African Journal of Science and Technology (AJST)
Science and Engineering Series Vol. 6, No. 1, pp. 80 - 89

SWAHILI TEXT-TO-SPEECH SYSTEM

K. Ngugi, W. Okelo-Odongo, P. W. Wagacha

School of Computing & Informatics, University of Nairobi, PO Box 30197, Nairobi, Kenya

ABSTRACT:- *Text-to-speech (TTS) applications have been applied in diverse areas all over the world. Considering the fact that Swahili pronunciation is not complicated, and the language spoken by about 45 – 100 million people as their first or second language,, we considered the feasibility, and developed a Swahili Text-to-Speech (TTS) system. This paper gives an account of the Swahili TTS system developed. It discusses the analysis, design, achievements, points out some fundamental issues encountered in its development, and suggestions for extensions to achieve a complete Swahili TTS system.*
.

INTRODUCTION

To many of us, the term *speech synthesis* evokes memories of mechanical, monotonous or repetitive voices but what really is a Text-to-Speech system? Simply defined, it is written text transformed into speech; reading or dictating machines; the part of speech technology, which is concerned with automatically generating speech from a computer [Univ. Birmingham.1999]. Typically, input is text and the desired output is an acoustic speech signal, hence: text-to-speech synthesis.

A Text-To-Speech (TTS) synthesizer is a computer-based system that should be able to read *any* text aloud, whether it was directly introduced in the computer by an operator or scanned and submitted to an Optical Character Recognition (OCR) system [Dutoit, 1999]. It is the process which allows the transformation of a string of phonetic/ syllabic and prosodic symbols into a synthetic signal, i.e. the automatic production of speech, through a grapheme-to-phoneme transcription of the sentences to utter. ***Grapheme*** are the letters in a words dictionary listing whilst ***Phoneme*** is the smallest unit of speech that differentiates one word from another.

Text-to-Speech Types and Components

TTS systems can be broadly categorized into two: Parameterised and Concatenative.

Parameterised TTS systems can be further categorized into:

Formant based - use rules based on signal from spoken input.
Articulatory - use model of vocal tract based on electro-acoustics theories.

Concatenative synthesis makes a mathematical model based on phonemes/syllables and uses the speech fragments to act out this model. The resulting speech, though slightly artificial, sounds remarkably like the original speaker who sampled his or her voice. The disadvantage of this type of synthetic speech is that design of a powerful algorithm is required. In addition, memory is needed for each unit of speech.

Concatenative TTS systems can be further categorized into:

Concatenative – word: record all the words you need and concatenate the isolated words to form a sentence. Applicable when a limited vocabulary is required e.g. announcement of arrivals in train station
Concatenative – phoneme: generate the phonemes of the given language and concatenate them to form the desired word. It is a text-to-phone conversion e.g. "Kuja hapa" (come here) approaches to the synthesis Phone-sequence acoustic signal "Ku-ja-ha-pa". Makes more natural sounding speech more like sound sampling.

Prosody Generation

Prosody refers to certain properties of the speech signal which are related to acoustic changes in pitch, loudness, and syllable length. It ensures appropriate rhythm, tempo, accent, intonation and stress are achieved. In concatenative speech synthesis it consists of segmental duration control to model human temporal characteristics. One can use source-filter model to separate excitation signal from vocal tract shape. Vocal tract shape descriptions can then be concatenated and an appropriately smooth fundamental frequency pattern can be added separately. One of the most difficult problems in speech to date is prosodic modeling.

Uses of TTS Systems

The TTS systems find application in varied fields such as: (1) Language education (TTS synthesis can be packaged together with a Computer Aided Learning system to provide tool to learn a new language), (2) Text-to-speech Synthesis for Linguistic and Psycholinguistic Experimentation (TTS synthesizers provide interesting laboratory tools), (3) Settings where data entry possible without keyboard, (4) Telecommunication services: Mobiles, Multimedia, man-machine Communication, (5) Excellent for hands/eyes busy situations, (6) Email Reader, Talking Books and Toys, (7) Vocal Monitoring, and (8) Aid Handicapped persons e.g. Visual disabilities.

Given that Swahili is considered a language with a simple structure in terms of pronunciation as virtually every letter in a word is pronounced and every letter (or letter combination) corresponds to only one Swahili sound, then the question that we try to answer is, "is a Swahili TTS System viable?" The objective of this work is to develop and determine the viability of a Swahili TTS System. The scope of the work will to: (1) Pronounce common valid Kiswahili words, and (2) Extend them to pronounce a sentence and block of text.

Whereas the text normalization component of dates, numbers and abbreviations is essential for a complete TTS in the development of the Swahili TTS this procedure will not be included owing solely to the fact that no comprehensive and valid set of abbreviations exist in this language. A text normalization component identifies numbers, abbreviations, acronyms and idiomatics and transforms them into full text. For example, the number 12 when normalized would be "kumi na mbili". The texts were 'normalised' manually; all abbreviations, numerals, and non-Swahili words were simply deleted.

SWAHILI / KISWAHILI

Swahili in the World

Swahili is a widely spoken language, with somewhere between 45 and 100 million people that use it as their first or second language. Swahili, which means "the coast", is the mother tongue of the peoples that live on the East coast of Africa, that stretches from the South of Somalia to the North of Mozambique, via the islands of Pate, Lamu, Pemba, Zanzibar and Mafia. From East to West, the area of influence of Swahili extends from Tanzania and Kenya through the interior of Congo (previously Zaïre), up to Uganda, Rwanda, Burundi, Zambia and Malawi.

Swahili belongs to the large family of Bantu languages, spoken in all the southern half of the African continent. However, it distinguishes itself from the other Bantu languages by its vocabulary of mixed Bantu and Arabic origins. It is arguably one of the easiest African languages to learn, for it doesn't contain any unpronounceable sounds or "tones" as is found in many western languages.

Swahili Spelling and Pronunciation

Swahili spelling is a good guide to its pronunciation. Virtually every letter in a word is pronounced and every letter (or letter combination) corresponds to only one Swahili sound. The language is a continuous pronunciation of sounds i.e. a group of sounds are combined to build up a given word.

Swahili phoneme is divided into three major categories with reference to English: Consonants, Vowels, and Digraph

The difference between them is that the vowels are pronounced without blocking the nasal cavity whereas with the consonants, one has to block the nasal cavity.

The Swahili Alphabet

The alphabet used for writing Swahili is the same as the international Roman alphabet, with letters chosen to represent the Swahili phones. It also uses digraphs i.e. a sequence of two letters to represent single sounds. According to Steere [Steere, J. 1998], the Swahili alphabet has

Five vowels: *a*, *e*, *i*, *o*, *u*.
Nineteen consonants: *b*, *c*, *d*, *f*, *g*, *h*, *j*, *k*, *l*, *m*, *n*, *p*, *r*, *s*, *t*, *v*, *w*, *y*, *z* (exclude x and q).
Nine Digraphs: *ch*, *dh*, *gh*, *kh*, *ng'*, *ny*, *sh*, *th*, *ng*.

As with many other Bantu languages, Kiswahili has only five vowels, and it is therefore possible to write the language using the Roman alphabet with one-to-one grapheme-phoneme (Total in number 33 =5 vowels + 19 constants + 9 digraphs) correspondence. For the most part this has been achieved, and it is possible to read the language using one-to-one simple grapheme-phoneme correspondences, with a few digraphs, but without any additional Orthographic knowledge [Alcock *et. al*. 1999].

The basic principle which was retained to establish the Swahili alphabet, is that every distinct sound or phoneme should always be transcribed by the same distinct written form (either a single letter, or a cluster of letters), and conversely. There are some exceptions to this rule, but only in words of foreign origin that have not been assimilated completely into the language.

Consonants

Consonants in Swahili generally have English values, but as in the case of vowels, there are considerable differences. Some, however, are pronounced without the aspiration or air puff that characterizes the corresponding English sounds. While most of the consonants are similar to those in English language and do not offer any difficulty, special care must be paid to some of them. For example,

f : it has always the sound of the “f” in “fat”, never that of the “f” in “of”.
g : it is always hard like in “got”. It should never be pronounced soft like the “g” in “age”.
s : it has always the sound of the “s” in “sad”, never that of the “s” in “is” or “easy”.

Vowels

There are five vowel phonemes (distinctive sounds) in Swahili represented by the graphs *a, e, i, o, and u.* They are pronounced openly, without diphthongs. They must always be kept short.

There are important differences between English and Swahili vowels. For one, the Swahili vowels are short and are not diphthongized as are the comparable English ones. For instance, Swahili **’e’** is comparable to the vowel in English ‘**say’** without the lengthening or diphthong; it is also similar to the vowel in ‘**set’**, but not quite as low.

Each vowel should be given its full value whether accented or not. This is also true of vowels in juxtaposition; the vowels in ‘**au’** (or) and **‘bei’** (price) are all pronounced.

Vowel clusters

Unlike in English, two (or three) written vowels that follow each other never merge together to form a single sound. Each keeps its own sound.

Example : **‘ou’** is pronounced “o-oo” as in “go”, **‘au’** is pronounced “a-oo” as in “cow”, **‘ei’** is pronounced “e-ee” as in “bay”, **‘ai’** is pronounced “a-ee” as in “tie”, etc. These are sequences of non-identical vowels that are pronounced separately.

In theory, any vowel can be in succession with any other. It is not uncommon to meet two similar vowels in succession: they must be pronounced as one long vowel e.g. ‘kaa’ (sit)

Digraphs

Digraphs are combinations of consonants: They are pronounced as a single sound unit.

dh and **th** are both spoken “th” in English. **dh** is voiced as in “the”, “this”, “that”, “with” ... While **th** is unvoiced as in “think”, “thin”, “both” : **stakabadhi** (a receipt), **hadithi** (a story).

gh and **kh** are pronounced at the back of the throat. **gh** is voiced and close to the French “r” in “rare” : **ghali** (expensive), **shughuli** (affair, activity). **kh** is unvoiced and corresponds to a scraping of the throat : **subalkheri** (good morning).

ng’ although similar in sound to the English “ng” in “singing” poses a difficulty, for it usually occurs at the beginning of words. It is luckily quite rare: **ng’ambo** (foreign), **ng’ombe** (cow).

Syllable Structure

The most common sequence is that of a vowel preceded by a consonant e.g. ‘ka’ which has the most usage in word construction. A word can start either with a vowel or a consonant but always a word whose origin is Bantu must end with a vowel. This also applies to borrowed words.

A typical Swahili syllable is Consonant Vowel (CV) and virtually every word ends with a vowel. The syllable can be classified into three:

Consonant Vowel (CV), where Consonants precede the vowel.
Consonant or Vowel (C/V).
Digraph Vowel (DV) or Consonant Consonant Vowel (CCV), where the first consonant is generally nasal (m, n, ng').

Stress/accent

In Swahili, only the next-to-the-last syllable in the word receives stress. For instance, stress is found in the syllables that precede the last one e.g. 'nani?' is pronounced as 'nan-nni' i.e. the 'n' is prolonged.

Identification of the syllables of a word is important as it is used in determining stress and intonation in a given word. There are at least two intonation patterns for questions in Swahili, one used in reading questions or in emphatic contexts, another in non-emphatic or normal contexts.

METHODOLOGY: ANALYSIS & DESIGN

Structural Model:

Syllable structure review

As is widely known, human beings can produce/ pronounce infinitely many different sounds but this notwithstanding, every language utilizes only a small group or collection of sounds to construct all possible words in the language.

In Swahili these sounds are the basic alphabet of the language inclusive of the digraphs. Figure 1 shows the alphabet and its subdivision to various subgroups. It is meant to deduce the Speech Units needed to be recorded to represent the search space.

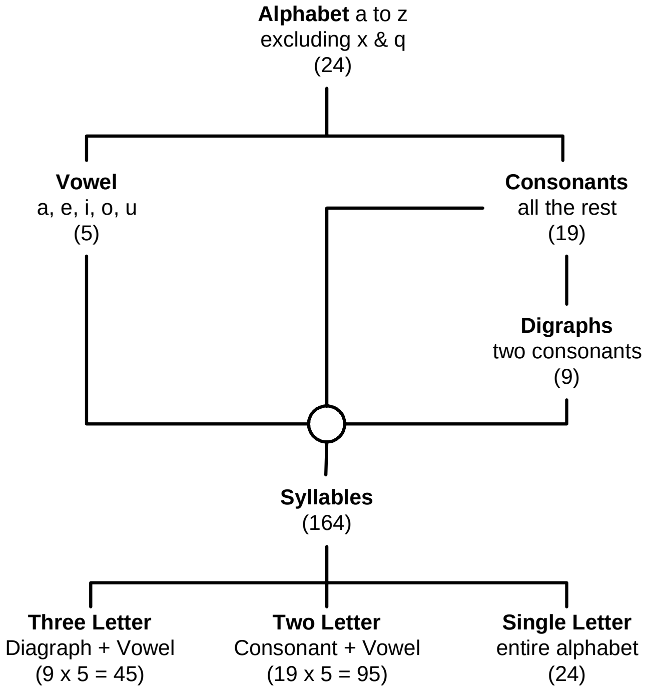


Figure 1: Syllable Structure

It is intended to deduce all the possible combination of syllables to enable a vast variety of words to be catered for.

As explained earlier virtually every letter in a word is pronounced and every letter (or letter combination) corresponds to only one Swahili sound. Swahili sounds can be divided into two major categories with reference to English:

(1) Sounds that are similar to those in English sounds and
(2) Sounds not found in English.

In the system, the syllable is the basic speech unit for concatenation.

Note: 'ch' is more of a digraph but it is generally considered to be part of the alphabet other than a digraph. In the system it was implemented as a digraph.

Syllable structure issues

The function of the syllable is to regulate the structure of complex segments. The syllable serves as a building block for higher-level phonological and morphological processes. Swahili pronunciations are basically based on the syllables. For example: 'ki-ta-bu' (book).

A typical Swahili syllable is
Consonant Vowel (CV), e.g. 'ki'
Consonant Consonant Vowel (CCV) i.e. Digraph Vowel (DV): e.g. 'gha'
Any letter of the alphabet depending on its location:

- Vowel followed by a Constant e.g. 'a-na-e-nda'.
- Consonant followed by a digraph e.g. 'm-b-wa'.
- Vowel followed by a vowel e.g. 'u-a'.

Every word ends with a vowel. This provides a fundamental check of validity of a word.

The importance of syllable structure

The syllable structure is used in the development of the system. Notably, (1) this structure is intended to help in recording the possible syllables and store them in the database. (2) It will also help in the design of the parsing algorithms, as these are the rules to be used to derive the syllables given a word. (3) It will help in the design of the Storage structure for easy and efficient retrieval of the audio files. (4) It will also help as a control to confirm the validity of a word with respect to its (contextual) structure.3.2.

Functional Models Text-to-Speech Synthesizer

The components of a TTS are as depicted in Figure 2. It consists of two major components: (1) Natural Language Processing (NLP), and (2) Digital Signal Processing (DSP).

Natural language processing (NLP)

This component performs the task of decomposing a sentence to its sequence of the parts of speech. It consists of: (i) Text Analyzer- Tokenize a block of text, (ii) Syllable Analysis- Deduce the syllables.

Input: A block of text
Processing: Parsing

- Block of text to Sentences punctuation used as the tokenizing boundaries
- Sentences to words assumed word boundaries

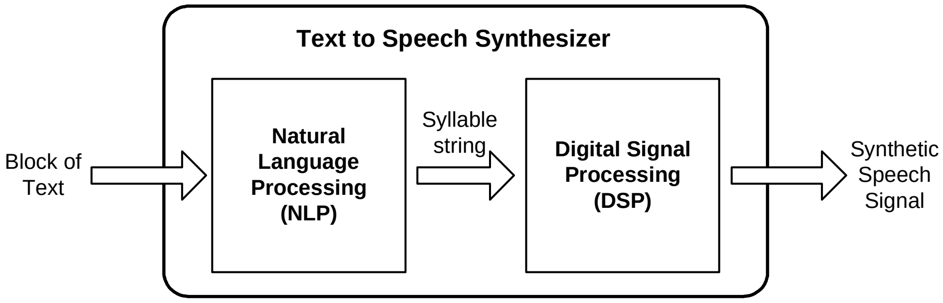

Text to Speech Synthesizer
Block of Text
Natural Language Processing (NLP)
Syllable string
Digital Signal Processing (DSP)
Synthetic Speech Signal

Figure 2: General Structure of a Swahili TTS (adapted from Thierry Dutoit, 2000)

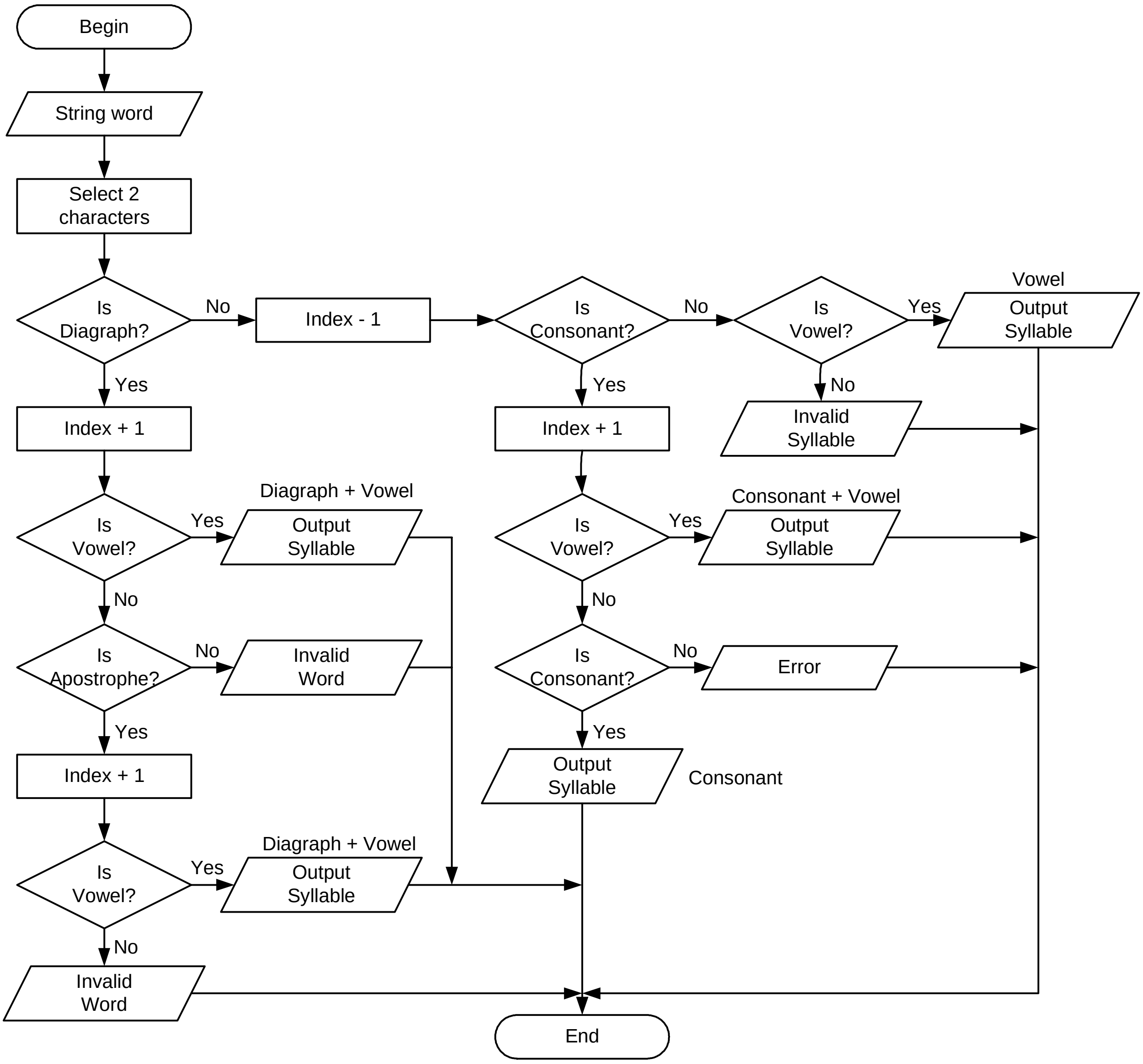


Figure 3: Syllable Generation flowchart

coincide with white spaces

– Words to Syllables. The algorithm is as shown in Figure 3.

Output: String of Syllables e.g. 'ku-ja-ha-pa'.

Digital signal processing (DSP)

This is the computer analogy of dynamically controlling the articulatory muscles and the vibratory frequency of the vocal so that the output signal matches the input requirements. It consists of: (i) Speech Processing-Lookup and concatenation (ii) Sound Processing-Processing the audio.

Input: Syllable String

Processing: Matching Syllables to Strings

- Concatenation
- Smoothening

Output: Pronunciation i.e. speech signal

Basically the above structures were intended for use in the design of the system. They were used to design the

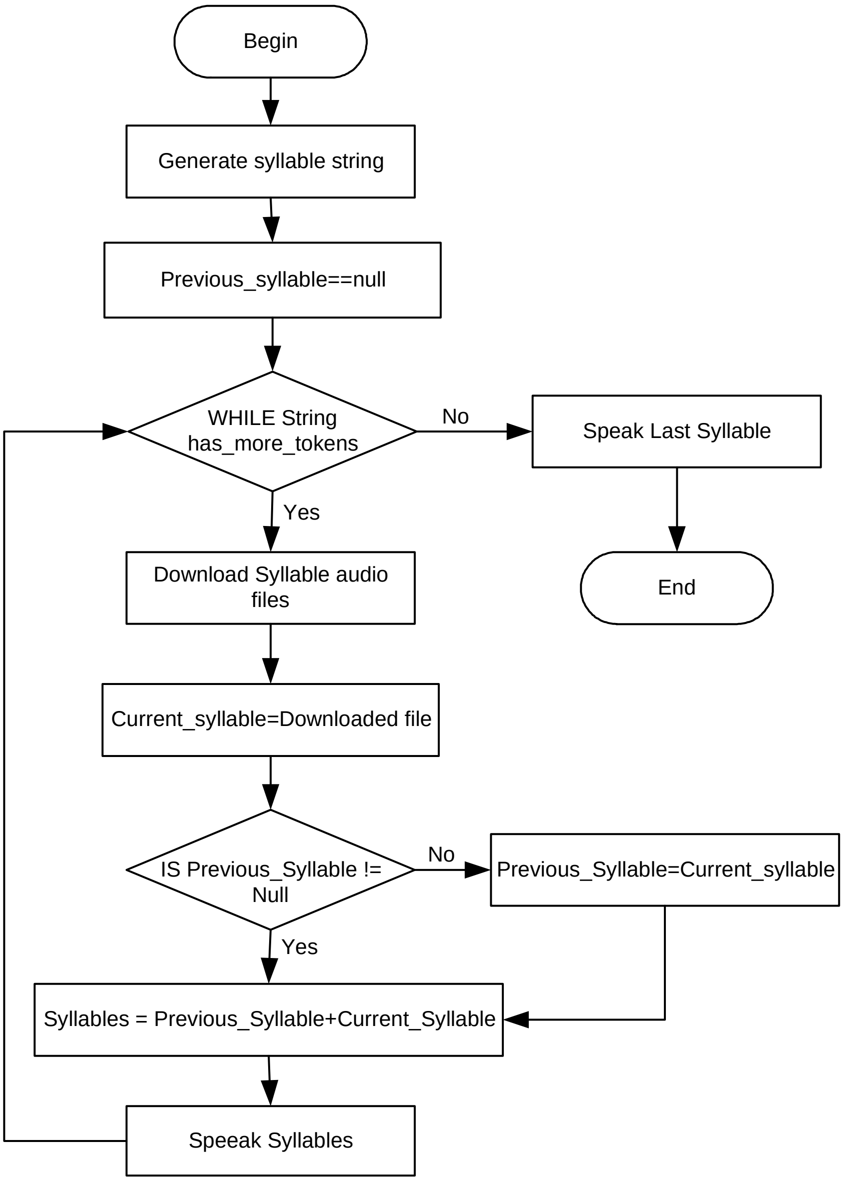

Begin
Generate syllable string
Previous_syllable==null
WHILE String has_more_tokens
No
Speak Last Syllable
Yes
End
Download Syllable audio files
Current_syllable=Downloaded file
IS Previous_Syllable != Null
No
Previous_Syllable=Current_syllable
Yes
Syllables = Previous_Syllable+Current_Syllable
Speeak Syllables

Figure 4: DSP Processing Flowchart

classes and the class relationship diagrams and in the generation of the various algorithms that were required for the system to be fully functional.

Design

The design basically consists of allocating portions of the specification to appropriate modules. It is concerned with the development of an appropriate hierarchy of program modules by transforming the structural analysis models into a specific design, which is physically realisable. In this section we review the various basic design issues of the system.

The methodology used in the design was Object Oriented Design (OOD). This methodology entailed the transformation of the various structural models into classes, Relationship diagrams and the interaction diagrams. The methodology was used as the programming language (Java) used to implement the system. It has the basic foundation of Object Oriented approach, which we intended to take advantage of.

Major classes

These are scheme patterns or templates describing many possible instances of the data objects applicable in the system. Some of the Major Classes implemented were:

1. **Alphabet:** Defines the Swahili alphabet. Error checks whether the provided characters are valid Swahili alphabet elements and syllable generation.
2. **Syllable:** Generates the syllabic structure of a word and error checks its contextual structure.
3. **SwaTalk:** Implements the Speech Engine employing Java sound API. Audio file concatenation.
4. **Sentence:** Implements a method that takes a block of text and breaks it into its constituent sentences and then breaks it further into its constituent words and syllables then reads them.
5. **SwahiliTTS:** Implements the User Interface and is the main class which executes.

Numbers, dates, abbreviations and time are not included in this design. Thus they will be reported as errors if they appear in a word.

Basically the above class structures are intended to be used for coding the system. The Object Oriented Approach was chosen as it gives an easier way of structuring the program and also due to the fact that we used the Java programming language which is basically structured in the OOP concept.

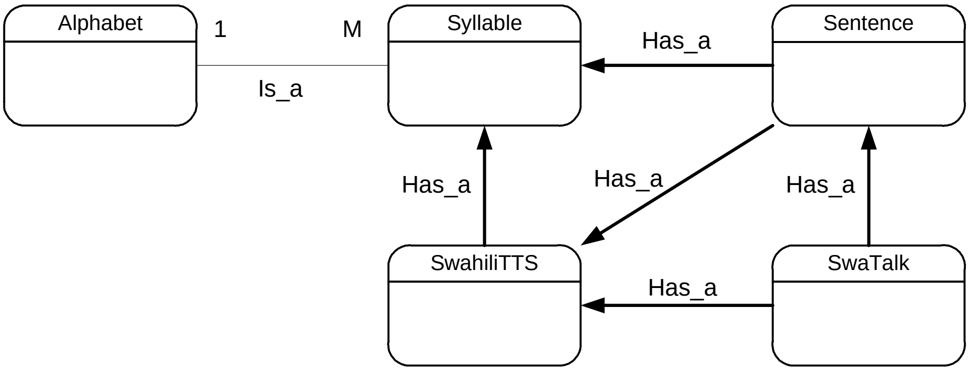


Figure 5: Class Relationship diagram

RESULTS

In this section, a full account of the results obtained is given with the aim of explaining how various issues helped in arriving at the result.

Implementation

It consists of both coding and integration of class modules into a progressively more complete solution of the ultimate system. This was achieved by transforming the design modules (Classes) as described in section 3.3. into structured programs and coding.

The implementation was based on the Object Oriented Programming (OOP) approach, which entailed the integration of the classes in separate files and coding of the methods to achieve the function described in the design. The system was implemented as a Java standalone application. .

Choice of Programming Language (Java)

Java is a modern open developed programming language that has its basic foundation on the Object Oriented paradigm. It is an interpreted language thus making it suitable for machine independent applications. The version of Java used is JSDK1.4.0_02 obtained from Sun Microsystems.

One would then ask why use Java for developing the system? The reasons for choosing Java are:

(1) Powerful: basically it provides many functionalities to perform various operations making the programmer's task of concentrating on the systems he/she is developing easier.

(2) Portability: Java is a portable language as it is interpreted which would make it suitable for platform.

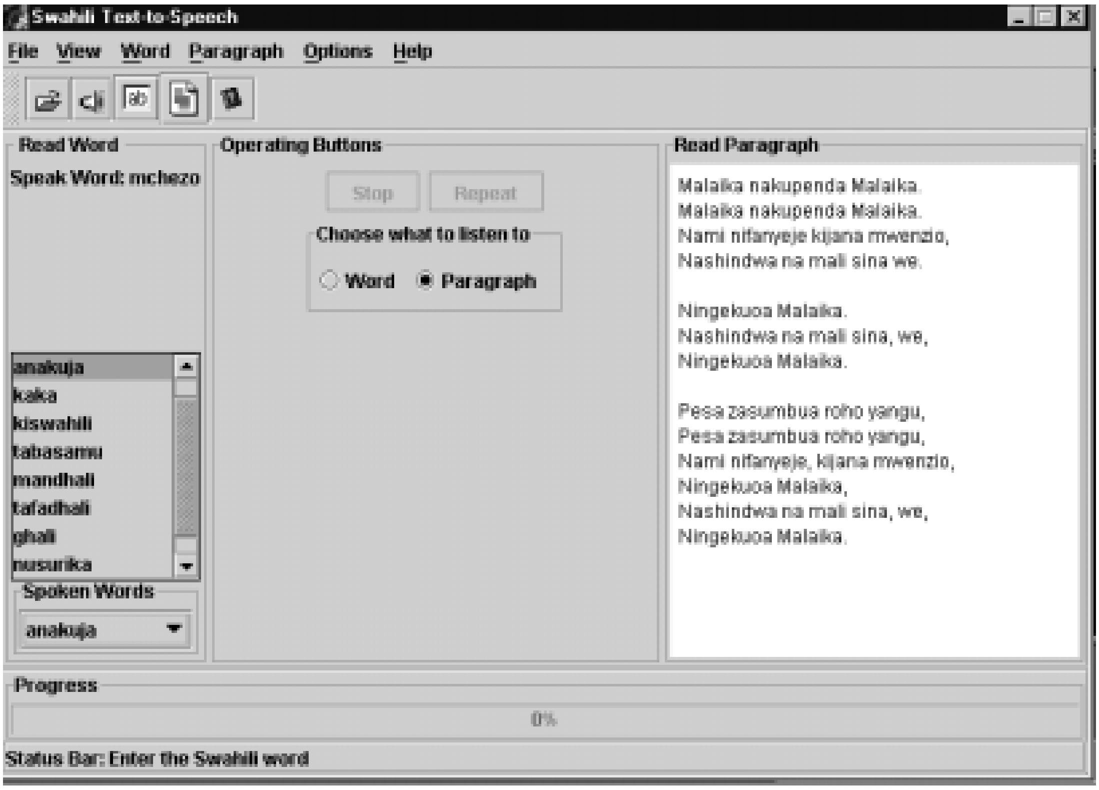
(3) Web Integration: due to its portability it is greatly used in web applications (creation of applets) and the system developed can be of great importance for web applications.

System Requirements

Processor: Tested on a 233MHz processor. It is advisable to use a faster processor for better performance.
Memory: Tested on a 64MB of RAM, its response is slow thus a RAM greater than this will improve performance greatly.
Operating System: Tested on Window NT but can run on any OS with a JVM or JRE.
Hardware: Requires Sound Card and sound Drivers.
Multimedia Kit: Speakers or Earphones.

Sample Screen

Figure 6 illustrates the main User Interface. The user provides the word or paragraph to be read. One can also open a text document (*.txt) that contains Swahili text. One then clicks speak from the respective menu item or click the sound icon on the toolbar menu.



Speak Word: mchezo

anakuja
kaka
kiswahili
tabasamu
mandhali
tafadhali
ghali
nusurika

Malaika nakupenda Malaika.
Malaika nakupenda Malaika.
Nami nifanyeje kijana mwenzio,
Nashindwa na mali sina we.

Ningekuoa Malaika.
Nashindwa na mali sina, we,
Ningekuoa Malaika.

Pesa zasumbua roho yangu,
Pesa zasumbua roho yangu,
Nami nifanyeje, kijana mwenzio,
Ningekuoa Malaika,
Nashindwa na mali sina, we,
Ningekuoa Malaika.

Status Bar: Enter the Swahili word

Figure 6: Sample Screen of Swahili TTS System

DISCUSSION

Achievements

Figure 6 shows that most of the objectives were achieved and a workable application was obtained. Although the pronunciation was not actually smoothened the system still provides the user with the capabilities of word pronunciation.

Constraints

Strict Word validity from user was not achieved, as there are no Swahili digital Dictionaries available, thus the system checks the validity of a word structurally. For example, a word such as "Rogue" will be taken as a valid word as it is syntactically correct but semantically incorrect. Further, Also smoothening and prosody were not accomplished, as design and implementation of algorithms to achieve them were difficult considering that we were dealing with sound byte files. In spite of these drawbacks it is our considered opinion that the system can be tuned to a functioning capability given the recommendations outlined in section 5.3 below.

Suggestion of Further Work

1. Further work should be carried out to study issues on speech synthesis and provide algorithms for prosody to achieve proper synthesis.
2. Extending the system to handle numbers and dates by creating a parse algorithm.
3. Intonation should be handled. Since only the last letter is affected and one can achieve this by using Long vowels e.g. "ee, ii".
4. Integrating the system into applications such as web services and email which play a vital role in information technology.

CONCLUSION

Using Speech Synthesis ideas we were able to develop a system that pronounces common valid Kiswahili words provided by a user and extended them to include sentences and paragraphs. Clearly a Swahili Text-to-Speech System is viable and an achievable task.

REFERENCES

Alcock, Katherine J. Ngorosho and Damaris. Kiswahili Spelling Project, 1999, Tanzania.

Christophe d'Alessandro & Jean Sylvain. Synthetic Speech Generation. LIMSI-CNRS, Orsay France. www.cslu.cse.edu\HLTServer, 2002.

Cornelia E., Sevim, Sevde & Alex . English Loan-Words in Swahili, 2003.

D. H Klatt, Speech & Audio Processing & Recognition, 1992, Lecture Notes.

Dafydd Gibbon . Speech Synthesis Systems. CEST. 1998 www.coral.lili.uni-bielefeld.de/classes/winter98/Exphon/SynthesisDBNotes.

Ireri Mbaabu. Sarufi ya Kiswahili-Sauti za Kiswahili, 1992. Longman Kenya Ltd, Nairobi.

Michael W. Macon, ECE 580 - Speech Synthesis, 1999. www.ece.ogi.edu/~macon/Dictionaries and the Standardization of Spelling in Swahili, Institute for Kiswahili Research, 1998. University of Dar es Salaam, Tanzania.

Juergen Schroeter. The Fundamentals of Text-to-Speech Synthesis, 2000.

M. Edgington and A. Lowry. Residual-Based Speech Modification Algorithms for Text-to-Speech Synthesis, 2000.

Mark Hasegawa-Johnson. *Lecture notes in Speech Production, Speech Coding, and Speech Recognition*, 1998. University of Illinois, www.ifp.uiuc.edu/~hasegawa/notes/

Sun Microsystems, Inc. *Java^{TM} Sound API Programmer's Guide* Version 1.0, 1998.

Thierry Dutoit. The MBROLA Project, 2002, http://tcts.fpms.ac.be/synthesis/mbrola.html.

Thierry Dutoit. *An Introductory Course on Speech Processing ,2000* , http://tcts.fpms.ac.be/cours/1005-08/speech/

Thierry Dutoit, An Introduction to Text-to-Speech Synthesis, 1999, Kluwer Academic.

Tony Morales, Speech Synthesis-Applications to Universal Computer Access, 1999.

Univ. of Birmingham, 1999. Speech Synthesis, Part I: Concatenative synthesis, 1999 University of Birmingham, School of Electronic & Electrical Engineering lecture notes.