**Available online at** www.uofk.edu
Proceedings Vol. 1 pp. 198 - 203
Annual Conference of Postgraduate Studies and Scientific Research
17-20 February 2012-10-17 Friendship
Hall, Khartoum

# Design and Development of an Arabic Text-To-Speech Synthesizer

**Mazin Ahmed El-Hag Hamad, Mustafa Ibrahim Yousif Hussain & Mohammed Ali Hamad Abbas**

*Department of Electrical and Electronics Engineering, Faculty of Engineering, University of Khartoum,*

**Abstract**: Computers that can interact with humans via speech had become a dream for scientists since the early stages of the computer age. Scientists from different areas have been interested in producing human speech artificially for more than two decades and they made a lot of research to reach this dream. Many advances in Text To Speech (TTS) synthesis systems have been achieved in languages like English and French. Languages like Arabic have been largely ignored. This paper presents a TTS system for Arabic that uses allophone/diphone concatenation method. It takes a text as input and produces corresponding speech in Arabic. In this system, the output is available in one male voice only. Since Arabic is a verbal language, the TTS engine developed can be used also for other verbal languages with some minor modifications involving the specific language.

**Keywords:** *Text To Speech; TTS; Arabic; Speech Synthesis; Allophone/Diphone Concatenation.*

## 1. INTRODUCTION

Text-to-speech synthesis is a research field that has received a lot of attention and resources during the last couple of decades – for excellent reasons. One of the most interesting ideas (rather futuristic, though) is the fact that a workable TTS system, combined with a workable speech recognition device, would actually be an extremely efficient method for speech coding [1]. Speech synthesis/text-to-speech is an artificial intelligence science (AI is to do something at which at the moment human are better [2] down the track of the Natural Language processing (i.e. It studies the problems of automated generation and understanding of natural human languages).
A TTS system is a system that can convert a given text into speech signals. The source of this text can be very different. While the output of an Optical Character Recognizer (OCR) can be an input for this system, the text that is generated by a language generation system can also be an input for a TTS system. A block diagram of a general TTS engine is depicted in Figure 1.

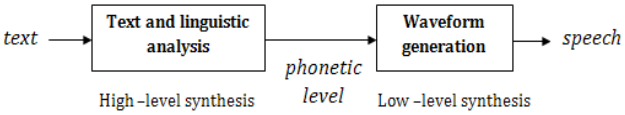


**Figure (1).** General Structure of a TTS Synthesizer

The aim of an ideal TTS system is to be able to process any text that a human can read. For example, a TTS system should be able to read numbers, handle abbreviations, resolve different spellings for a word, etc.
Text-to-speech synthesis systems are an essential component of modern human–machine communications systems and are used to do things like read email messages over a telephone, provide voice output from GPS systems in automobiles, etc. Another important application is in reading machines for the blind, where an optical character recognition system provides the text input to a speech synthesis system.

## 2. Concatenative Synthesis

One of the most popular methods of synthesizing speech from text is by stringing together, or concatenating, prerecorded words, syllables, or other speech segments [3]. This avoids many of problems encountered in phoneme-to-phoneme synthesis, such as the coarticulatory effect between neighboring speech sounds [4]. Still, even words do not usually occur in isolation: the words immediately preceding or following a given word influence its articulation, its pitch, its duration and stress – often depending on the meaning of the utterance.
Concatenative synthesis (the so called cut and paste synthesis) uses actual short segments of recorded speech that were cut from recordings and stored in an inventory ("voice database"), either as "waveforms" (uncoded), or encoded by a suitable speech coding method. Synthetic voices are made by concatenating units of sound that have been previously stored in a reference database. The contents of these units and methods of concatenation vary, but the

principle of concatenation is universal for TTS involving all but the briefest messages. Nowadays, the use of actual speech waveforms has become increasingly popular, where stored waveforms of various sizes are fetched as needed, with adjustments made mostly at unit boundaries, but sometimes more generally throughout the utterance [5]. Concatenative Synthesis Method uses a large database of source sounds, segmented into units, and a unit selection algorithm that finds the sequence of units that match best the sound or phrase to be synthesized, called the target. The selection is performed according to the descriptors of the units, which are characteristics extracted from the source sounds, or higher-level descriptors attributed to them. The selected units can then be transformed to fully match the target specification, and are concatenated.

A self-explanatory block diagram of a typical concatenative TTS system is shown in Figure 2.

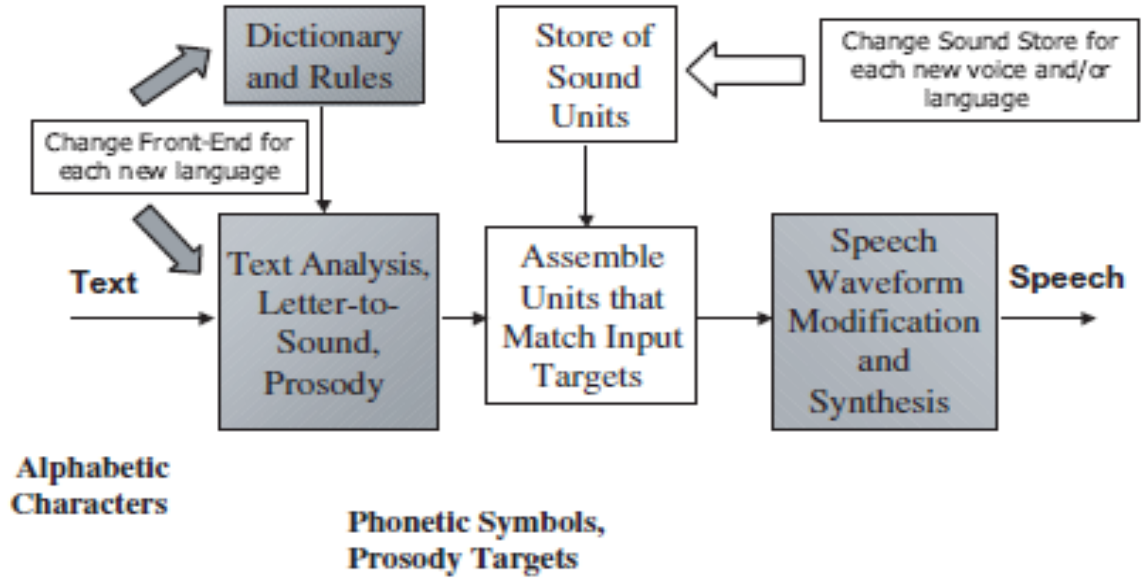


**Figure (2).** Block diagram of a concatenative text-to-speech system

## 3. Requirements For A Concatenative Synthesis System

Any Concatenative Sound Synthesis system must perform the following tasks, which may sometimes perform implicitly. These tasks or steps are:

- **Analysis:** The source sound files are segmented into units and analyzed to express their characteristics with sound descriptors.
- **Database:** Source file references, units and unit descriptors are stored in a database. The subset of the database that is pre-selected for one particular synthesis is called the corpus.
- **Target:** The target specification is generated from a symbolic score (expressed in notes or descriptors), or analyzed from an audio score (using the same segmentation and analysis methods as for the source sounds).
- **Selection:** Units are selected from the database that match best the given target descriptors according to a distance function and a concatenation quality function. The selection can be local (the best match for each target unit is found individually), or global (the sequence with the least total distance if found).
- **Synthesis:** is done by concatenation of selected units, possibly applying transformations.

## 4. ARABIC TTS SYSTEM MODEL

Figure 3 shows the suggested model for the Arabic TTS system. The input to the program is a plain Arabic text; this text is processed through different module operations, producing a read out text.

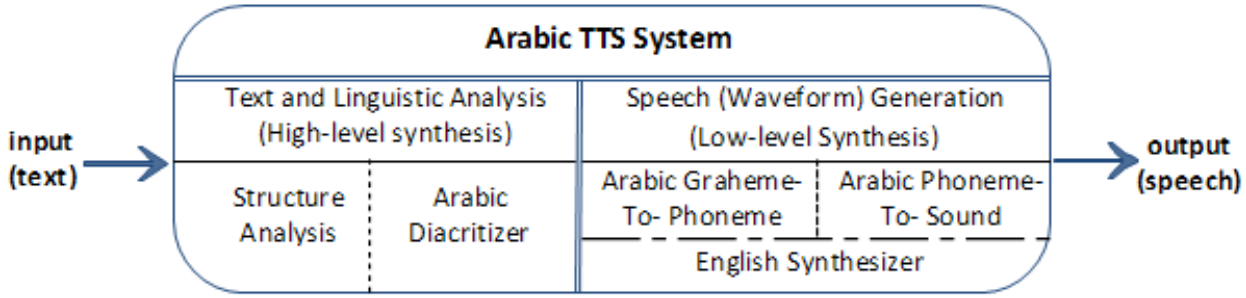


**Figure (3).** Full model of Arabic TTS system.

In the following sections we describe the specifications of the various processing stages:

**Text and Linguistic Analysis (High-level Synthesis)**

**a) Structure Analysis**

Figure 4 depicts the input/output for each sub-module showing each function.

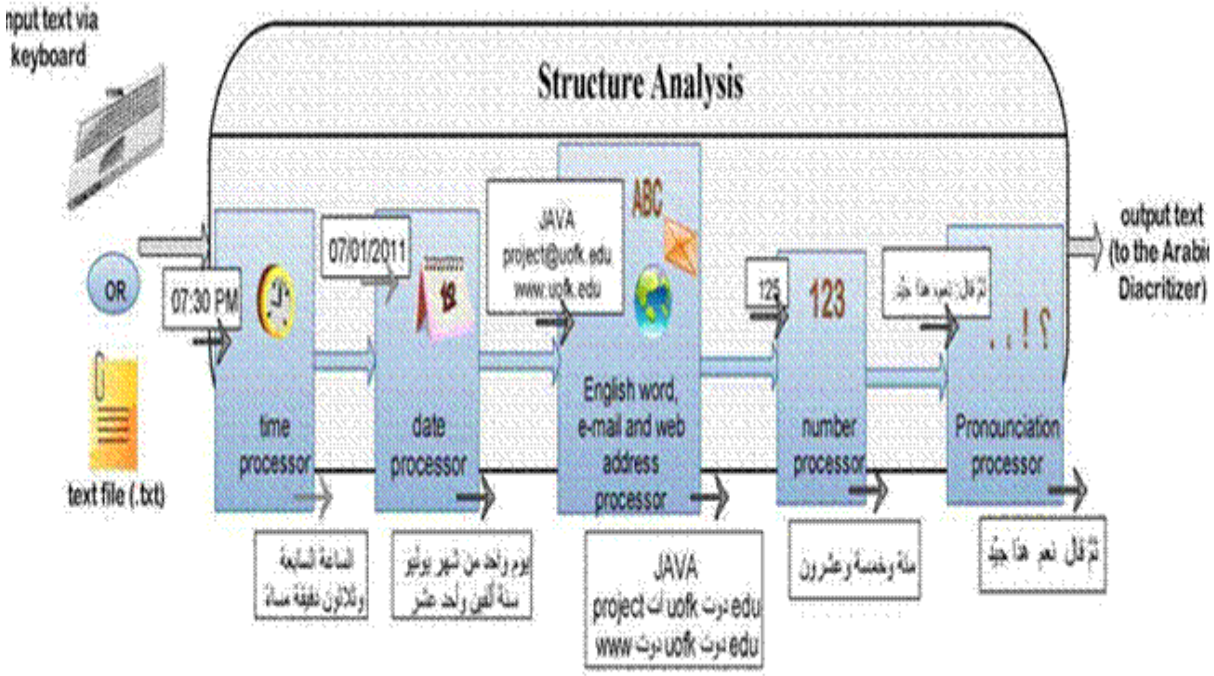


**Figure (4).** Functions of Structure Analysis module

This module is responsible for manipulation of both punctuations ("."/ "،" / ” ؟"/ "!"etc) and special text categories (times, dates, English words, e-mail or website addresses and numbers). The input for this module is the whole program input coming either from the user directly through keyboard or from stored text file (.txt). The function of this module is to produce an intermediary text (output) composed only from words (Arabic + English words, e-mails and website addresses) that represent the input for the next stage, namely the Arabic Diacritizer. Also, the input text is processed to determine where paragraphs start and end, sentences and punctuations.

**b) Arabic Diacritizer**

The diacritization process is shown in Figure 5. One of the biggest challenges that faced the Arabic text preprocessing is that the text must be diacritized to be read correctly by the synthesizer, so in the preprocessing step each character and its diacritic must be determined. A half-diacritized lexicon of sample Arabic words (deacritization database) is developed using Microsoft Access 2003(see Appendix...). The fully diacritized Arabic word can be obtained by passing the retrieved half-diacritized word (from diacritization database) back to the user to add the missing diacritic of the last letter in each word, otherwise the default diacritic “ً “ is assumed and added automatically to the word’s end. It is worth mentioning that all spaces (silence) and English words remaining from the previous stage passed this step untouched “English words have no diacritization”. The output of this module includes diacritized Arabic words plus English words from the previous stage in the sequence entered by the user (a FIFO queue is used to reserve the order).

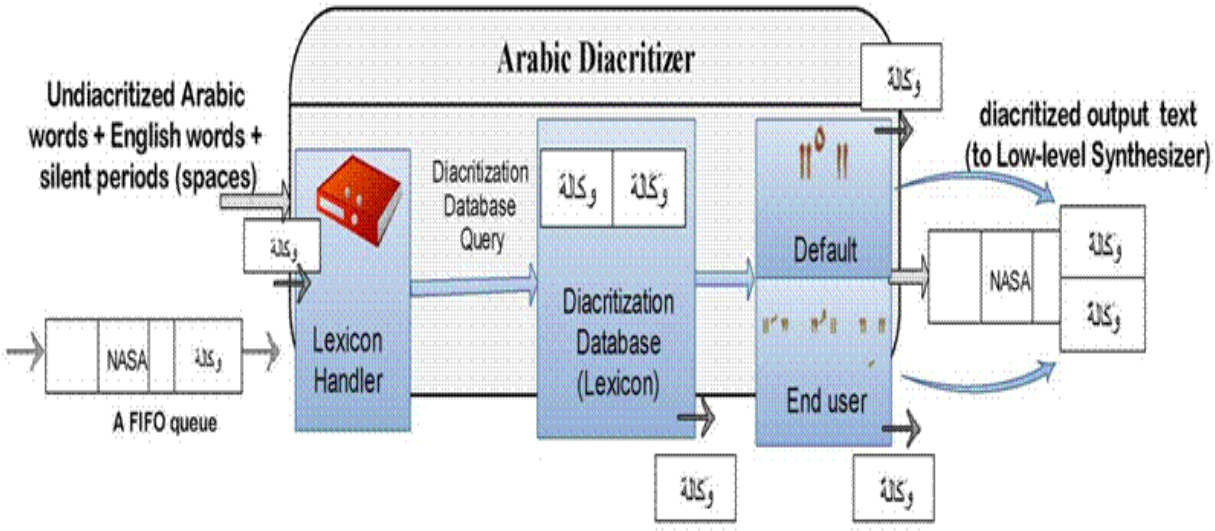


**Figure (5).** Arabic Diacritization module

**Speech (Waveform) Generation (Low-level Synthesis)**

**a) Arabic Grapheme-to-Phoneme conversion module**

The Arabic language has about 445 different phonemes which are classified to vowels and constants as shown in Figure .

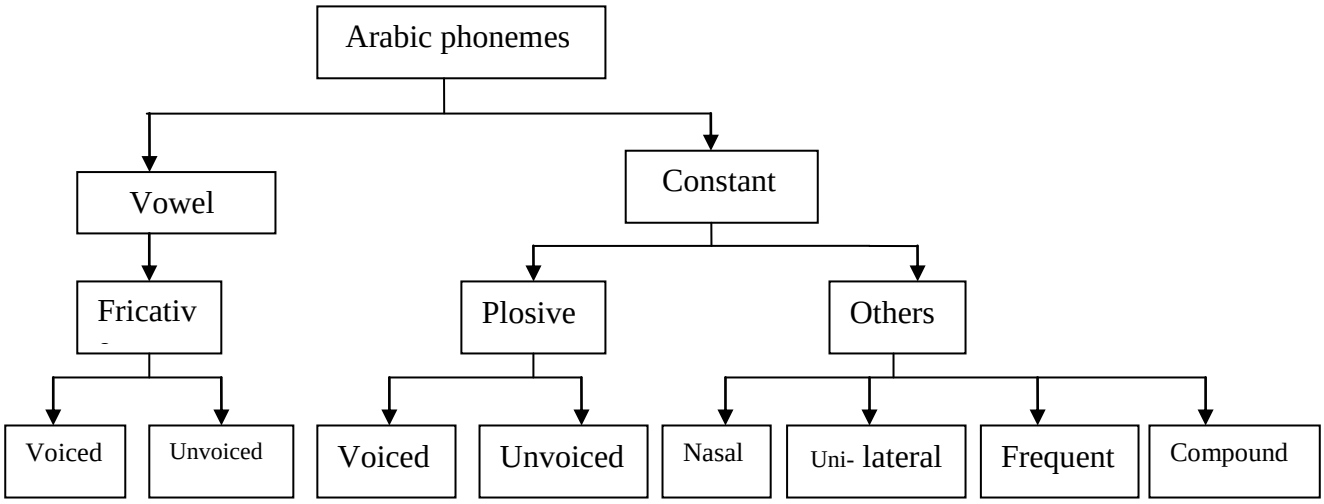


**Figure (6).** Classification charts of Arabic phonemes. [6]

In order to assign the closest phoneme (later, sound) to each grapheme taking into consideration that (in Arabic) pronunciation depends on the diacritic, where there are thirteen different diacritic, each of them gives different pronunciation for the letter. For that reason, sixteen sound files have been recoded for each character and its diacritic. For example: the letter "ف" have the following set of phonemes depending on its diacritic as:

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| فَ | فُ | فِ | فاً | فٍ | فٌ | فا | فُوْ | فِيْ | فَّ | فُّ | فِّ | فْ | فًا | فْيْ | فِّيْ |

**b) Arabic Phoneme-to-Sound conversion module (Arabic Synthesizer)**

After Arabic grapheme-to-phoneme conversion, all Arabic words are represented, each, by its phonemes corresponding to sequence of wave files that must be read sequentially to synthesize the target Arabic word, the following procedure is used:

Since required wave files names are available implicitly (as parts of the phonemic word representation), consecutive file names must be rewritten in an order that is suitable to the sequential wave player. This is achieved by simply entering a back slash (\) between each (.wav) file name and another, and preceding the first (.wav) file name by the absolute path in which the Arabic synthesis database is resident. For example, if the Arabic synthesis database is in folder that has an absolute path name as (C:\ArabicTTS\SynthesisDatabase) and we want the synthesizer to read aloud the word " قَلَمْ", then the following string should be used for the source file name: C:\\ArabicTTS\\SynthesisDatabase\\ga,la,mam2.wav

**c) English speech generation module (English Synthesizer)**

The main target of the project is to synthesize the Arabic text; thus the English synthesizer module is not of much interest. However, since the system also deals with special classes of text that cannot be written in Arabic language such as e-mail, website addresses and abbreviations which are provided in English language; an English TTS engine should be integrated with the Arabic synthesizer. English TTS systems have reached high stages in development compared to the Arabic ones, that's now these systems are integrated in a lot of daily user applications (PDF and text processors).

**d) Microsoft Speech API**

The Speech Application Programming Interface or SAPI is an API developed by Microsoft to allow the use of speech recognition and speech synthesis within Windows applications. In general all versions of the API have been designed such that a software developer can write an application to perform speech recognition and synthesis by using a standard set of interfaces, accessible from a variety of programming languages. In addition, it is possible for a 3rd-party company to produce their own Speech Recognition and Text-To-Speech engines or adapt existing engines to work with SAPI. In principle, as long as these engines conform to the defined interfaces they can be used instead of the Microsoft-supplied engines. Also, the Speech API is a freely-redistributable component which can be shipped with any Windows application that wishes to use speech technology [7].

## 5. RESULTS AND DISCUSSIONS

The program final layout is in the form of windows based program that allow the user to go through the steps of language processing and modifying –if needed- in specified steps and then play the synthesized text. Figure 7 shows the final form of the program window.

In some applications, for example reading machines for the blind, the speech intelligibility with high speech rate is usually more important feature than the naturalness. On the other hand, prosodic features and naturalness are essential when we are dealing with multimedia applications or electronic mail readers. The evaluation can be made at several levels, such as phoneme, word or sentence level, depending on what kind of information is needed.

The "Testing and Evaluation of TTS Systems" is done by selecting a group of random people and allows them to try the program while going through a questionnaire that will be used to evaluate the project throughput. A questionnaire was designed randomly to assess the intelligibility (clearness), naturalness, sound quality and the pronunciation on the level of phoneme word and sentence. The group was consisting of 20 people of different professions and language knowledge in order to obtain a good assessment (overview of the program's operation).

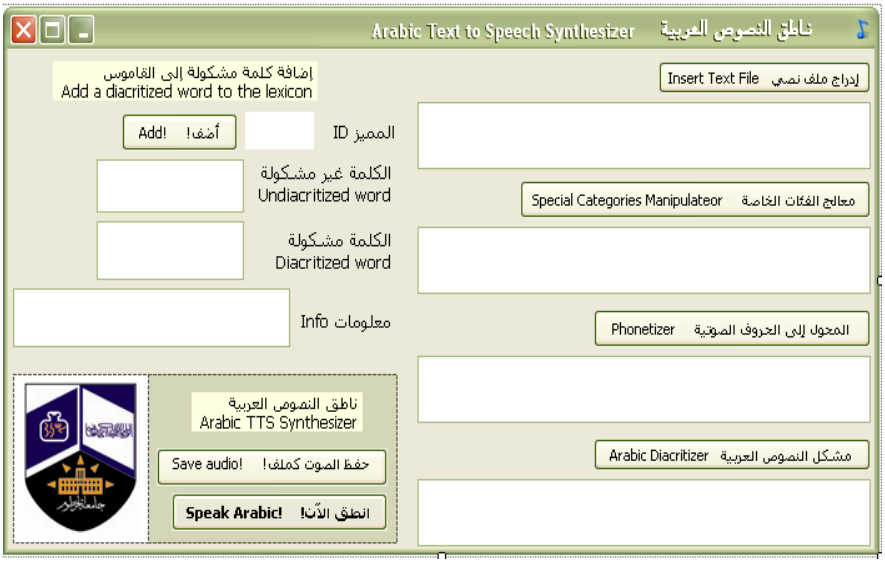


**Figure (7).** The final form of the program

By analyzing results, it was implied that, when it comes to the intelligibility of the system, the Arabic TTS Synthesizer System is successful. The participants can hear what is being said and recognize changes with the synthesized speech. The majority of both words and sentences were correctly recognized and perceived of the majority of the listeners and the evaluation of the overall quality of the system is satisfying at this stage. Since the concatenated speech is produced from a prerecorded phoneme units; the discontinuity problem arise in its clearest form because of the large co-articulatory effects that exists between adjacent phones and longer units are more likely to result in higher quality synthesis, given that the rate of concatenation is lower than in the case of shorter units, also the quality with some consonants may vary considerably and the controlling of pitch and duration may be in some cases difficult, especially with longer units.
Putting all that into consideration, the degraded naturalness of the concatenated word that has appeared in the results is reasonable. The results also showed that some phonemes were difficult to get and must be rerecorded such as letters diacritized with "ً". The concentration results show that the concentration could be as normal as listening to the news.

## 6. CONCULSIONS

In this paper, the development and evaluation of the TTS system for Arabic language based on the concatenation synthesis method. The design of an allophone/diphone database and the natural language processing modules developed has been described. We have demonstrated a working system that gets an Arabic text as input and generates corresponding speech signal for this text is obtained. Although the software part of the system is nearly complete according to aims of the projects, the allophone/diphone database is not perfectly complete yet. However, the results obtained from the current database are quite satisfactory and gives an indication that a complete database will match the aims of the project much better.

## 7. ACKNOWLEDGEMENTS

Special thanks are due to Dr. El-Sadig Babiker (Faculty of Arts, UofK) for valuable assistance. The authors would like to acknowledge the technical staff at the Electronics Laboratory (UofK) for their hospitality and willingness to provide services and facilities under their control.

## REFERENCES

[1] Huang X., Acero A., Hon H., Ju Y., Liu J., Mederith S., and Plumpe M., "Recent Improvements on Microsoft's Trainable Text-to-Speech System", Whistler, Proceedings of ICASSP 97 (2): pp. 959-934,1997.
[2] Dr. Elsadig Babiker, Artificial Intelligence course note, 2006.
[3] [Olive, J.P., Concatenative Syllables, Progress in Speech Synthesis, Springer, New York, pp. 261 – 262, 1996.
[4] Schroeter J., Articulatory Synthesis and Visual Speech, Progress in Speech Synthesis, Springer, New York, pp. 179 – 184, 1996.
[5] Browman, C., "Rules for demi-syllable synthesis using LINGUA language Interpreter", Proceedings of International Conference on Acoustics, and signal Processing, IEEE, 1980.
[6] Dr. El-Sadig Babiker, "A Hybrid Rough Sets K-Means Vector Quantization Model for a Neural Networks Based Arabic Speech Recognitio", a PhD Thesis University Putra Malaysia, 2002.
[7] Wikipedia.[Online]en.wikipedia.org/wiki/Speech_Application_Programming_Interface, last accessed on June 27, 2011