Multilayered Extensions to the Speech Synthesis Markup Language for Describing Expressiveness

E. Eide, R. Bakis, W. Hamza, and J. Pitrelli

IBM T.J. Watson Research Center, P.O. Box 218, Yorktown Heights, NY 10598 U.S.A.
{eeide,bakis,hamzaw,pitrelli}@us.ibm.com

Abstract

In this paper we discuss possible extensions to the Speech Synthesis Markup Language (SSML) to facilitate the generation of synthetic expressive speech. The proposed extensions are hierarchical in nature, allowing specification in terms of physical parameters such as instantaneous pitch, higher-level parameters such as ToBI labels, or abstract concepts such as emotions. Low-level tags tend to change their values frequently, even within a word, while the more abstract tags generally apply to whole words, sentences or paragraphs. We envision interfaces at different levels to serve different types of users; speech experts may want to use low-level interfaces while artists may prefer to interface with the TTS system at more abstract levels.

1. Introduction

Expressive speech synthesis enables efficient interaction. Consider the following interaction from a travel-planning system:

CUSTOMER: I’d like a flight from New York to Denver tomorrow morning.
SYSTEM: I have a flight from New York to Denver tomorrow afternoon.

Without an ability to express the contrast between the requested morning flight and the provided afternoon flight, the customer is likely to attribute erroneously the mismatch to the system misunderstanding his request, and follow with a repeat request. If, instead, the system has an ability to specify to the synthesis component that afternoon be spoken in such a way as to express contrast, the customer is more likely to understand that the system has understood him and is providing the best alternative, thereby sparing the diversion and avoiding a lengthier interaction.

Beside efficiency of interaction, another need for expressive synthesis is appropriateness of the expression. Generally, an authoritative-sounding voice may be best for an information-provision application such as the travel-planning system. However, in a case in which the system does not understand the customer’s request or cannot comply with it, a less authoritative expression would be more appropriate, to complement rather than conflict with the words expressing confusion about the request or remorse about the system’s inability to comply with the request. In such a case, the appropriateness of the expression is likely to factor into the customer’s satisfaction with the quality of the system; efficiency may be an issue here as well, as the system may impose less cognitive load on the listener if the expression coordinates with the text of the message rather than conflicts with it.

Although no markup language, including SSML [1], is currently rich enough to facilitate expressive TTS, SSML provides a suitable framework which may be easily extended to support tags useful for generating expressive synthetic speech. The remainder of the paper is devoted to a proposed extension to SSML which will allow for a more complete and suitable interface to expressive TTS systems.

2. The Need for a Hierarchical Interface

Independent of the approach taken by a synthesizer to generate expressive speech, conveying the desired expressive state to the synthesizer is necessary for efficient and appropriate speech-based interactions between man and machine. For example, in Cahn’s early work [2], an emotion to be synthesized is translated into a sequence of low-level controls for a formant synthesizer. In a unit-selection-based synthesizer, one could collect large databases spoken in different expressive states in order to generate synthetic speech with an emotional content [3], [4], or one could use signal processing to adapt neutral speech to achieve a desired emotion. Either way, though, the desired expressive state needs to be specified to the synthesis system. We choose a hierarchical interface to the TTS system, in which characteristics that can be defined, measured, or observed on short signal segments tend to belong to levels which are low in the hierarchy, while longer-term effects are typically assigned to higher levels. Thus, “instantaneous pitch” sits near the bottom of the hierarchy, while “gradual build-up of dramatic tension,” would reside in a higher level.

Our model adopts a reductionist philosophy, in which high-level effects may be realized in terms of lower levels. For example, an increase in tension might be conveyed through a gradual rise in base pitch, an increase in breathing rate, etc. The focus of this paper is not to present a specific list of all effects at all levels, or even a list of the levels themselves. Rather, we discuss methods by which such a structure should be established and we describe possible software architectures, with a few examples for clarity.

Consider, for example, the concept of emphasis, which is clearly at least one level above elementary signal characteristics like pitch and loudness. Common wisdom holds that emphasis can be reduced to pitch, loudness, and duration. Each of these, however, is a complex phenomenon in its own right. Pitch and loudness, as functions of time, can vary in many different ways even within a single word. Duration changes may affect individual parts of a word in different ways; there may be changes in glottal wave shape or breathiness. Speech, when represented by such low-level attributes, requires a multi-dimensional description which is difficult to interpret in terms of percept. How, then, do we determine the mapping from a concept like “emphasis” to the lower-level manifestations of it? Among the possibilities are:
Handwritten rules.
Artistic judgement.
Statistically-trained methods.

Artistic judgement is really a special case of handwritten rules, although in this case, rather than a linguist relying on knowledge built up by studying a number of examples, we envision a drama coach or sound designer fine-tuning parameters of the system to realize a specific effect for a particular waveform. Indeed, as TTS becomes an important communications medium, it will benefit from the same kinds of artistic expertise as the older modalities of radio, theater, and film.

As there are well-established advantages and disadvantages of using rules vs. statistical models, the challenge for TTS researchers is to devise a user interface that accommodates both the intuitive approach of the artist and the analytic methods of the speech scientist.

Currently, SSML accepts two kinds of specification, numerical:

\[
\text{\texttt{<prosody contour="(0\%,+20)(45\%,-30\%)">}} \\
\text{\texttt{hello}} \quad \text{\texttt{</prosody>}}
\]

and descriptive:

\[
\text{\texttt{<prosody pitch="high">}} \\
\text{\texttt{hello}} \quad \text{\texttt{</prosody>}}
\]

In the above numerical example, the pitch is specified to be 20 Hz higher than the previously specified value, starting at the beginning of the following phrase, with an increase of 30%, occurring 45% of the way through the utterance, whereas in the descriptive example, the pitch is simply specified to be high.

Low levels of the specification hierarchy lend themselves well to numerical specification. Fundamental frequency, for example, is a one-dimensional variable that is objective and measurable. At higher levels, such as “enthusiasm,” where time span and dimensionality increase, descriptive terms are more useful than lengthy numerical specifications.

3. Proposed Extensions to SSML

3.1. Hierarchical Structure

A hierarchical structure for interfacing marked-up text into an expressive TTS system is proposed in Figure 1. The interface is layered in nature; tags explicitly controlling physical parameters reside in the lowest layer, with increasingly abstract tags belonging to higher layers. The upper end of the layered structure is open to accommodate additional, scenario-level markup layers. The figure shows examples of layers belonging to different levels in the hierarchical structure. At the lowest level, tags such as pitch contour specification, duration, and voice quality information are present. Most tags in the current SSML reside in this layer. Note that it would be difficult for a human application developer interfacing with a text-to-speech system to translate mentally from low-level tags to the auditory impression which they would evoke if those tags were supplied, and it would be even more difficult for the developer to guess an appropriate set of low-level tags to produce a particular expressive effect.

Moving up a bit in the structure, layers start to contain information which has some linguistic meaning, while still relating to acoustic properties in the synthesized speech, such as Tones and Break Indices (ToBI) [5],[6] tags. Continuing up to mid-level layers, tags begin to carry more linguistic and expressive information, such as word emphasis and contrastive stress. At this level of abstraction it becomes easier for the developer to provide the tags in order to instill a particular quality in the output speech. Slightly higher levels are associated with speaker states such as expressing urgency and conveying uncertainty. The highest levels of the markup structure are appropriate for describing quite abstract information such as emotions and scenarios.

The left-hand side of the figure shows interfaces between the multilayered structure and users of the expressive text-to-speech system. These interface boxes enable developers with different areas of expertise to interact with the system in the manner that is most natural and suitable for them. For example, a natural-language-generation developer could interface with the mid-level markup layer of the figure, since that developer would know the expressive states used in a given dialog system. The states such as “showing contrast” or “asking a question” could be passed to the expressive TTS system directly at this level. A movie director, on the other hand, would have a concept of the emotion he is trying to portray, and would most suitably interface with the system at a high level of abstraction. Furthermore, the interfaces themselves could vary; anything from assembly code to a GUI could be used to interface with the system.

Tags belonging to each layer are generated by a human (possibly computer-aided), generated by another computer system such as a dialog manager, or are translations of tags belonging to another, typically higher, layer in the markup structure.

3.2. During Synthesis

The multilayered framework for specifying expressiveness creates a rich, annotated text to be used by the synthesizer. The expressive speech synthesizer deals with tags or tag layers using one of the following three alternatives:

1. Use the tags directly in the speech synthesis process.
2. Translate tags from one layer to tags in a lower-level layer using tag translators.
3. Ignore tags not supported.

Option 1 allows the synthesizer to use various layers of tags to provide better-quality speech output. This is demonstrated in the figure by arrows going from all layers to the synthesizer. In the case of a concatenative synthesizer having a set of expressive databases available, say, one for each of a set of desired emotions to be synthesizable, the horizontal arrows which directly pass the high-level tag to the synthesizer would be appropriate. Option 2 enables the design of new tag-layers along with their user-interfaces, and the use of those new layers with legacy synthesizers by developing appropriate translators to translate new tags to tags belonging to layers understood by the synthesizer. In the case where expressive speech is achieved via signal processing on an e.g. neutral database, vertical arrows would be invoked to transform abstract specifications to quantitative ones. Option 3 allows extension to the structure while preserving backward compatibility with a legacy synthesizer.

Note the proposed extensions may well exacerbate the problem of conflicting markup, which already exists in the current SSML; it will be the responsibility of the synthesizer interpreting the markup to resolve such conflicts.

\[1\]The word translator here is defined to be the set of rules or systems that map tags in one layer to corresponding tags in another.
3.3. Extensibility

The process of adding a new layer consists of:
- Defining the layers’ tags and their meanings.
- Defining an interface appropriate to edit those tags.
- Developing translators to other layers understood by the synthesizer and/or using those tags in the synthesizer internally.

For example, consider the process of introducing a layer that contains ToBI labels as tags. An interface program would enable a linguist to insert ToBI labels in the input. The synthesizer could use the ToBI labels directly to produce the output speech, or a translator could be developed to map those tags into lower level tags such as duration and pitch which are understood by legacy synthesizers. Even if the set of tags in one layer is translated to other tags in another, both layers could be used by the synthesizer. In the case of ToBI labels, the labels themselves as well as pitch and duration information derived from the labels could be used by a unit-selection-based synthesizer.

It is not necessary, however, that there exist a mapping from tags in one layer to tags in a lower level layer. As a matter of fact some layers might be independent of each other. For example, specifying the “sad” or “happy” tag in a high-level layer is largely independent of the “emphasis” tag in a middle layer.

3.4. Examples

Table 1 illustrates some specific attributes and their values which we propose, in order to control the pitch, speaking rate, loudness, emotion, attitude, and scenario under which the speech is requested. All attributes are optional.

In addition to the existing SSML attribute “volume” which is defined as a linear increase in amplitude, we propose expanding the loudness concept to encompass proximity to microphone and degree of shouting by adding the attributes “proximity” and “shouting.” Additional attributes to be included in the extended SSML are breathiness, creakiness, and articulation. We also propose the inclusion of ToBI attributes to the prosody element. The syntax for ToBI, would be, for example:

\[
<\text{prosody tobi=H* 2} />
\]

which would indicate that the word following the markup would have an “H*” pitch accent on the lexically-stressed syllable, and that word would be followed by a break index of 2. Since the set of labels for pitch accents, phrase accents, boundary tones, and break indices are mutually exclusive, all of the values may be specified together, or any of them may be excluded, without ambiguity.

In the following example [7], we envision the following marked-up text, making use of the proposed extensions, being supplied to an expressive TTS system.

\[
<\text{prosody emotion=calm} \\
\text{attitude=confident}> \\
\text{Yes sir, the package will be on your desk tomorrow. And I say that with the } <\text{prosody tobi=L+H**}/> \text{ utmost confidence.} \\
\text{<emphasis> I </emphasis> will take <emphasis> care </emphasis> of it.} \\
\text{</prosody>}
\]
As the level of abstraction in attributes increases, the difficulty in designing the mapping from higher to lower levels increases, so that translating a scenario into low-level attributes becomes increasingly complex. We hope that, once mappings for the different phenomena to low-level parameters.

The proposed extension is hierarchical in nature, allowing for interfaces at various levels of abstraction, for use by experts in different facets of speech production. In the proposed extensions, we preserve SSML’s concept of specification being either numerical or descriptive. A third method of specification, not currently present in SSML and not yet feasible, but one which should be kept in mind for the future, is specification in terms of a situation. For example, a director might describe a scene to an actor as “glum, despair, sad, angry, ...” with the specification “x-high, high, medium, low, x-low, default.”

TTS systems of the future need to be able to handle these complex situational specifications, at least in a limited way. The support for both numerical and descriptive specification currently implemented in SSML leads to a natural extension for a hierarchical description of speech. Defining and implementing a markup language to assist TTS Designers will call for extension of the current terminology, overcoming difficult technical challenges, and reconciling conflicting terminologies and world views between artistic, psychological and technical communities. This paper is intended as a step in that direction.

### 5. Acknowledgements

Thanks to Andy Aaron of IBM for enlightening discussions on how TTS Designers would prefer to interface with a synthesis system.

### 6. References


