THE USE AND POTENTIAL OF EXTENSIBLE MARK-UP (XML) IN SPEECH GENERATION

M. Huckvale

Phonetics & Linguistics
University College London
Phone: +44 20 7679 7401; Fax: +44 20 7383 0752
Email: M.Huckvale@ucl.ac.uk

ABSTRACT

Keywords: XML, Mark-up, Speech Synthesis, System Architecture, Annotation. This article discusses the importance of the extensible mark-up language (XML) to the development of current and future speech synthesis systems. XML is an emerging standard of textual mark-up which is well suited to efficient computer manipulation. It provides a means for representing and processing the complex linguistic structures used in synthesis in an open and non-proprietary manner. Currently XML is being used to mark-up text for input into text-to-speech systems, for the annotation of corpora, and for the implementation of speech driven applications. However recent research has shown how XML can be used to create an open architecture for synthesis through the mark-up of working data structures and through the creation of a language for manipulating XML data. The article concludes by suggesting that work towards standards for marking up the information structure of text and its discourse function with XML would help create a new generation of intelligent sounding computer voices.

1. INTRODUCTION

The Extensible Markup Language (XML) is a simple dialect of Standard Generalised Markup Language (SGML) designed to facilitate the communication and processing of textual data on the Web in more advanced ways than is possible with the existing Hypertext Markup Language (HTML). XML goes beyond HTML in that it attempts to describe the *content* of documents rather than their *form*. It does this by allowing authors to design markup that is specific to a particular application, to publish the specification for that markup, and to ensure that documents created for that application conform to that markup. Information may then be published in an open and standard form that can be readily processed by many different computer applications.

XML is a standard proposed by the World Wide Web Consortium (W3C). W3C sees XML as a means of encouraging: "vendor-neutral data exchange, media-independent publishing, collaborative authoring, the processing of documents by intelligent agents and other metadata applications" [1].

XML is a dialect of SGML specifically designed for computer processing. XML documents can include a formal syntactic description of their markup, called a Document Type Definition (DTD), which allows a degree of content validation. However the essential structure of an XML document can be extracted even if no DTD is provided. XML markup is hierarchical and recursive, so that complex data structures can be encoded. Parsers for XML are fairly easy to write, and there are a number of publicly available parsers and toolkits. An important aspect of XML is that it is designed to support Unicode representations of text so that all European and Asian languages as well as phonetic characters may be encoded.

Here is an example of an XML document:

```
<?xml version='1.0'?>
<!DOCTYPE LEXICON [
<!ELEMENT LEXICON (ENTRY)* >
<!ELEMENT ENTRY (HW, POSSEQ,
PRONSEQ) >
<!ELEMENT HW (#PCDATA) >
<!ELEMENT POSSEQ (POS)* >
<!ELEMENT POS (#PCDATA) >
<!ELEMENT PRONSEQ (PRON)* >
<!ELEMENT PRON (#PCDATA) >
<!ATTLIST ENTRY
      ID    ID     #REQUIRED>
<!ATTLIST POS
      PRN   CDATA #REQUIRED>
<!ATTLIST PRON
      ID    ID     #REQUIRED>
]>
<LEXICON>
```

[1] http://www.w3c.org/XML

```
<ENTRY ID="READ">
 <HW>read</HW>
 <POSSEQ>
  <POS PRN="#ID(READ-1)">
     V(past)</POS>
  <POS PRN="#ID(READ-2)">
     V(pres)</POS>
  <POS PRN="#ID(READ-2)">
     N(com,sing)</POS>
 </POSSEQ>
 <PRONSEQ>
  <PRON ID="READ-1">'red</PRON>
  <PRON ID="READ-2">'rid</PRON>
 </PRONSEQ>
</ENTRY>
...
</LEXICON>
```

In this example the heading '`<?xml ... ?>`' identifies an XML document in which the section from '`<!DOCTYPE LEXICON [`' to '`]>`' is the DTD for the data marked up between the `<LEXICON>` and `</LEXICON>` tags. This example shows how some of the complexity in a lexicon might be encoded. Each entry in the lexicon is bracketed by `<ENTRY>`; within this are a headword `<HW>`, a number of parts of speech `<POSSEQ>`, and a number of pronunciations `<PRONSEQ>`. Each part of speech section `<POS>` gives a grammatical class for one meaning of the word. The `<POS>` tag has an *attribute* `PRN`, which identifies the `ID` attribute of the relevant pronunciation `<PRN>`. The DTD provides a formal specification of the tags, their nesting, their attributes and their content.

XML is important for development work in speech synthesis at almost every level. XML is currently being used for marking up corpora, for marking up text to be input to text-to-speech systems, for marking up simple dialogue applications. But these are only the beginning of the possibilities: XML could also be used to open up the internals of synthesis-by-rule systems. This would give access to their working data structures and create open architectures allowing the development of truly distributed and extensible systems. Joint efforts in the standardisation of mark-up, particularly at the higher linguistic levels, will usefully force us to address significant linguistic issues about how language is used to communicate.

Section 2 of this article describes some of the current uses of XML in speech generation and research, while section 3 discusses how XML has been used in the ProSynth project[2] to create an open synthesis architecture, and in the SOLE project[3] to encode textual information essential for effective prosody generation.

2. CURRENT USE OF XML IN SPEECH GENERATION

2.1 Mark-up for Spoken Language Corpora

The majority of spoken language corpora available today are distributed in the form of binary files containing audio and text files containing orthographic transcription with no specific or standardised markup. This reflects the concentration of effort in speech recognition on the mapping between the signal and the word sequence. It is significant that missing from such data is a description of the speaker, the environment, the goals of the communication or its information content. Speech recognition systems can not, on the whole, exploit prior information about such parameters in decoding the word sequence. On the other hand, speech synthesis systems must explicitly model speaker and environment characteristics, and adapt to different communication goals and content.

Two recent initiatives at improving the level of description of spoken corpora are the American Discourse Resource Initiative[4] and the Multi-level Annotation Tools Engineering project[5] (MATE). The latter project aims to propose a standard for the annotation of spoken dialogue covering levels of prosody, syntax, co-reference, dialogue acts and other communicative aspects, with an emphasis on interactions between levels. In this regard they have been working on a multi-level XML description [5] and a software workbench for annotation.

In the multi-level framework, the lowest level XML files label contiguous stretches of audio signals with units that represent phones or words, supported by units representing pauses, breath noises, lip-smacks, etc. The next level XML files group these into dialogue moves by each speaker. Tags in this second level link to one or more units in the lowest level file. Further levels can then be constructed, referring down to the dialogue moves, which might encode particular dialogue strategies. Such a multi-level structure allows correlations to be drawn between the highest level goals of the discourse and the moves, words and even the prosody used to achieve them.

[2] http://www.phon.ucl.ac.uk/project/prosynth.htm

[3] http://www.cstr.ed.ac.uk/projects/sole.html

[4] http://www.georgetown.edu/luperfoy/Discourse-Treebank/dri-home.html

[5] http://mate.nis.sdu.dk/

2.2 Mark-up of Text for Input to TTS

SABLE is an XML based markup scheme for text-to-speech synthesis, developed to address the need for a common text-to-speech (TTS) control paradigm[6]. SABLE provides a standard means for marking up text to be input to a TTS system to identify particular characteristics of the text, or of the required speaker, or the required realisation. SABLE is intended to supersede a number of earlier control languages, such as Microsoft SAPI, Apple Speech Manager, or the Java Speech Markup Language (JSML).

SABLE provides markup tags for Speaker Directives: for example: emphasis, break, pitch, rate, volume, pronunciation, language, or speaker type. It provides tags for text description: for example to identify times, dates, telephone numbers or other common formats; or to identify rows and columns in a table. It can also be extended for specific TTS engines and may be used to aid in synchronisation with other media.

Here is a simple example of SABLE:

```
<DIV TYPE="paragraph">
  New e-mail from
  <EMPH>Tom Jones</EMPH> regarding
  <PITCH BASE="high" RANGE="large">
    <RATE SPEED="-20%">
      latest album</RATE>
  </PITCH>.
</DIV>
<AUDIO SRC="beep.aiff"/>
```

In this example, the subject of an e-mail is emphasised by setting a higher base pitch, a larger pitch range and a slower rate. Information necessary to specify such a requirement would come from the e-mail reader application which has privileged access to the structure of the source data. The message is terminated by an audible beep.

2.3 Mark-up of Speech Driven Applications

SpeechML is an XML based language for building network based conversational applications[7]. Such applications interact with users by voice output-input in a manner analogous to how a web browser interacts with a user using screen and keyboard. SpeechML is supported by a voice-driven browser that exploits the recognition and synthesis technology of IBM ViaVoice products. SpeechML is not designed for general purpose dialogue systems, but can be used to build conversational applications that involve menu choices, form filling and TTS.

To construct a SpeechML application, pages of SpeechML marked text are processed by the voice browser which speaks prompts and accepts verbal responses restricted by menus or validated form fields. At the heart of SpeechML are the tags `<PAGE>`: which groups SpeechML elements like an HTML page; `<MENU>`: which presents a set of choices and target links; `<BODY>`: which identifies a chunk of text to be spoken; and `<FORM>`: which groups fields of information required from user. Output text can be marked up with JSML, and input responses can be constrained by a simple grammar.

Here is a simple example of SpeechML:

```
<BODY NEXT="#menu1">
  <JSML> Welcome to the IBM ViaVoice
    <EMP>Conversational
      Browser</EMP>.
  </JSML>
</BODY>
<MENU NAME="menu1">
  Please choose from the main menu.
  <CHOICE TARGET="e-mail">
    E-mail.</CHOICE>
  <CHOICE TARGET="news">
    News.</CHOICE>
  <CHOICE TARGET="nav">
    Navigation.</CHOICE>
  <CHOICE TARGET="mcform">
    Food ordering.</CHOICE>
  <CHOICE TARGET="weather">
    Weather information.</CHOICE>
</MENU>
```

In this example, the welcome message in the first `<BODY>` tag, is followed by the `<MENU>` called "menu1" which presents a list of choices to the user. If the user repeats back one of the prompts, the relevant page is loaded according to the `TARGET` attribute of the `<CHOICE>` tag.

3. POTENTIAL FOR XML IN SPEECH GENERATION

The emerging standards for mark up described above: the MATE project for corpora, the SABLE system for TTS and the SpeechML system for applications are important to the development of speech synthesis systems, but they do not address a number of significant issues. This section draws examples from the recent research projects to demonstrate how XML could help address the problems of proprietary synthesis architectures, knowledge representation, and inexpressive delivery.

[6] http://www.bell-labs.com/project/tts/sable.html
[7] http://www.alphaworks.ibm.com/formula/speechml

3.1 Opening up Synthesis Architectures

An important contribution to current research and development activities in speech synthesis has been made by open source initiatives such as Festival[8], and public domain resources such as MBROLA[9]. However even these systems retain proprietary data formats for working data structures, and use knowledge representation schemes closely tied to those structures. This means that Phoneticians and Linguists willing and able to contribute to better synthesis systems are presented with complex and arbitrary interfaces which require considerable investment to conquer.

An alternative is to provide open, non-proprietary textual representations of data structures at every level and stage of processing. In this way additional or alternative components may be easily added even if they are encoded in different computer languages and run on different machines. In the ProSynth project [1], XML is used to encode the external data structures at all levels and stages. Synthesis is a pipeline of processes that perform utterance composition and phonetic interpretation. These processes are constructed to take XML marked input, to modify structures and attributes, and to generate XML marked output. As well as the representation of the utterance undergoing interpretation, XML is also used to mark up the input text and the pronunciation lexicon. For output, the XML format is converted to proprietary formats for MBROLA, HLSyn (see [2]) or for prosody-manipulated natural speech.

Here is a fragment of working data structure from ProSynth:

```
<AG ACCENT="H*L" TYPE="NUCLEAR">
 <FOOT DUR="1" FPITCH="100"
IPITCH="140" LON="50" POF="30"
PON="23" STRENGTH="STRONG">
  <SYL DUR="1" FPOS="1" RFPOS="2"
RWPOS="2" STRENGTH="STRONG"
WEIGHT="HEAVY" WPOS="1"
WREF="WORD3">
   <ONSET DUR="1" STRENGTH="STRONG">
    <CNS AMBI="N" CNSCMP="N"
CNSGRV="N" CNT="Y" DUR="1"
INHDUR="0.125" MINDUR="0.08" NAS="N"
RHO="N" SON="N" STR="Y" VOCGRV="Y"
VOCHEIGHT="OPEN" VOCRND="N"
VOI="N">s</CNS>
   </ONSET>
   <RHYME CHECKED="Y" DUR="1"
STRENGTH="STRONG" VOI="N"
WEIGHT="HEAVY">
    <NUC CHECKED="Y" DUR="0.4896"
LONG="Y" STRENGTH="STRONG" VOI="N"
WEIGHT="HEAVY">
     <VOC DUR="1" GRV="Y"
HEIGHT="OPEN" INHDUR="0.11"
MINDUR="0.035" RND="N">A</VOC>
     <VOC DUR="1" GRV="Y"
HEIGHT="OPEN" INHDUR="0.11"
MINDUR="0.035" RND="N">A</VOC>
    </NUC>
    <CODA DUR="1" VOI="N">
     <CNS AMBI="N" CNSCMP="N"
CNSGRV="Y" CNT="N" DUR="0.85"
INHDUR="0.11" MINDUR="0.05" NAS="Y"
RHO="N" SON="Y" STR="N" VOCGRV="Y"
VOCHEIGHT="OPEN" VOCRND="N"
VOI="Y">m</CNS>
     <CNS AMBI="Y" CNSCMP="N"
CNSGRV="Y" CNT="N" DUR="0.85"
INHDUR="0.08" MINDUR="0.06" NAS="N"
RHO="N" SON="N" STR="N" VOCGRV="Y"
VOCHEIGHT="OPEN" VOCRND="N"
VOI="N">p</CNS>
    </CODA>
   </RHYME>
  </SYL>
```

This extract is the syllable 'samp' from the phrase "it's a sample". The phone transcription `/sAAmp/` is marked by `CNS` (consonant) and `VOC` (vocalic) nodes. These are included in `ONSET`, `NUC` (nucleus) and `CODA` nodes, which in turn form `RHYME` and `SYL` (syllable) constituents. The `SYL` nodes occur under `FOOT` nodes, and the `FOOT` under `AG` (accent group) nodes. Phonetic interpretation has set some attributes on the nodes to define the durations and fundamental frequency contour.

3.2 Declarative Knowledge Representation

A continuing difficulty in the creation of open architectures for speech synthesis is the interdependency of rules for transforming text to a realised phonetic transcription. Context sensitive rewrite rules formalisms are a particular problem: the output of one rule typically feeds many others in ways that make it difficult to know the effect of a change. Often a new rule or a change to the ordering of rules can break the system.

It is generally accepted that the weaknesses of rewrite rules can be overcome with a declarative formalism. With a declarative knowledge representation, a structure is enhanced and enriched rather than modified by matching rules. Changes to the structure are always performed in a reversible way, so that rule ordering is not an issue. In

[8] http://www.cstr.ed.ac.uk/projects/festival/
[9] http://tcts.fpms.ac.be/synthesis/mbrola.html

ProSynth, the context for phonetic interpretation is established by the metrical hierarchy extending within and above the syllable. Thus the realisation of a phone can depend on where in a syllable it occurs, where the syllable occurs in a foot, and where the foot occurs in an accent group or intonation phrase. Thus context is established hierarchically rather than left and right. Knowledge for phonetic interpretation is expressed as declarative rules which modify attributes stored in the working data structure which is externally represented as XML.

The language formalism for knowledge representation is called ProXML. Phonetic interpretation knowledge stored in ProXML is interpreted to translate one stream of XML into another in the synthesis pipeline. The ProXML language draws on elements of Cascading Style Sheets as well as the 'C' programming language (see [4] for more information).

Here is a simple example of ProXML:

```
/* Klatt Rule 9: Postvocalic context
of vowels */
NUC {
  node coda = ../RHYME/CODA;
  if (coda==nil)
    :DUR *= 1.2;
  else {
    node cns = coda/CNS;
    if ((cns:VOI=="Y")&&
        (cns:CNT=="Y")&&
        (cns:SON=="N"))
      :DUR *= 1.6;
    else if ((cns:VOI=="Y")&&
             (cns:CNT=="N")&&
             (cns:SON=="N"))
      :DUR *= 1.2;
    else if ((cns:VOI=="Y")&&
             (cns:NAS=="Y")&&
             (cns:SON=="Y"))
      :DUR *= 0.85;
    else if ((cns:VOI=="N")&&
             (cns:CNT=="N")&&
             (cns:SON=="N"))
      :DUR *= 0.7;
  }
}
```

This example, based on Klatt duration rule 9 [6], operates on all `NUC` (vowel nucleus) nodes. The relative duration of a vowel nucleus, `DUR`, is calculated from properties of the rhyme: in particular whether the coda is empty, has a voiced fricative, a voiced stop, a nasal or a voiceless stop. The statement `':DUR *= 0.7'` means adjust the current value of the `DUR` attribute (of the `NUC` node) by the factor 0.7.

3.3 Modelling Expressive Prosody

Despite recent improvements in signal generation methods, it is still the case that synthetic speech sounds monotonous and generally inexpressive. Most systems deliberately aim to produce neutral readings of plain text; they do not try to interpret the text nor construct a spoken phrase to have some desired result. This lack of expressiveness is due to the poverty of the underlying linguistic representation: text analysis and understanding systems are simply not capable of delivering high-quality interpretations directly from unmarked input. However for many applications, such as information services, the text itself is generated by the computer system, and its meaning is available alongside information about the state of the dialogue with the user.

The problem then becomes how to mark up the appropriate information structure and discourse function of the text in such a way that the speech generation system can deliver appropriate and expressive prosody. Note that neither the SABLE system nor the MATE project address this problem directly. As can be seen from the example, SABLE is typically used to simply indicate emphasis, or to fiddle with prosody parameters directly. Mark up in MATE is a standard for actual human discourse, not for input to synthesis systems.

In the SOLE project, descriptions of museum objects are automatically generated and spoken by a TTS system. The application thus has knowledge of the meaning and function of the text. To obtain effective prosody for such descriptions, XML mark-up is used to identify rhetorical structure, noun-phrase type, and topic/comment structure, on top of standard punctuation [3].

Here is a simple example of text marked up for rhetorical relations:

```
<rhet-elem type="contrast">
  <nucleus> The
    <rhet-emph type="object">
      god </rhet-emph>
    was
    <rhet-emph type="property">
      gilded </rhet-emph>;
  </nucleus>
  <nucleus> the
    <rhet-emph type="object">
      demon </rhet-emph>
    was
    <rhet-emph type="property">
       stained in black ink and
polished to a high sheen
    </rhet-emph>.
  </nucleus>
```

```
</rhet-elem>
```

In this example, a contrast is drawn between the gilding of the god and the staining of the demon. The rhetorical structure is one of contrast, and contains elements of rhetorical emphasis appropriate for objects and properties.

It is clear that much further work is required in this area: in particular to decide on which aspects of information structure or discourse function have effects on prosody. Mark-up for dialogue would also have to take into account the modelled state of the listener; it would indicate which information was given, new or contradictory. Such mark-up might also express the degree of 'certainty' of the information, it might convey 'urgency' or 'deliberation'; even 'irritation' or 'conspiracy'.

4. CONCLUSIONS

This is an exciting time for synthesis: open architectures and open sources, large corpora, powerful computer systems, quality public-domain resources. But the availability of these has not replaced the need for detailed phonetic and linguistic analysis of the interpretation and realisation of linguistic structures. Progress will require the efforts of a multidisciplinary team distributed across many sites. XML provides standards, open architectures, declarative knowledge formalisms, computational flexibility and computational efficiency to support future speech generation systems. Rather than being a regressive activity, standards development forces us to address significant issues in the classification and representation of linguistic events in spoken discourse.

5. ACKNOWLEDGEMENTS

Thanks to the ProSynth project team in York, Cambridge and UCL. Thanks to COST258 for providing a forum. ProSynth is supported by the U.K. Engineering and Physical Sciences Research Council.

6. REFERENCES

[1] Hawkins, S., House, J., Huckvale, M., Local, J., Ogden, R., (1998). ProSynth: an integrated prosodic approach to device-independent natural-sounding speech synthesis. *Proc. Int. Conf. Spoken Language Processing, Sydney*, 1707-1710.

[2] Heid, S., & Hawkins, S., (1998) PROCSY: A hybrid approach to high-quality formant synthesis using Hlsyn. *Proc. 3rd ESCA/COCOSDA International Workshop on Speech Synthesis, Jenolan caves, Australia*, 219-224.

[3] Hitzeman, J., Black, A., Mellish, C., Oberlander, J., Poesio, M., Taylor, P. (1999) An annotation scheme for concept-to-speech synthesis, *Proc. European Workshop on Natural Language Generation, Toulouse France*, 59-66.

[4] Huckvale, M.A., (1999). Representation and processing of linguistic structures for an all-prosodic synthesis system using XML. *Proc. EuroSpeech-99, Budapest*, 1847-1850.

[5] Isard, A., McKelvie, D., Thompson, H., (1998). Towards a minimal standard for dialogue transcripts: a new SGML architecture for the HCRC map task corpus. *Proc. Int. Conf. Spoken Language Processing, Sydney*, 1599-1602.

[6] Klatt, D., (1979). Synthesis by rule of segmental durations in English Sentences. *Frontiers of Speech Communication Research, ed. Lindblom, B., Ohman, S.*, New York:Academic Press.