SSML for Arabic Language

Noor Shaker[1], Mohamed Abou-Zleikha[1], and Oumayma Al Dakkak[2]

[1] Department of Artificial Intelligence, University of Damascus, Damascus, Syria
noor.shaker@gmail.com, mhd_az@hotmail.com
[2] HIAST P.o.Box 31983, Damascus, Syria
odakkak@hiast.edu.sy

Abstract. This paper introduces SSML for using with Arabic language. SSML is part of a larger set of markup specifications for voice browsers developed through the open processes of the W3C. The essential role of the markup language is to give authors of synthesizable content a standard way to control aspects of speech output such as pronunciation, volume, pitch, rate, etc. across different synthesis-capable platforms. We study SSML, the validity to extend SSML for Arabic language by building Arabic SSML project for parsing SSML document and extracting the speech output.

Keywords: Speech synthesis, SSML markup language, Arabic text-to-speech system.

1 Introduction

SSML (Speech Synthesis Markup Language) is one of the standards that have been developed by Voice Browser Working Group to enable access to the Web using spoken interaction; it is designed to provide a rich, XML-based markup language for assisting the generation of synthetic speech in Web and other applications.

In the objective of building a complete Text-to-Speech system of standard spoken Arabic with a high speech quality, based on SSML, we've built our Arabic SSML system.

The input to this system is an XML document, containing the vocalized Arabic text enclosed in SSML tags. An expert system based on TOPH (Orthographic-Phonetic Transliteration) language [1,2] and [3] transcripts the text into phonetic codes, then MBROLA diphones [4] are used to generate speech. In fact, MBROLA permits the control of some prosodic features such as fundamental frequency F0 and duration. It enabled us to construct our prosodic models and test it. In what follows, we discuss automatic speech generation in our Arabic SSML.

2 Arabic SSML

To achieve our goals, we adopted the same steps as introduced in the W3C [7] see Figure 1. They are: XML parse, Structure analysis, Text normalization, Text-to-phoneme conversion, Prosody analysis, and Waveform production.

P. Sojka et al. (Eds.): TSD 2008, LNAI 5246, pp. 657–664, 2008.

Arabic SSML is composed of various modules that represent each stage, and has the capability of parsing SSML documents.

When dealing with Arabic Language there are several specific points to be taken into account, which are not handled in SSML: the choice of the correct pronunciation of numbers and plural nouns because they depend on the context and this feature is not supported in SSML. In fact, if we want to say “nine pens” or “nine sheets” in Arabic, the pronunciation of the word “nine” changes according to the gender of the following word. This pronunciation changes again if the word is definite, and with it the two words permute. Similar and more complicated cases arise for other numbers.

In addition, in Arabic, plural can be of two types: dual or trial and above, the morphology and the pronunciation change according to the plural and to the part of speech of the word (subject, object, adverb. . .)

To handle the above problems and similar ones, we customized some tags by adding some attributes that provide the context and allow us to choose the correct word to be pronounced (gender, number, definite or not, POS. . .).

3 Structure of Arabic SSML System

The architecture is designed by specifying the data format serving as input and/or output for each module. This design enables us to easily extend the system to other languages and to change entire modules without affecting the others.

Our Arabic SSML system architecture is described in Figure 1.

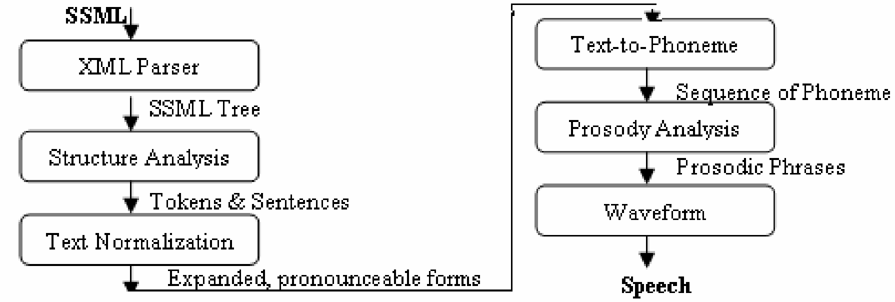

Fig. 1. The architecture of the Arabic SSML system

It contains an XML parser, an Analysis Module, and a Text Normalization Module; the output of this Module constitutes the input to the speech synthesizer containing the Text to phoneme Module, a Prosody generator and a speech waveform generator. We describe briefly, each of the system components.

3.1 XML Parser

The input markup language is translated by this module into a tree. We convert this extracted tree into another more detailed tree, upon which subsequent modules will operate. The tree we build consists of nodes, each node corresponds to a tag in the SSML document.

For example: if we have the following SSML document:

<speack> <p>
الساعة الآن
</p> <say-as interpret-as="Time" format ="hh:mm">10:15</say-as>
</speack>

We build the corresponding XML tree, and from it, we build another one that contains all the details to be added by each module. In this stage the tree will be as in Figure 2.

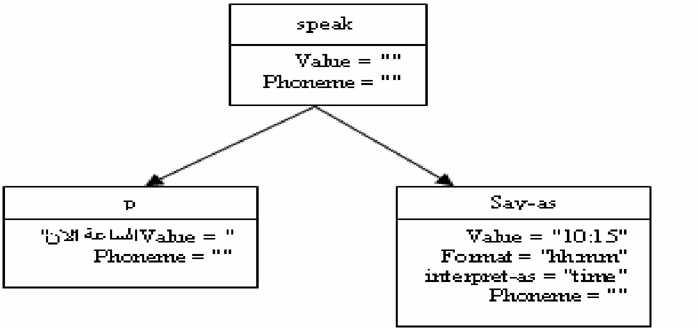

Fig. 2. The extracted tree

3.2 Structure Analysis

The structure of a document influences the way in which a document should be read. The input text needs to be segmented into sentences and words, relying on blanks and punctuation marks. For example, there are common speaking patterns associated with paragraphs and sentences.

The **p** and **s** elements, defined in SSML, which stand for Paragraph and Sentence respectively, explicitly indicate document structures that affect the speech output.

3.3 Text Normalization

Numbers, special symbols, abbreviations, have to be conveniently expanded to normalize the text into a standard format. Text normalization is an automated process that performs a conversion of the written form (orthographic form) into the spoken form.

By the end of this step, the text to be spoken has been converted completely into tokens.

In this module, we process the following SSML elements [8,5]:

<sub> element The substitution element indicates that the text in the alias attribute value replaces the contained text for pronunciation.

For example, ل.س may be spoken as "ليرة سورية" /Syrian pound/, etc.

_{ل.س}

<say-as> element This element allows the author to indicate information on the type of text contained within the element, and to specify the level of detail to render the contained text. The say-as element has three attributes:

- **interpret-as:** always required, indicates the content type of the contained text. interpret-as value: date, time, telephone, characters, cardinal, and ordinal.
- **format:** optional attribute, its legal value depends on the value of the interpret-as attribute. For instance: an Interpret as Time may have the format: hh:mm:ss or hh:mm followed by am. or pm., etc.
- **Detail:** optional attribute, indicates the level of detail to be rendered.

For example:

- date

  <say-as interpret-as="date"> ٢٠٠٢/١٠/٣ </say-as>
  Should be interpreted as
  "الثالث من تشرين الأول عام ألفين و اثنين" ;/3rd of October, 2002/
  or
  "الثالث من شوال عام ألفين و اثنين" ;/3rd of Shawaal, 02 /
  Here, the say-as element should be expanded to include a way to differentiate between Gregorian calendar date and Hejri calendar date.
- Time

  <say-as interpret-as="time" format="hms24"> ٩:٢١:٣٠ </say-as>
  Should be interpreted as
  "التاسعة و واحدٌ و عشرون دقيقة و ثلاثون ثانية صباحاً"
  or "التاسعةِ وواحدٍ وعشرين دقيقة و ثلاثين ثانية صباحاً" /9:21:30 am /.
  In fact, Arabic has diacretizations which are often omitted short vowels in the written language. These short vowels change according to the part of speech (POS) of the words. Other words change phonemes depending on their POS like male plural which formed from singular by adding /uun/ if the word is a subject or /iin/ if the word is an object or is preceded by a preposition [10]. In the above examples the actual spoken word (“ثلاثين”,”ثلاثون”) /30/ is as male plural, depends on the context, which should be provided to determine the correct pronunciation of the word.
- Telephone number
  <say-as interpret-as="telephone" format="963">٠٩٨٥٩٥٩١</say-as>
  This is a telephone number in Syria (country code is "963"). As a mobile phone number, it should be interpreted as
  "تسعة ستة ثلاثة صفر تسعة ثمانية خمسة تسعة خمسة تسعة واحد"
- Cardinal number
  <say-as interpret-as="cardinal" detail=",">٢٣٤</say-as>
  Should be interpreted as
  "مئتان و أربعٌ وثلاثون" or " مئتين و أربع و ثلاثين " ; /234 Two hundred.../
  Here again, the correct words to be spoken depend on the context which must be provided. It depends on the POS of the word, its gender, its number, etc. the same note applies to ordinal numbers.

- Ordinal number
  <say-as interpret-as="ordinal">٣٢</say-as>
  This example will likely be spoken as ;/32/
  "الثانية و الثلاثون" Or "الثاني و الثلاثون” or "الثانية و الثلاثين" Or ”الثاني و الثلاثين”
  In Arabic, the actual spoken numbers must cope with Arabic numbers' rules which depend on implied context which must be provided. The words indicating the numbers can be themselves male or female. In some cases they take the gender of the numbered objects, in others they take the opposite.

  In our implementation of the system we took into consideration all these rules and incorporated them in the system, and added the proper attributes in order to get the appropriate output speech.

3.4 Text-to-Phoneme Conversion

Once the synthesizer processor has determined the set of words to be spoken, it must derive pronunciations for each word. Word pronunciations may be conveniently described as sequences of phonemes.

In this module, we process the following SSML element:

<phoneme> element This element allows a phonemic sequence to be provided for any word sequence. All other words in the text are converted to its corresponding set of phonemes using TOPH after its adaptation to Arabic Language.

For example

<phoneme alphabet="ipa" ph="ibda;"> إبدأ </phoneme>

3.5 Prosody Analysis

Prosody is the set of features of speech output that includes pitch, timing, pausing, speaking rate, and emphasis on words, etc. Producing human-like prosody is important for making speech sound natural, and for correctly conveying the meaning of the spoken language. The prosody module adds prosody information for the phoneme according to what the author indicates in SSML tags.

The system generates automatic prosody for all other sentences where the prosody is not given explicitly. This prosody relies only on the punctuation to give the type of the sentence: Exclamation if it ends by the exclamation point "!", Interrogative if it ends by a question mark "?". Continuous affirmation if ended by comma "," and Affirmation (Long) if ended by a point "." [2]

This module processes the following SSML elements:

<emphasis> element This element implies speaking the text with emphasis. The attributes are:

- **level:** the optional level attribute indicates the strength of emphasis to be applied. Defined values are "strong'", "moderate", "none" and "reduced". The default level is "moderate". For example
  <emphasis level = "strong"> ما </emphasis> أروع عملك هذا! ;/What marvelous is your work! /

<break> element The break element is an empty element that controls the pausing or other prosodic boundaries between words. The attributes on this element are:

- **strength:** optional attribute having one of the following values: “none”, “x-weak”, “weak”, “medium” (default value), “strong”, or “x-strong”. This attribute is used to indicate the strength of the prosodic break in the speech output.
- **time:** optional attribute indicating the duration of a pause to be inserted in the output in seconds or milliseconds.

If a break element is used without strength or time attributes, a break will be produced according to the type of the sentence.

For example

رجاءً اضغط الرمز واحد أو انتظر سماع الصوت <break time="3s"/> ;/Press one or wait till you hear the tone/

<prosody> element The prosody element permits control of the pitch, speaking rate and volume of the speech output. These attributes are all optional. They are the following [3]:

- **pitch:** the baseline pitch for the contained text, Legal values are: a number followed by “Hz”, a relative change or “x-low”, “low”, “medium”, “high”, “x-high”, or “default”.
- **contour:** sets the actual pitch contour for the contained text.
  <prosody contour="0%, +20Hz" (10%, +30%) (40,+10Hz)> صباح الخير </prosody> ;/good morning/
- **range:** the pitch range (variability) for the contained text. Legal values are: a number followed by “Hz”, a relative change or “x-low”, “low”, “medium”, “high”, “x-high”, or “default”.
- **rate:** a change in the speaking rate for the contained text. Legal values are: a relative change or “x-slow”, “slow”, “medium”, “fast”, “x-fast”, or “default”.
  سعر القطعة <prosody rate = "-25%"> 100 ل.س</prosody> ;/The price is 32SP/
- **duration:** a value in seconds or milliseconds for the desired time to take to read the element contents.
- **volume:** the volume for the contained text in the range 0.0 to 100.0 specifying a value of zero is equivalent to specifying “silent”. Legal values are: number, a relative change or “silent”, “x-soft”, “soft”, “medium”, “loud”, “x-loud”, or “default”. The default value is 100.0.

<voice> element This element is a production element. It requests a change in the voice speaker. Attributes are: age, variant, name, gender, xml:lang (language). They are all optional.

3.6 Waveform Production

This module uses the phonemes and the generated parameters from the prosodic information to produce the audio waveform. The format for the output is compatible with MBROLA since it’s the tool we use to produce the speech output [4].

4 Implementation

An interface has been designed and implemented, which allows the user to easily investigate parts of the system's architecture tree, each intermediate processing result can serve as input, and any subsequent processing result can be output.

Individual processing steps can be carried out, allowing the user to understand the function of each module, or to investigate the source of an error. We followed the design architecture described in MARY [6].

For text-to-phoneme and prosody, we used the system described in [3] and incorporated it in our interface. Our System has been developed and tested using Java programming language. The system handles the following SSML tags: speak, p, s, sub, say as, phoneme, emphasize, break, prosody; including all their attributes.

5 Future Perspectives

1. It would be useful if SSML contains elements to modify prosody parameters to express emotions such as (happiness, sadness, anger, surprise, fear).
2. Some text type like messages and SMS contains symbols like 😆 😁 😉 and it would be useful if SSML contains elements to express such symbols.
3. Extend SSML to create audio versions of the mathematical expressions. That may include an extension to say-as element to handle math formats like $\sqrt{a+b/d}$.

6 Conclusion

An automated tool has been developed for Arabic SSML. An overview of the processing components of the system has been given. It has been described how a system-internal tree-based data representation, can be used to make partial processing results available outside the system. The advantages of this design architecture are:

1. All intermediate processing results can be made visible.
2. These intermediate results can be modified and fed back as input into the system for future improvements.
3. Each module can be easily replaced by another one in case they have the same input and output data type.

These features are very helpful for teaching purposes, for non-technical users and for research and development of TTS synthesis.

References

1. Ghneim, N., Habash, H.: Text-to-Phonemes in Arabic. Damascus University Journal for the Basic Sciences 19(1) (2003)
2. Al-Dakkak, O., Ghneim, N., Abou Zliekha, M., Al-Moubayed, S.: Prosodic Feature Introduction and Emotion Incorporation in an Arabic TTS.In: ICTTA (2006)

3. Al-Dakkak, O., Ghneim, N., Abou Zliekha, M., Al-Moubayed, S.: Prosodic Feature Introduction and Emotion Incorporation in an Arabic TTS. In: IEEE Int. Conf. on Information and Communication Technologies.In: ICTTA. Damascus-SYRIA (2006)
4. Dutoit, T., Pagel, V., Pierret, N., Bataille, F., van der Vrecken, O.: The MBROLA project: towards a set of high quality speech synthesizers free of use for non commercial purposes. In: ICSLP 1996. Proceedings. Fourth International Conference (1996)
5. Bonardo, D., Baggia, P.: (Loquendo) SSML 1.0: an XML-based language to improve TTS rendering (2005)
6. Schröder, M., Trouvain, J.: The German Text-to-Speech Synthesis System MARY: A Tool for Research. Development and Teaching (2003)
7. ScanSoft Speech Synthesis Markup Language (SSML) Version 1.0, W3C Recommendation 7 September (2004), http://www.w3.org/TR/speech-synthesis/
8. SSML 1.0 say-as attribute values W3C Working Group Note 26 May (2005), http://www.w3.org/TR/2005/NOTE-ssml-sayas-20050526