Using SSML for Indian Languages Text to Speech Synthesis
Position paper for SSML workshop.
Submitted by: Vibhu Agarwal

Introduction: The objective of SSML is to allow users of speech synthesis systems to control almost all aspects of speech synthesis. SSML is a XML based markup language that allows such control through a number of tags and accompanying attributes that are used to suitably annotate text. Text formatted according to the SSML version 1.0 specification when presented to a SSML compliant synthesis engine is thus accompanied by control information that the engine uses to suitably adjust synthesis parameters at run time.

TTS developers are all too familiar with the extent to which annotation can impact synthesis output. One only needs to briefly consider the speech synthesis methodology in order to fully appreciate the importance of annotation in controlling synthesis. Text pre-processors attempt to normalize, expand, punctuate and annotate text aggregates in order to help the core synthesizer produce speech output that is meaningful and natural. The nature of pre-processing is different for different levels of aggregation. For instance, the string of phonetically transcribed words in a sentence will need to be modified in order to account for continuous speech. A sentence or phrase may need to be assigned prosodic rules. Similarly, at the word level grapheme to phoneme conversion for words may occasionally require explicit phonetic annotation or reference to special purpose lexicons. SSML allows such annotations to be made in a uniform manner, thus enhancing application sophistication and encouraging a vendor-agnostic application development.

Text to speech synthesis in Indian languages: Prologix is one of the earliest developers of speech synthesis technology for Indian languages. Its TTS engine ‘Vaachak’ is presently the only TTS initiative for Indian languages that is being used extensively by Indian telecom companies, IVR developers, web page readers and screen readers for visually disabled people. Vaachak is available with two voices ‘Rakhi’ (Hindi) and ‘Rupali’(Indian accented English). In the process of working with Indian speech application developers, Prologix has gleaned some interesting insights regarding the various synthesis controls commonly required at run time while rendering text from common sources into speech. We feel that some of these experiences may be useful in guiding the evolution of the SSML specification so that it is of greater relevance to Indian speech application developers.

There are 18 official languages that are recognized by the Indian constitution. The actual number of dialects is many times this number. Whereas these languages have significant differences in their phone sets, there are some common shared characteristics.

1. They are generally phonetic
2. Common usage involves a liberal use of words loaned from other languages, including English.
3. Text is often encoded using non-standard encoding schemes. Very often encoding schemes vary across fonts.
4. Since the development of content and user interfaces in native language has lagged behind somewhat, much of electronic content in Indian languages is actually stored as ASCII in a Romanized representation. Again, there are multiple native-Roman transliteration schemes in use.

Challenges in rendering Indian texts: Because of characteristics 2-4 listed above, rendering free form Indian language text poses several challenges. The use of loan words from English is extremely common in all Indian languages. There is a proliferation of font families that use

their own encoding. Data that is available as transliterated Roman form could confirm to one of the many popular transliteration schemes. It is also difficult to automatically distinguish Roman transliterated text from English text. Text formatted with SSML elements could help solve these problems.

Suggestions:

1. Domain specific <break> and <emphasis> element: It would be useful to have a mechanism to indicate a particular kind of <break> or <emphasis> to the speech synthesizer. For instance

   Hello this is <break domain="name" > Amit Kumar Sharma<break> how are you ?

   could be used to indicate

   Hello this is [pause] Amit [pause] Kumar [pause] Mallik [pause] how are you ?

   And

   <break domain="interlanguage"> इस software की performance बेकार है<break/>

   could be used to indicate

   इस [pause] software [pause] की [pause] performance [pause] बेकार है

2. Roman transliterated text can be handled by indicating the transliteration scheme to the TTS. A transliteration scheme is simply an ordered map that indicates equivalence of characters and character compounds between two different scripts. By providing the URL of the map, for a given transliterated text, any transliteration scheme can be used easily. For instance

   <transliteration format="myscheme" url=www.myschemeurl.map> mera nam vachak hai </transliteration>

   could be used to indicate

   मेरा नाम वाचक है

3. Similarly appropriate element and attributes can be introduced to indicate non-standard encoding and a corresponding encoding converter available at a URL