A Unified Framework for Multilingual Text-to-Speech Synthesis with SSML Specification as Interface *

WU Zhiyong (吴志勇)[1,2,**], CAO Guangqi (曹光琦)[1], MENG M. Helen (蒙美玲)[1,2], CAI Lianhong (蔡莲红)[2]

[1] Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong SAR, China;

[2] Tsinghua-CUHK Joint Research Center for Media Sciences, Technologies and Systems,
Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China

Abstract: This paper describes the design of a unified framework for a multilingual text-to-speech (TTS) synthesis engine – Crystal. The unified framework defines the common TTS modules for different languages and/or dialects. The interfaces between consecutive modules conform to Speech Synthesis Markup Language (SSML) specification for standardization, interoperability, multilinguality, and extensibility. Detailed module divisions and implementation technologies for the unified framework are introduced, together with possible extension for algorithm research and evaluation of TTS synthesis.

Key words: text-to-speech (TTS) synthesis; multilingual; unified framework; speech synthesis markup language (SSML)

Introduction

There is ever increasing need for text-to-speech (TTS) technology in various applications on different platforms such as information kiosk, e-leaning, and message access over internet and/or hand-held devices. Several TTS engines have been developed during past decades [1-5]. However, these engines have used different architectures for different specific languages. This leads to the difficulty and complexity while trying to port from one engine to another. Moreover, it also introduces the inconvenience in evaluating the function of modules for different engines.

This paper reports our recent work on designing a unified framework for a multilingual TTS synthesis engine – Crystal – conforming to the Speech Synthesis Markup Language (SSML) specification [6,7].

The objective of Crystal is to develop a unified framework for TTS involving multiple and mixed languages (e.g. Chinese Putonghua and Cantonese) which are commonly and widely used in South China. The unified framework aims to *define the basic common TTS modules* which are to be shared among different languages and dialects. This will ensure the functional consistency and reduce the redundancy for developing TTS engines with different languages. Meanwhile, the unified framework tries to *define the standard interfaces* between consecutive TTS modules by virtue of the world wide SSML specification. It provides a consistent mechanism for controlling different TTS engines. This will ensure the compatibility and interoperability between different TTS engines and will ease the development of applications which involve several TTS engines of different languages. Furthermore, the unified framework also intends to *provide good extensibility* to evolve to support new enhanced features or interfaces in the future for both common interfaces and internal modules of TTS engine without compromising the interoperability.

Received: 2009-04-03

* Supported by the Innovation and Technology Fund – Guangdong-Hong Kong Technology Cooperation Funding Scheme (No. GHP024/06) of Hong Kong SAR and the National Natural Science Foundation of China (No. 60805008)

** To whom correspondence should be addressed.
E-mail: zywu@sz.tsinghua.edu.cn; Tel: 86-755-26036870

1 Motivations of Using SSML

SSML specification is a World Wide Web Consortium (W3C) recommendation [6,7], which is designed to provide an XML-based markup language for assisting the generation of synthetic speech in a variety of application contexts.

We have designed a unified framework for a multilingual TTS engine – Crystal. In the unified framework, SSML specification serves as the common interface for both input and output of all TTS modules. SSML specification has been chosen as the interface in the unified framework for the following benefits.

1.1 Standardization

Support of international standards is very important for the development of applications which often involves cross-vendor or cross-engine platforms.

As a W3C recommendation, SSML specification provides an international and standardized way to instruct the design and implementation of not only TTS

applications but also TTS engines.

1.2 Interoperability

Interoperability may be achieved if various TTS engines or vendor-specific components all conform to SSML specification.

All TTS modules conforming to SSML specification are reconfigurable to be used together; they can be inserted, removed, rearranged or replaced with little effort at either module or engine level. TTS modules conforming to SSML specification are also exchangeable between different engines; this could save most of efforts for porting from one engine to another or integrating modules of one engine into other engines.

1.3 Multilinguality

The objective of this work is to develop a unified framework for multilingual TTS synthesis. Given this scope, the interface should be appropriate for rendering any language, such as Chinese Putonghua, Chinese Cantonese, English, etc.

SSML specification has been designed to support a variety of different human natural languages by providing element "<lang>" and attribute "xml:lang" for identifying the language, and also element "<w>" for assisting the identification of word for the languages that do not use white space as word boundary identifier (e.g. Chinese).

1.4 Extensibility

SSML specification is based on the eXtensible Markup Language (XML). Therefore, it can be easily extended by adding new elements or attributes for either common interfaces or internal modules.

For example, in expressive TTS synthesis, the PAD (Pleasure, Arousal, and Dominance) emotional model is borrowed from psychology study as quantitative measurement of word expressivity [8]. To label PAD value of a word, element "<w>" can be extended by adding "PAD" attribute as shown in Table 1. In this way, "PAD" attribute can be supported without modifying any other modules excepting those that really deal with PAD value for prosody prediction.

Table 1 Illustration on the extensibility of SSML

太平山顶是香港**<w PAD="1 1 0">最</w>**受欢迎的景点.
(The Victoria Peak is the **most** popular sights in Hong Kong.)

2 Unified Framework for Multilingual TTS

To ease the development of applications that will involve multiple languages, a unified framework for multilingual TTS has been designed. As we are focusing on Chinese Putonghua and Cantonese at present, we will mainly take these two languages (or dialects) as examples for detailed description in this paper. However, the proposed unified framework is language independent, and could be easily extended to other languages for multilingual TTS synthesis.

Several aspects should be considered while designing the unified framework: 1) TTS engines for different languages must obey the same module division and workflow; 2) The external interfaces should conform to SSML specification; 3) Common modules can be shared between engines for different languages to reduce redundancies; and 4) Different engines can use their own way and extended internal interfaces to handle language specific issues in the internal modules.

Module division and flowchart of the proposed unified framework is illustrated in Fig. 1. "XML parse", "Structure analysis", "Text normalization", "Text-to-phoneme conversion", "Prosody analysis", and "Waveform production" are six major external modules corresponding to six steps for converting input document to synthetic speech as suggested in SSML specification [6,7]. All the interfaces of these six external modules conform to SSML specification. There are one to three internal modules that undertake real function implementation of corresponding external module. The interfaces of these internal modules are also based on SSML specification, but with necessary extensions for some modules to assist the implementation of TTS engine (will be elaborated later).

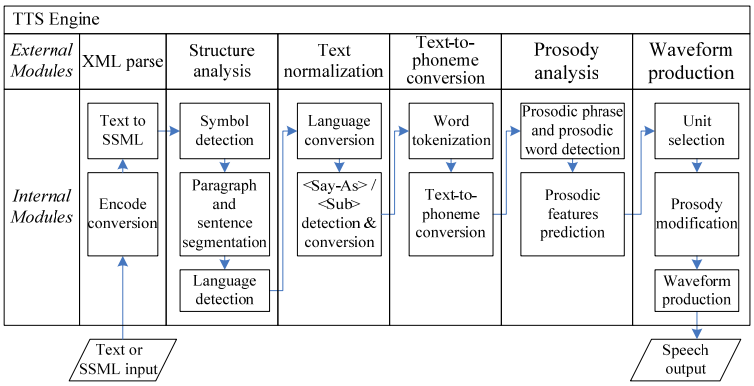

Fig. 1 Module division and flowchart of the proposed unified framework of TTS engine

Fig. 2 demonstrates the hierarchy inheritance between language independent base engine and two language specific engines (i.e. Putonghua and Cantonese engines). The base engine implements all common and language independent modules to be shared among different languages including "XML parse", "Structure analysis", and "Waveform production". Putonghua and Cantonese engines implement all other language dependent modules which are inherited from the base engine in the unified framework. This design enables different engines share the same framework while keeping the language dependent behavior, which ensures the functional consistencies and reduces the redundancies as much as possible.

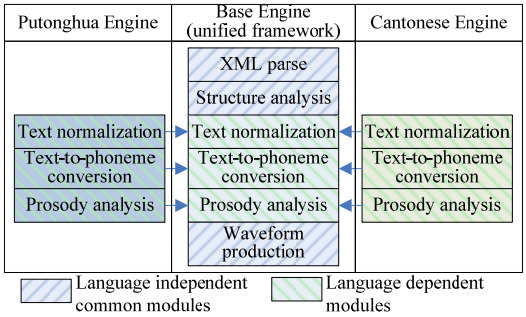

Fig. 2 Inheritance hierarchy of language independent and language dependent modules

3 SSML Interface and Implementation for Each Module in the Unified Framework

To explain how each module works and to introduce SSML input and output interfaces for six external modules in the unified framework, an example text input with partial SSML tag is given in Table 2. In the example, there are ambiguity for Chinese words "说道" (means *say*) and "道哥" (means *Brother DAO*). The SSML element "<w>" is used to serve as word boundary identifier for word disambiguation; and thus "道哥" is treated as the desired word.

Table 2 An example input text with partial SSML tag

我说<w>**道哥**</w>，你还欠我 HK$10,000.00 呢！有冇搞錯！
(**Brother DAO**, you still owe me HKD 10,000.00! Be honest!)

3.1 XML parse

The function of this module is to convert input text from any encodings to 16-bit Unicode Transformation Format (UTF-16) [9], and to construct a well-formed SSML document with proper header and elements.

UTF-16 is introduced as the only internal encoding scheme during the whole processing of the framework. All input texts in other encodings (e.g. GB18030 for Simplified Chinese and BIG5 for Traditional Chinese) are first converted into UTF-16 for later processing.

After processing, a well-formed SSML document will be constructed as the output (Table 3). The proper header (e.g. xml version and encoding) and element "<speak>" are added. The value of attribute "xml:lang" in element "<speak>" will be "zh-cmn" for Putonghua TTS synthesis, and "zh-yue" for Cantonese TTS. Partial SSML tags (e.g. "<w>") are retained unchanged in the SSML output.

Table 3 SSML output of XML parse module

```
<?xml version="1.0" encoding="UTF-16" ?>
<speak version="1.1"
  xmlns="http://www.w3.org/2001/10/synthesis"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.w3.org/TR/speech-synthesis/
  synthesis.xsd" xml:lang="zh-cmn">
    我说<w>道哥</w>，你还欠我 HKD$10,000.00 呢！
    有冇搞錯！
</speak>
```

3.2 Structure analysis

The function is to segment input document into paragraphs and sentences, and to detect the natural language of the written content for each character piece.

The structure of a document influences the way in which a document should be read. For example, there are common speaking patterns associated with paragraphs and sentences. In Chinese, several punctuations such as full-stop (".", "。"), exclamation mark ("!", "！") are usually served as the signature of sentence delimiter. However, some of them may also appear in many special constructs of symbols such as URLs (e.g. http://www.cuhk.edu.hk), numeric expressions (e.g. 12,345.67, 12'34"), etc.

To eliminate the impacts of special constructs during structure analysis, we detect and mark these constructs with SSML "<say-as>" element as early as possible. All kinds of possible special constructs are collected and classified into different categories (Table 4). Regular expressions (RE) are used to detect these constructs before structure analysis. Thereafter, ambiguous punctuations within special constructs are marked as SSML "<say-as>" element which no longer introduces trouble during structure analysis (Table 5).

Table 4 Taxonomy of special constructs (symbols)

Category	Sub-Category	Examples
Net	IP	127.0.0.1
	Net	http://www.sohu.com/
Duration	Duration	1h23'23"88, 1h23:345
Date	YMD	2006/03/12, 2006-03-12
	MDY	03/12/2006, 03-12-2006
	DMY	12/03/2006, 12-03-2006
Time	Time	6:20, 7:30 am, 23:11:13
Measure	Measure	USD14, HK$15, HK$16/kg
Range	Range	15-16kg, 12-14, ₵12-₵43
Fraction	Fraction	1/3, 233/324
	Proportion	106:89
Number	Telegram	+852-62785001, 13800138000
	Cardinal	+3.1415926, 1,234.343
Symbol	Symbol string	Fwef234fe

Input document may contain multiple written languages. A simple statistical method is proposed to detect the natural language of written content for each sentence. The sentence will be assigned the written language with the highest probability:

$$\hat{l} = \arg\max_{l\in\{S,T\}} P_l = \arg\max_{l\in\{S,T\}} \frac{N_{Pl} + N_O/2}{N_T} \quad (1)$$

Where l can be S or T for Simplified or Traditional Chinese respectively; P_l is the probability that current sentence is written in language l; N_T is the total number of characters in current sentence; N_{Pl} is the number of peculiar characters in sentence that belong to language l only; and N_O is the number of overlapped characters in sentence which belong to both languages. For example, in sentence "有冇搞錯！", "冇" and "錯" are peculiar characters for Traditional Chinese, other characters are overlapped. The written language will be detected as Traditional Chinese given that P_T=0.7 and P_S=0.3 (N_{PT}=2, N_{PS}=0, N_O=3, and N_T=5).

The final SSML output of structure analysis module is shown in Table 5, where some attributes of element "<speak>" are neglected for clarity. "xml:lang" attribute of element "<s>" marks the written language of sentence. "zh-Hans" and "zh-Hant" represents Simplified and Traditional Chinese respectively.

Table 5 SSML output of structure analysis module

```
<speak version="1.1" ..... xml:lang="zh-cmn">
  <p>
    <s xml:lang="zh-Hans">我说<w>道哥</w>，你还欠我
    <say-as interpret-as="measure"
      format="measure">HKD$10,000.00</say-as>呢！</s>
  </p><p>
    <s xml:lang="zh-Hant">有冇搞錯！</s>
  </p>
</speak>
```

3.3 Text normalization

This module converts all other written languages to engine-specific language and all written form of special construct (e.g. symbols, numeric expressions, etc.) into corresponding spoken form (e.g. Simplified Chinese characters for Putonghua TTS engine).

No strict one-to-one mapping exists between Traditional and Simplified Chinese characters. For example, there is no character in Simplified Chinese which *directly* corresponds to Traditional Chinese character "冇" (*no*), but character with the same *semantic meaning* "没"; Character "发" in Simplified Chinese may correspond to two characters "髮" (頭髮, *hair*) and "發" (發財, *enrichment*) in Traditional Chinese depending on the word context.

For language conversion between Traditional and Simplified Chinese, we have defined a mapping table with word context information. As a result, the Traditional Chinese sentence "有冇搞錯！" in the previous example will be converted into Simplified Chinese sentence "有没搞错！" for Putonghua TTS synthesis.

All special constructs have been detected in the previous structure analysis module and converted to "<say-as>" elements. This module converts all these elements into corresponding Chinese characters according to translation rules for each specific construct. For example, "HKD$ 10,000.00" in the previous example will be converted to "<w>港币</w><w>一万块</w>" in Simplified Chinese, where "<w>" elements are added directly according to translation rule.

In Table 6, the value of "xml:lang" attribute in "<speak>" element is changed to "zh-cmn-Hans" as there is only one written language (Simplified Chinese) for the entire document.

Table 6 SSML output of text normalization module

```
<speak version="1.1" ..... xml:lang="zh-cmn-Hans">
  <p>
    <s>我说<w>道哥</w>，
    你还欠我<w>港币</w><w>一万块</w>呢！</s>
  </p><p>
    <s>有没搞错！</s>
  </p>
</speak>
```

3.4 Text-to-phoneme conversion

The function of this module is to tokenize sentence into words according to lexicon, and then to derive the pronunciation for each word.

Word boundary information is important for determining the meaning and pronunciation of the words in sentence. For instance, the sentence "南京市长江大桥" has two different meaning and pronunciation because of different word tokenization result: the first meaning is "The major of Nanjing city, Jiang Daqiao" for word tokenization "<w> 南 京 </w><w> 市 长 </w><w>江大桥</w>"; and the second meaning is "The Nanjing Long River Bridge" for word tokenization "<w>南京市</w><w>长江</w><w>大桥</w>".

Part-of-speech (POS) information is a very important property of word. The pronunciation of a Chinese word is related to its POS. For example, the character "种" has the pronunciation "zhong4" (*plant*) in word "种花" (*plant the flower*) as its POS is "verb"; while the pronunciation is "zhong3" (*seed*) in word "花种" (*flower seed*) because its POS is "noun".

We utilize a POS bigram model for word tokenization. Let the sentence be $\mathbf{C}=C_1C_2\ldots C_N$, where C_i is the ith character in the sentence. If word tokenization result for the sentence is $W_1W_2\ldots W_K$, then $\mathbf{C}$ can also be represented by the word tokenization result $\mathbf{C}=\mathbf{W}=W_1W_2\ldots W_K$. Let the corresponding POS for word W_k is T_k, and the POS sequence from W_1 to W_k is

S_k (i.e. S_k=$W_1W_2…W_k$, S_K=**C**), then the optimized word tokenization result could be computed as:

$$\begin{aligned}\hat{\mathbf{W}} &= \arg\max P(W_k, S_k \mid \mathbf{C}) = \arg\max P(W_k, S_k) \\ &= \arg\max P(W_k \mid T_k) P(S_k) \\ &= \arg\max P(W_k \mid T_k) P(T_k \mid S_{k-1}) P(S_{k-1}) \\ &= \arg\max P(W_k \mid T_k) \prod P(T_k \mid T_{k-1})\end{aligned} \quad (2)$$

$P(W_k|T_k)$ represents the probability for the *k*th word being W_k given its POS T_k, and $P(T_k|T_{k-1})$ is the transition probability from previous POS T_{k-1} to current POS T_k. The two probabilities could be computed as:

$$P(T_k \mid T_{k-1}) = \frac{F(T_k T_{k-1})}{F(T_{k-1})}, P(W_k \mid T_k) = \frac{F(W_k, T_k)}{F(T_k)} \quad (3)$$

Where $F(T_k)$, $F(T_kT_{k-1})$ represents the appearance frequency of POS T_k and POS sequence T_kT_{k-1} in the corpus; while $F(W_k,T_k)$ is the appearance frequency of POS T_k in all the instances of word W_k.

After word tokenization using POS bigram model, the sentences are tokenized into words with POS information. The appropriate pronunciation of a word is then retrieved from the lexicon by matching both word and POS information.

"Tone Sandhi" should also be processed. For example, the original Pinyin for character "一" (*one*) is "yi1" with tone 1; it will change to tone 2 when it appears before a character with tone 4. In Table 6, "一万" (*10,000*) should be pronounced as "yi2 wan4" instead of "yi1 wan4". Some of the results of Table 6 are shown in Table 7.

Table 7 SSML output of text-to-phoneme module

```
<s> ……
  <w role="r"><phoneme alphabet="pinyin" ph="ni3">你
    </phoneme></w>
  <w role="d"><phoneme alphabet="pinyin" ph="hai2">还
    </phoneme></w>
  <w role="v"><phoneme alphabet="pinyin" ph="qian4">欠
    </phoneme></w>
  <w role="r"><phoneme alphabet="pinyin" ph="wo3">我
    </phoneme></w>
  <w role="n">
    <phoneme alphabet="pinyin" ph="gang3 bi4">港币
    </phoneme></w>
  <w role="m">
    <phoneme alphabet="pinyin" ph="yi2 wan4 kuai4">一万块
    </phoneme></w>
  <w role="y"><phoneme alphabet="pinyin" ph="ne0">呢
    </phoneme></w>
  <w role="wt">！ </w>……
</s>
```

In SSML, the predefined value for "alphabet" attribute of "<phoneme>" element is "ipa" for International Phonetic Alphabet [10]. However it is very complicated and difficult to lean and to understand. SSML also allows defining vendor-specific values for "alphabet" in the Pronunciation Alphabet Registry [7]. According to the Chinese Putonghua Romanization scheme [11] and the LSHK Cantonese Romanization scheme [12], we propose to use "pinyin" for representing Pinyin for Putonghua (e.g. "pin1 yin1" for "拼音") and "jyutping" for representing Jyutping for Cantonese (e.g. "jyut6 ping3" for "粤拼") while conforming to the SSML specification.

3.5 Prosody analysis

The function of this module is to generate prosodic structures (boundaries for prosodic word and phrase), and to predict target prosodic information (pitch, duration, emphasis) for a certain word.

Prosodic structures have been playing important roles in speech communication. Prosodic word and prosodic phrase are the most two important grouping levels for producing synthetic speech with high intelligibility and naturalness. There is minor break (short duration) at prosodic word boundary; and major break (longer duration than prosodic word, but shorter than sentence) at prosodic phrase boundary [13].

SSML specification does not provide direct mechanism and elements for representing prosodic structures, but offers element "<break>" to control pausing or related prosodic boundaries between different tokens. Hence, we utilize "<break>" element with different "strength" attribute values for controlling different boundaries. Table 8 illustrates the mapping relationship between pre-defined values of "strength" attribute and related prosodic boundary types, where "syllable" corresponds to Chinese characters, "sub-sentence" relates to minor-sentences separated by comma, colon, etc. within a sentence marked by "<s>" element.

Human speeches carry not only verbal, propositional information about facts, but also non-verbal, communicative information carrying the intention, emotion, emphasis, etc. of the speaker. SSML specification provides two elements including "<prosody>" and "<emphasis>" for controlling the prosody and style of the synthetic speech so as to make it more natural and even more expressive (Table 9).

Table 8 Mapping relationship between pre-defined values of "strength" attribute and prosodic boundary types

x-weak	weak	medium	strong	x-strong
syllable	prosodic word	prosodic phrase	sub-sentence	sentence

3.6 Waveform production

The function of this module is to select appropriate speech units from speech corpus for concatenative TTS synthesis, to perform prosody modification to match the target prosodic information predicted in prosody analysis module, and finally, to concatenate speech units to generate the final speech output.

SSML does not provide detailed specification for this external module. We have proposed an extension to SSML so that unit selection results and even internal debugging information can be recorded in the SSML output of the internal unit selection module. As illustrated in Table 10, we have added two attributes to "<w>" element. The "id" attribute indicates the index identity of the selected speech unit in speech corpus, which is to be used in "prosody modification" and "waveform production" modules to retrieve wave data of the unit from speech corpus. The "src" attribute records the original source file (e.g. wave file) from which the speech unit comes. This attribute is provided for tracking speech unit back to original file for debugging and evaluation purpose use, and providing readable and traceable information for quality assurance of TTS engine.

Table 9 SSML output of prosody analysis module

```
<s> ……
  <break strength="strong" />
  <w role="r"><phoneme alphabet="pinyin" ph="ni3">你
    </phoneme></w>
  <w role="d"><phoneme alphabet="pinyin" ph="hai2">还
    </phoneme></w>
  <break strength="weak" />
  <w role="v"><phoneme alphabet="pinyin" ph="qian4">欠
    </phoneme></w>
  <w role="r"><phoneme alphabet="pinyin" ph="wo3">我
    </phoneme></w>
  <break strength="medium" />
  <w role="n">
    <phoneme alphabet="pinyin" ph="gang3 bi4">港币
    </phoneme></w>
  <break strength="weak" />
  <w role="m">
    <emphasis level="strong">
    <phoneme alphabet="pinyin" ph="yi2 wan4 kuai4">一万块
    </phoneme></emphasis></w>
  <w role="y"><phoneme alphabet="pinyin" ph="ne0">呢
    </phoneme></w>
  <w role="wt">！</w>……
</s>
```

Table 10 SSML output of unit selection module with trace back information for quality assurance of TTS engine

```
<s> ……
  <w role="n"
    id="gang3:3 bi4:10" src="pth32 pth43">
    <phoneme alphabet="x-pinyin" ph="gang3 bi4">港币
  </phoneme></w>
  <break strength="weak" />
  <w role="m"
    id="yi2:5 wan4:23 kuai4:15" src="pth23 pth60 pth33">
    <emphasis level="strong">
      <phoneme alphabet="x-pinyin" ph="yi2 wan4 kuai4">
      一万块
  </phoneme></emphasis></w>……
</s>
```

4 Conclusions and Future Work

This paper reports our recent work on designing a unified framework for a multilingual TTS synthesis engine – Crystal. The unified framework aims to define the basic common TTS modules for different languages and/or dialects. The framework also defines the standard interface between consecutive TTS modules using SSML specification for better standardization, interoperability, multilinguality, and extensibility. The detailed module divisions and implementation technologies for the unified framework are introduced. The possibility to extend SSML specification for assisting the algorithm research and evaluation of TTS engine is also proposed.

The proposed unified framework has been focused on audio modality only. We are planning to extend this work to support visual modality and finally propose a unified framework for text-to-audio-visual speech (TTAVS) synthesis (i.e. Talking Avatar).

References

[1] Campbell N, Black A. Prosody and the selection of source units for concatenative synthesis. Progress in Speech Synthesis. Berlin, Germany: Springer-Verlag, 1996: 279-282.

[2] Yi J, Glass J. Natural sounding speech synthesis using variable-length units. In: Proceedings of the ICSLP. Australia, 1998: 1167-1170.

[3] Chu M, Peng H, Yang H, et al. Selecting non-uniform units from a very large corpus for concatenative speech synthesizer. In: Proceedings of the ICASSP. USA, 2001: 785-788.

[4] Xu J, Huang D, Wang Y, et al. Hierarchical non-uniform unit selection based on prosodic structure. In: Proceedings of the InterSpeech. Belgium, 2007: 2861-2864.

[5] Meng H, Keung C, Siu K, et al. CU Vocal: Corpus-based syllable concatenation for Chinese speech synthesis across domains and dialects. In: Proceedings of the ICSLP. USA, 2002: 2373-2376.

[6] Speech Synthesis Markup Language version 1.0, W3C recommendation. http://www.w3.org/TR/speech-synthesis/.

[7] Speech Synthesis Markup Language version 1.1, W3C working draft. http://www.w3.org/TR/speech-synthesis11/.

[8] Yang H, Meng H, Wu Z, Cai L. Modeling the global acoustic correlates of expressivity for Chinese text-to-speech synthesis. In: Proceedings of the IEEE SLT. Aruba, 2006: 138-141.

[9] Unicode encoding standard version 4.1: ISO 10646:2003. http://en.wikipedia.org/wiki/ISO_10646.

[10] International Phonetic Association's organization website. http://www.arts.gla.ac.uk/ipa/ipa.html.

[11] GB/T 16159-1996. Basic rules for Hanyu Pinyin orthography: national standard of P.R. China.

[12] The Linguistic Society of Hong Kong (LSHK) Cantonese Romanization scheme. http://www.iso10646hk.net/jp/index.jsp.

[13] Tseng C, Pin S, Lee Y, et al. Fluent speech prosody: framework and modeling. *Speech Communication*, 2005, 46(3-4): 284-30.