal happens. In concatenative synthesis the selection and linking of speech segments take place. For individual sounds the best option (where several appropriate options are available) are selected from a database and concatenated.

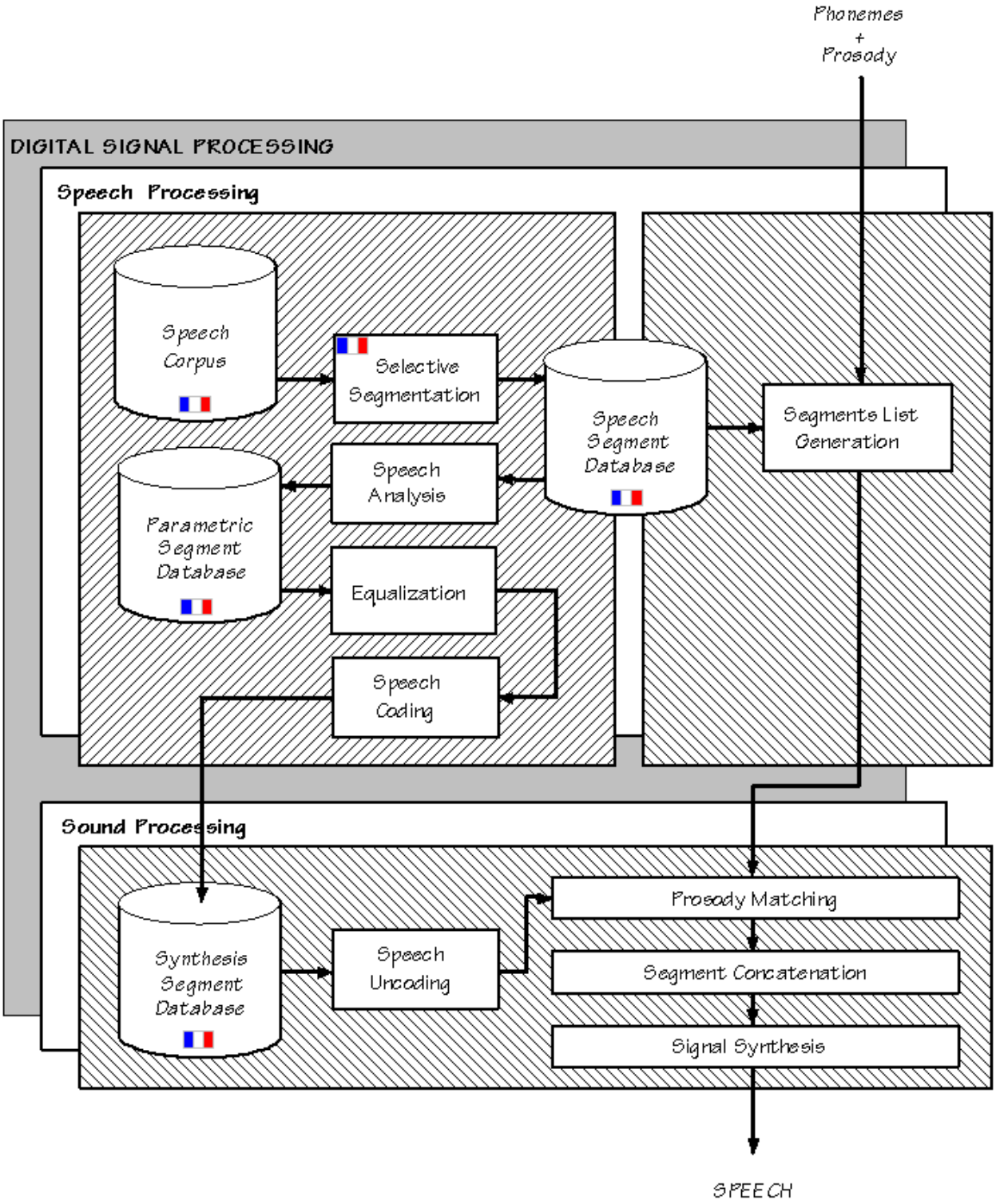


Fig 2: The DSP component of a general concatenation- based synthesizer (Isewon, Oyelade and Oladipupo, 2012).

DESIGN & IMPLEMENTATION

The methodology used for this study is called SSADM. It is a structured system analysis and design methodology. System Analysis and Design Method or SSADM is a methodology utilized for studying and designing systems. The SSADM is a commonly used method for carrying out system analysis and design phases of any kind of information technology project. The SSADM is the UK's standard method for developing information technology projects. Waterfall diagram of the SSADM is presented in the figure below.

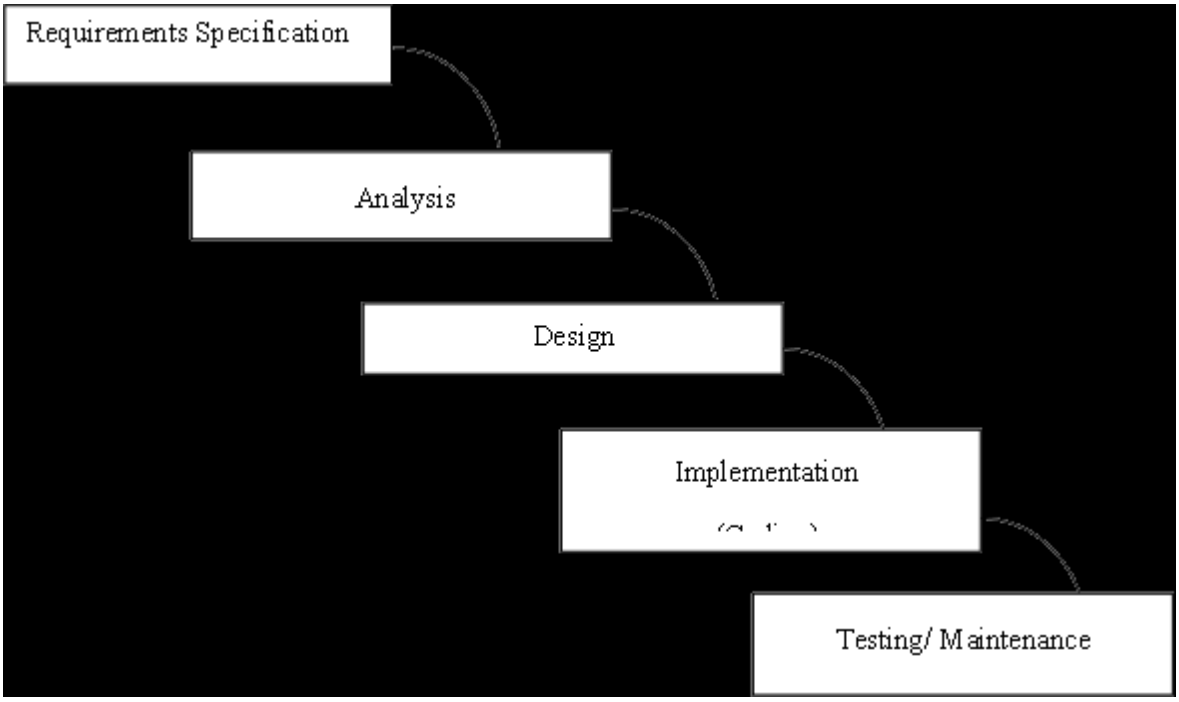


Fig 3: Waterfall Diagram of SSADM (Raiyetunbi and Ayeh, 2020)

There are various tools that are used to construct a model for the analysis and design stage. One of these tools is the UML. Many notations for describing object-oriented designs were proposed during the 1980s and 1990's. The UML is a language that is used to describe and modify the artifacts of object-oriented software. UML is a general-purpose modeling language. It was created by the Object Management Group. UML involves a set of graphic techniques used to create representations of

complex software intensive systems. These include designing business processes and databases, describing activities and actors, and programming language statements. Sequence diagram, Use case and Flow chart diagrams are the UML models that were used in designing the text to speech software.

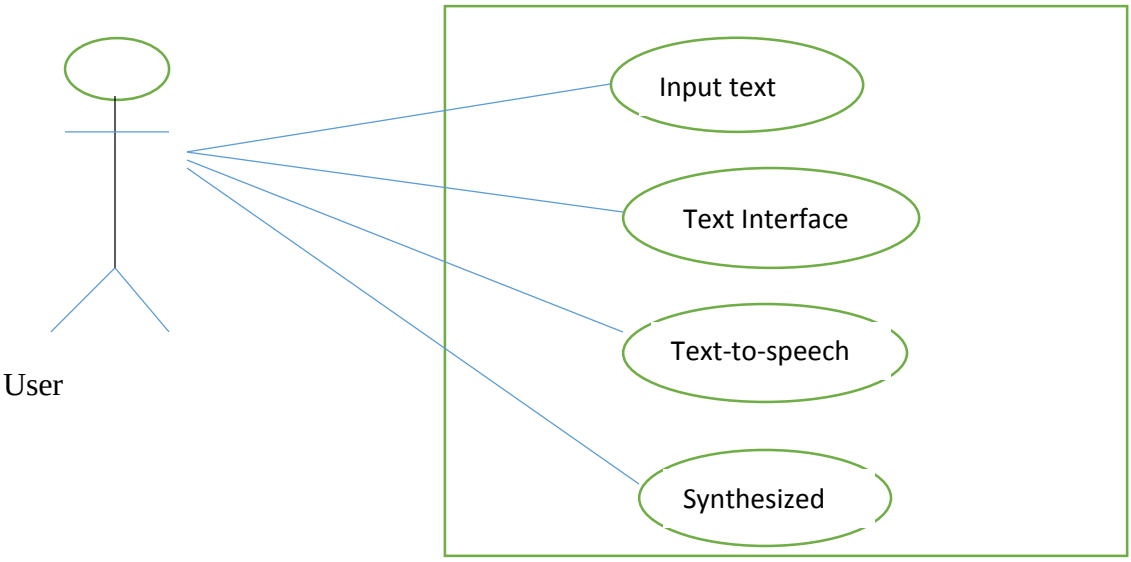


Fig 4. Use case diagram for the voice synthesizer

Behavior diagrams help describe the behavior of a system. They are commonly used to describe the functions of software systems. The use case diagram presented in this paper is an example of a system's model in terms of actors and their goals. This model describes the system's functionality in terms of actors and their goals. Use case diagrams depict the interactions between a system and its users. They help determine the user's expectations and how the system will interact with them. Figure above shows the use case diagram in this work. . From the case diagram the user is required to type in / input his/her preferable text in the right text format. Before converting text to synthesized speech, the user must first type in a valid text document. The post conditions are either a failure end or a success end. At the failure end, the system prompts the user to re-input a new text. At the success end, the user Type the text and click read, the system produces an audible speech.

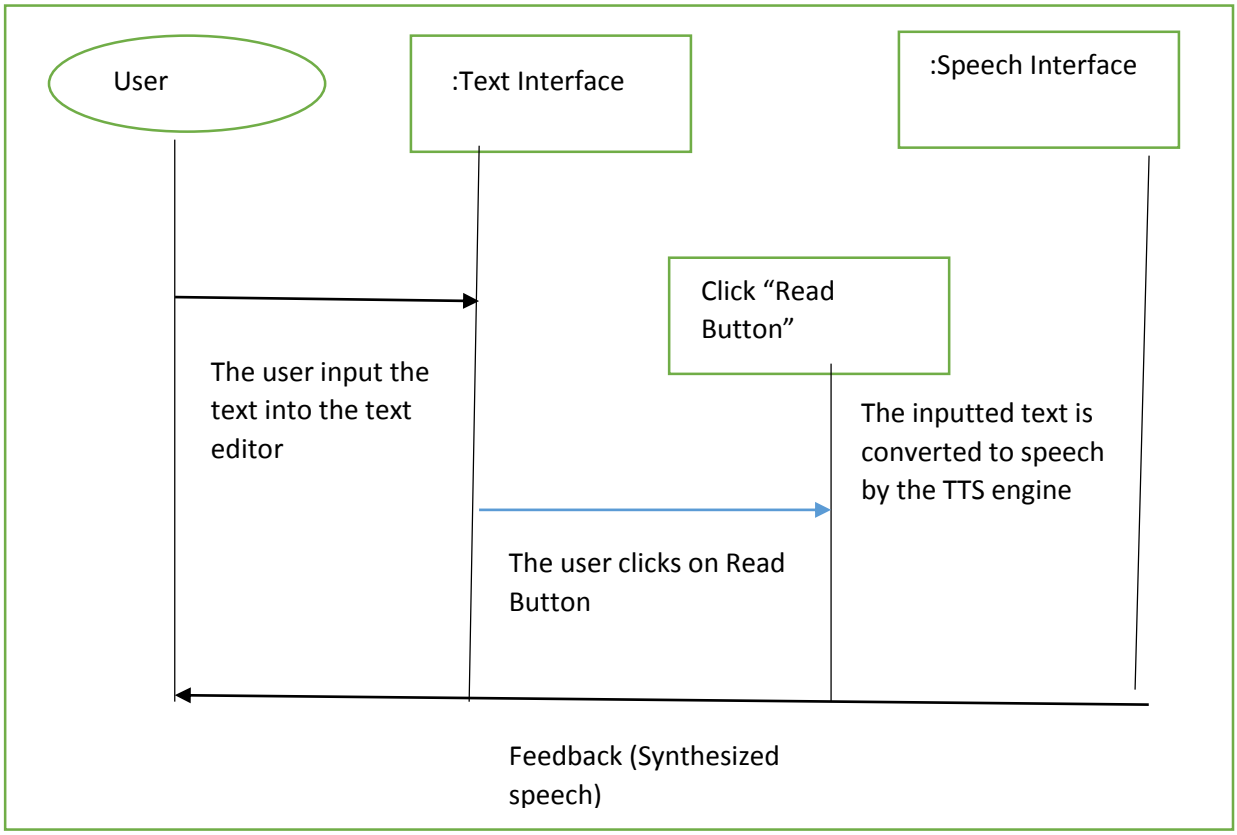


Fig 5: Sequence diagram for the TTS voice synthesizer

A sequence diagram shows the flow of control and data in a system. A sequence diagram shows how objects communicate with one another in terms of a sequence or a message. It also shows the lifespan of the objects relative to the messages. It shows how objects communicate with each other through messages in the execution of use cases or operations. They illustrate how messages are sent and received between objects and in what sequence.

System Requirement and Functions

This system is a text-to-speech application that can easily translate text to audio, a simple Web-based Text-To-Speech application with the text to speech functionality. The system was built using JavaScript, which is a programming language that is commonly used for web applications with HTML5 and CSS for the interface design. JavaScript is a robust programming language that can easily build

robust text to speech systems. It is also light-weight in terms of programming complexity.

JavaScript Text to Speech using SpeechSynthesis Interface

Javascript is a programming language that enables developers to create text to speech apps. To build a text to speech app, they need to implement the Web Speech API. The following are speechsynthesis interface used in the development:

JavaScript SpeechSynthesis Interface

The speech synthesis service's main controller interface is used to control the creation and operation of speech. This interface is used to start the speech, stop the speech, pause it and resume, along with getting the voices supported by the device. The Following are the methods available in this Interface:

speak(): Add the utterance(object of SpeechSynthesisUtterance) in the queue, which will be spoken when there is no pending utterance before it. This is the function, we will be using too.

pause(): To pause the current ongoing speech.

resume(): To resume the paused speech.

cancel(): To cancel all the pending utterances or speech created, which are not yet played.

getVoices(): To get list of all supported voices which the device supports.

```
if ('speechSynthesis' in window) {


  var synthesis = window.speechSynthesis;
```

```
 // Regex to match all English language tags e.g en, en-US, en-GB
 var langRegex = /^en(-[a-z]{2})?$/i;


 // Get the available voices and filter the list to only have English speakers
 var voices = synthesis.getVoices().filter(voice => langRegex.test(voice.lang));
 // Log the properties of the voices in the list
 voices.forEach(function(voice) {
   console.log({
     name: voice.name,
     lang: voice.lang,
     uri: voice.voiceURI,
     local: voice.localService,
     default: voice.default
   })
 });
} else {
 console.log('Text-to-speech not supported.');
}
```

In the above code, you get the list of available voices on the device, and filter the list using the langRegex regular expression to ensure that we get voices for only English speakers. Finally, you loop through the voices in the list and log the properties of each to the console.

JavaScript SpeechSynthesisUtterance Interface

This is the interface in which the researchers actually create the speech or utterance using the text provided, setting a language type, volume, pitch of the voice, rate of speech, etc. Once we have created an object for this interface, we provide it to the SpeechSynthesis object's speak() method to play the speech. The Following are the properties provided by this interface to configure it:

lang: To get and set the language of speech.

pitch: To get and set the pitch of the voice at which the utterance will be spoken.

rate: To get and set the speed at which the utterance will be spoken.

volume: To get and set the volume.

text: To get and set the text which has to be spoken.

voice: To get or set the voice to be used.

```
if ('speechSynthesis' in window) {
  var synthesis = window.speechSynthesis;
  // Get the first `en` language voice in the list
  var voice = synthesis.getVoices().filter(function(voice) {
    return voice.lang === 'en';
  })[0];
  // Create an utterance object
  var utterance = new SpeechSynthesisUtterance('Hello World');
  // Set utterance properties
  utterance.voice = voice;
```

```
  utterance.pitch = 1.5;
  utterance.rate = 1.25;
  utterance.volume = 0.8;

  // Speak the utterance
  synthesis.speak(utterance);

} else {
  console.log('Text-to-speech not supported.');
}
```

In the code above, you get the first en language voice from the list of available voices. Next, you create a new utterance using the SpeechSynthesisUtterance constructor. You then set some of the properties on the utterance object like voice, pitch, rate, and volume. Finally, it speaks the utterance using the speak() method of SpeechSynthesis.

Speaking an Utterance

In the code below instances are simply pass in the SpeechSynthesisUtterance instance as argument to the speak() method to speak the utterance.

```
var synthesis = window.speechSynthesis;
var utterance1 = new SpeechSynthesisUtterance("Hello Friend");
var utterance2 = new SpeechSynthesisUtterance("My name is Emmanuel Philip.");
var utterance3 = new SpeechSynthesisUtterance("I'm a web developer from Nigeria.");
```

```
synthesis.speak(utterance1);

synthesis.speak(utterance2);

synthesis.speak(utterance3);
```

There are a couple of other things that can be done with the SpeechSynthesis instance such as pause, resume and cancel utterances. Hence the pause(), resume() and cancel() methods are available as well on the SpeechSynthesis instance.

JavaScript window.speechSynthesis Property

This property of the Javascript window object is used to get the reference of the speech synthesis controller interface, on which we call the speak() method.

Getting a reference to a SpeechSynthesis object can be done with a single line of code:

```
var synthesis = window.speechSynthesis;
```

It is very useful to check if SpeechSynthesis is supported by the browser before using the functionality it provides. The following lines of code shows how to check for browser support:

```
if ('speechSynthesis' in window) {
  var synthesis = window.speechSynthesis;
} else {
  console.log('Text-to-speech not supported.');
```

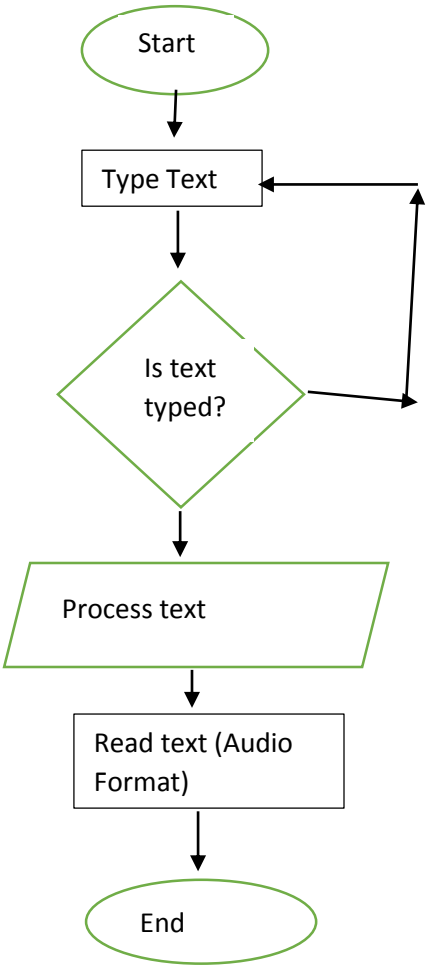


Fig 6: System overview

Software Requirements

The following are the softwares requirement specification for our Ultimate Text Reader based on the Text to Speech JS Library:

Google Chrome Chrome 4 to 32 does not support Speech Synthesis API property. Chrome 33 to 67 supports Speech Synthesis API property.

Mozilla Firefox Speech Synthesis API is not supported by Mozilla Firefox browser version 2 to 30. Speech Synthesis API is not supported by Mozilla Firefox browser version 31 to 48 by default but can be enabled in Firefox. Speech Synthesis API is supported by Mozilla Firefox browser version 49 to 61.

Microsoft Edge Microsoft Edge browser version 12 to 13 does not support this property speech-synthesis-api. Microsoft Edge browser version 14 to 17 supports this property speech-synthesis-api.

Opera Opera version 10.1 to 26 doesn't support Speech Synthesis API. Opera version 27 to 53 supports Speech Synthesis API.

Interface Design

Below are the screenshots of the Web based text-to-Speech application

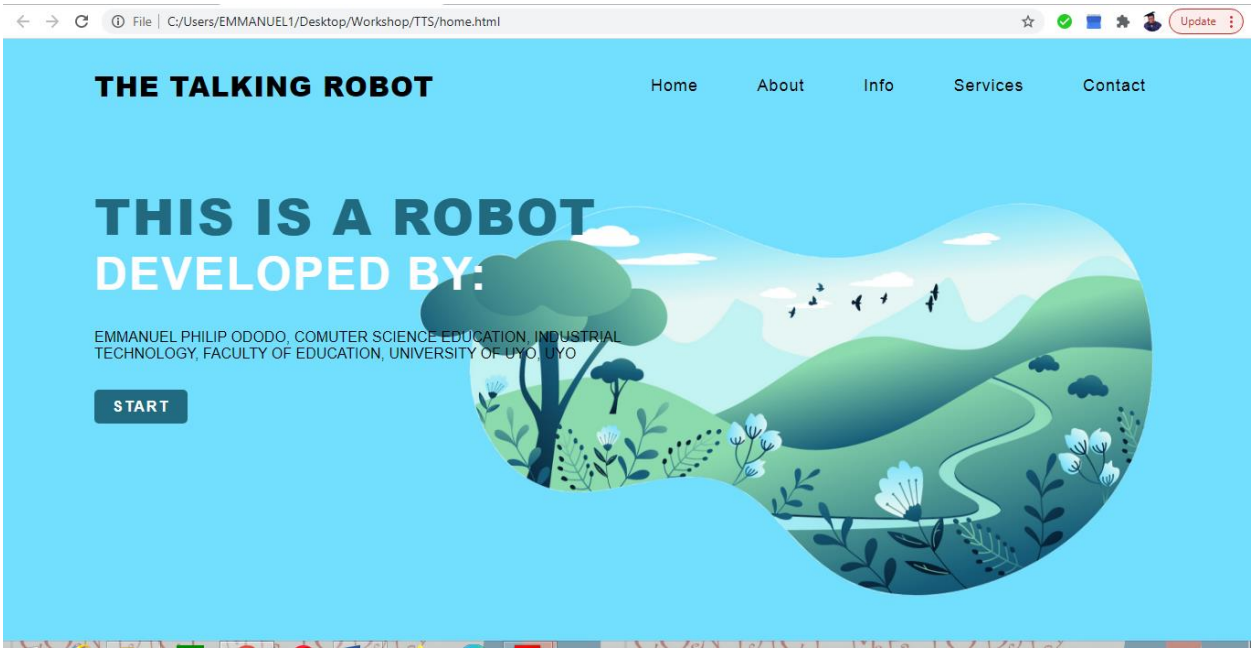


Fig 7: Landing page of the web-based text-to-speech application

Fig 8: web-based text-to-speech application interface where text is entered

Fig 9: Reading interface

Educational Implication for the Study

The study has implications for Government (Federal and State), higher institutions administrators and the ministry of education, higher institutions lecturers. The lecturers have important roles to play for updating their knowledge

through seminars, workshops and retraining on the current way of helping the hearing impaired to learn. Government should sponsor lecturers to attend seminars and workshops so as to increase their exposure on how to utilize intelligent system of this kind. The study have implications for government, ICT policy cannot be achieved without the government providing adequate technology facilities that can be used by lecturers in teaching and learning not only on administration purposes to higher institutions. However, both government and education administrators should study and understand the factors responsible for poor application of technology in teaching and learning in higher institutions with a view to make budgetary provision to incorporate such needs as the provision of infrastructural facilities, in-service training, curriculum review to accommodate technology application in teaching and learning in education setting. Ministry of education should through federal and state government procured ICT equipment/facilities to Special learners at subsidized rate to enable them apply it for individualized learning so as to motivate and give them confidence in applying them in teaching and learning in higher institutions.

Conclusion

Text to speech synthesis is a rapidly growing aspect of computer technology (Natural Language processing of the Artificial Intelligence) and is increasingly playing a more important role in the way we interact with the system and interfaces across a variety of platforms. Various operations and processes involved in text to speech synthesis have been identified. A very simple and attractive graphical user interface which allows the user to type in his/her text provided in the text field in the application have also been developed. The system interfaces with a text to speech engine developed for American English. This system can also be replicated in other areas or platforms where text to speech technology would be an added advantage and increase functionality such as telephony systems, ATM machines, video games etc.

Recommendations

The following recommendations were made:

1. Government should assist teachers and instructors with the necessary materials including funds to develop this intelligent system for easy teaching and learning of the hearing impaired in Nigeria.

2. Government should provide enabling environment for in-service training programmes for Special Education teachers and instructor on how to use intelligent system

3. Government should set up a regulate body that will be responsible for development and evaluation of intelligent text-to-speech and other related instructional package

4. Government also should ask the Special Education curriculum planners to integrate intelligent text-to-speech in implementing process

5. Finally, Government, NGOs, and philanthropist should assist Special Learners with the intelligent text-to-speech system and intelligent teaching room.

References

Isewon, I., Oyelade, O. J., & Oladipupo, O. O. (2012). Design and implementation of text to speech conversion for visually impaired people. *International Journal of Applied Information Systems, 7(2), 26-30.*

Kaladharan, N. (2015). An English Text to Speech Conversion System, International Journal of Advanced Research in Computer Science and Software Engineering, vol 5, Issue 10, October- 2015.

Kylene, B. (1998). "Listen While You Read: Struggling Readers and Audiobooks." School Library Journal 44 (4), 34–35.

Raiyetunbi, O. J. and Ayeh, E. (2020) An Interactive Cloud Based User Oriented, Dynamic and Intelligent Text-To-Speech Module. East African Scholars Journal of Engineering and Computer Sciences. Vol. 3(1) DOI: 10.36349/easjecs.2020.v03i01.001

Serafini, F. (2004). "Audiobooks and Literacy: An Educator's Guide to Utilizing Audiobooks in the Classroom." New York: Listening Library. http://www.frankserafini.com/classroom-resources/audiobooks.pdf (accessed March 21, 2020).

Stone-Harris, S. (2008). *The benefit of utilizing audiobooks with students who are struggling readers*. Walden University.

Suendermann, D., Höge, H., and Black, A., (2010). Challenges in Speech Synthesis. Chen, F., Jokinen, K., (eds.), *Speech Technology*, Springer Science + Business Media LLC.

Text-to-speech technology: In Linguatec Language Technology Website. Retrieved February 21, 2021, from http://www.linguatec.net/products/tts/information/t echno logy.

Wolfson, G. (2008). Using audiobooks to meet the needs of adolescent readers. *American Secondary Education*, 105-114.