An Introduction of S3ML

CHEN Ming, LV Shinan, Li Xiulin
Beijing InfoQuick SinoVoice Speech Technology Corp.
{chenming, lixiulin} @sinovoice.com.cn

1. Background

Found in October 2000, Beijing InfoQuick SinoVoice Speech Technology Corp. (SinoVoice) is now a leading speech technology and service provider in China. SinoVoice has successfully deployed more than 1,000 real systems based on jTTS, the leading Chinese TTS software developed by SinoVoice with own intellect propriety.

SSML (Speech Synthesis Markup Language) is designed to provide a rich, XML-based markup language for assisting the generation of synthetic speech in Web and other applications. 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. SSML is a W3C recommendation and the SSML specification and more information can be found in http://www.w3.org/TR/speech-synthesis/。

Conformed to SSML, SinoVoice defined S3ML (SinoVoice Speech Synthesis Markup Language) since the launching of jTTS 4.0 in March, 2004. S3ML is based on SSML specification, but defines the details of some tags which SSML does not define precisely. At the same time, S3ML defines some extension aiming at Chinese TTS. So we want to introduce S3ML in W3C Workshop on Internationalizing the SSML, and propose some suggestions.

2. PinYin Support

PinYin is a phoneme annotation method for Chinese characters. In S3ML, <phoneme> tag is used to support Chinese PinYin.

In SSML, the <phoneme> tag is defined as following: *The **phoneme** element provides a phonemic/phonetic pronunciation for the contained text. The **ph** attribute is a required attribute that specifies the phoneme/phone string. The **alphabet** attribute is an optional attribute that specifies the phonemic/phonetic alphabet.*

So we use “py” as **alphabet** attribute if the text contained by **ph** attribute is a PinYin string. The description of PinYin should be conformed to “Chinese Mandarin PinYin Specification“. We can use a series of PinYin, and each PinYin need tone information. 1 to 4 means high flat, rising, diving and falling tone respectively，0 or 5 means light tone.

Here is an example:

```
<?xml version="1.0" encoding="GB2312"?>
<speak version="1.0" xml:lang="zh-cn">
      <phoneme alphabet="py" ph="zha1">查</phoneme>良镛
      <phoneme alphabet="py" ph="zha1 liang2yong1">查良镛</phoneme>先生
</speak>
```
(查良镛 is name of a famous Chinese writer. 查 is a polyphone character, which will be read as ‘zha1’ when it is a surname and will be read as ‘cha2’ when it means search)

Although TTS system will use specified phoneme series to do synthesis, and ignore the actual text, we still suggest to put phoneme series in **ph** attribute, and to put the readable text in the tag. Then it can also be understood if the SSML document is not used by synthesis system.

Furthermore, there is the possibility that the PinYin string is included in the normal text. In such case, we can use **<say-as>** tag with the attribute which value is “**phoneme**”. Then system should read the text contained by **<say-as>** tag as PinYin series. It’s also an extension of S3ML. For example:

```
Next station is <say-as interpret-as="phoneme" format="py">di4 tan2</say-as>
```

In CSSML defined by iFlyTek, <phoneme> tag is used to support PinYin, but the PinYin is placed in the normal text and they use a new defined **lang** attribute to indicate it is a PinYin string. For example:

```
<?xml version="1.0" encoding="GB2312"?>
<speak xml:lang="cn">
            <phoneme lang="zh-cn">zha1</phoneme>良镛
</speak>
```

Comparing with this method, we think S3ML suggestion is more compatible with the SSML specification.

3. <say-as> Definition

The detail attribute of <say-as> tag is not defined in SSML. *Defining a comprehensive set of text format types is difficult because of the variety of languages that have to be considered and because of the innate flexibility of written languages. SSML only specifies the* ***say-as*** *element, its attributes, and their purpose. It does not enumerate the possible values for the attributes.*

For the aim of practical usage, we define the detail ‘interpret-as’ and ‘format’ attributes of <say-as> tag in S3ML. The specification is as following:

interpret-as	**Format**	**Interpretation**	**Examples**
Letters		interpret the content as letters	<say-as interpret-as="letters">IBM</say-as>: *i bee am*
Words		interpret the content as words	<say-as interpret-as=”words”>ASCII </say-as>: *askie*
Number		Automatic decided	
	Cardinal	a cardinal number	<say-as interpret-as="number" format="cardinal">Ⅶ</say-as> : *seven* <say-as interpret-as="number" format="cardinal"> 123 < /say-as> : *one hundred and twenty three*
	Ordinal	an ordinal number	<say-as interpret-as="number" format="ordinal">5</say-as> : *fifth*
	telegram digits	Read digits one by one	<say-as interpret-as="number" format="telegram"> 123 </say-as> : *one two three*
	Score	A score	The score is <say-as interpret-as="number" format="score"> 3:1</say-as>：*three vs. one*
	Fraction	A fraction	get <say-as interpret-as="number" format="fraction">1/3</say-as>：*one third*
	Telephone	A telephone number	<say-as interpret-as="number" format="telephone">123-456-7890</say-as>
date		Automatic decided	
	ymd	a date in year-month-day format	Today is <say-as interpret-as="date" format="ymd">2003/7/3</say-as>
	mdy	a date in month-day-year format	Today is <say-as interpret-as="date" format="mdy">7/3/2003</say-as>
	dmy	a date in day-month -year format	Today is <say-as interpret-as="date" format="dmy">7/3/2003</say-as>
	ym	a date in year-month format	It was taken place in <say-as interpret-as="date" format="ym">2003/7</say-as>
	my	a date in year-month format	It was taken place in <say-as interpret-as="date" format="my">7/97</say-as>
	md	a date in year-month format	Today is <say-as interpret-as="date" format="md">12/1</say-as>

	dm	a date in year-month format	Today is <say-as interpret-as="date" format="dm">7/3</say-as>
	y	a date in year-month format	<say-as interpret-as="date" format="y">2008</say-as> Olympic Games
time	hms	a time in hour-minute-second format	The meeting will be started at <say-as interpret-as="time" format="hms">14:30:09</say-as>
	hm	a time in hour-minute format	The meeting will be started at <say-as interpret-as="time" format="hm">14:30</say-as>
	h	a time in hour format	The meeting will be started at <say-as interpret-as="time" format="h">14</say-as>
duration	hms	a duration in hour-minute-second format	The record is <say-as interpret-as=" duration " format="hms">14:30:09</say-as>
	hm	a duration in hour-minute format	The record is <say-as interpret-as=" duration " format="hm">14:30</say-as>
	ms	a duration in minute-second format	The record is <say-as interpret-as=" duration " format="hm">14’30</say-as>
	h	a duration in hour format	Duration: <say-as interpret-as=" duration " format="hm">14</say-as>: *14 hours*
	m	a duration in minute format	Duration: <say-as interpret-as=" duration " format="m">14</say-as>: *14 minutes*
	s	a duration in second format	Duration: <say-as interpret-as=" duration " format="s">14</say-as>: *14 seconds*
	msms	a duration in minute-second-millisecond	The record is<say-as interpret-as=" duration " format="msms">14'30"12</say-as>
	sms	a duration in second-millsecond format	The record is<say-as interpret-as=" duration " format="sms">14"30</say-as>
phoneme	ipa	International Phonetic Alphabet	The pronunciation of ‘tomato’ is <say-as interpret-as="phoneme" format="ipa">tɒmɑtoʊ</say-as>
	py	PinYin	Next station is <say-as interpret-as="phoneme" format="py">di4 tan2</say-as>
name		Automatic decided	

	person	Person name	<say-as interpret-as="name" format="person">Michael</say-as>
	company	Company name	<say-as interpret-as="name" format="company">SinoVoice</say-as>
address		Postal address	<say-as interpret-as="address" > #12, ShangDi Information Road</say-as>
math		Math expression	The expression is <say-as interpret-as="math">a+486-1014-30 2003/12/1*(-10291)</say-as>
net	email	Email address	<say-as interpret-as="net" format="email">abc@xyz.com</say-as>
	url uri	URL	<say-as interpret-as="net" format="uri"> http://www.sinovoice.com.cn </say-as>

4. Domain support

The customized TTS is used more and more popular in real systems, because it can do a lot of optimization works according to the text limited in a specific domain and then get better voice quality than standard version. But we can not find a proper method to support reading text in a specific domain in SSML. The only possibility is to define a new voice name for each domain and then use **<voice>** tag and **name** attribute. But we think it is better to define a new tag or new attribute because it is normally to support several different domains by using a same voice library.

In S3ML, **<domain>** tag is defined for this purpose. The synthesis engine will use customized TTS to read the text included in <domain> tag according to name attribute of this tag if there is the customized package. The **name** attribute will use a vendor-specific name to indicate which domain the text belongs to. **<domain>** tag will not change voice, so if a voice library does not have this customized package, the tag will be just ignored.

If we want the system to select a voice which can have best support of this domain, we can use an extended **'domain'** attribute of **<voice>** tag.

For example:

```
<?xml version="1.0" encoding="GB2312"?>
<speak version="1.0" xml:lang="zh-cn">
        <domain name="weather"> Today, mostly sunny. Highs 65 to 75. </domain>。
        <voice domain="weather"> Tonight, partly cloudy. </voice>。
</speak>
```

5. Conclusion

In this paper, we proposed some extensions to SSML according to S3ML specification. We hope it will be helpful to define the standard for internationalizing SSML in the future.