XiSTS – XML in Speech Technology Systems

Michael Walsh Stephen Wilson Julie Carson-Berndsen
Department of Computer Science
University College Dublin
Ireland
{michael.j.walsh, stephen.m.wilson, julie.berndsen}@ucd.ie

Abstract: This paper describes the use of XML in three generic interacting speech technology systems. The first, a phonological syllable recognition system, generates feature-based finite-state automaton representations of phonotactic constraints in XML. It employs axioms of event logic to interpret multilinear representations of speech utterances and outputs candidate syllables to the second system, an XML syllable lexicon. This system enables users to generate their own lexicons and its default lexicon is used to accept or reject the candidate syllables output by the speech recognition system. Furthermore its XML representation facilitates its use by the third system which generates additional lexicons, based on different feature sets, by means of a transduction process. The applicability of these alternative feature sets in the generation of synthetic speech can then be tested using these new lexicons.

1. Introduction

The flexibility and portability provided by XML, and its related technologies, result in them being well suited to the development of robust, generic, Natural Language Processing applications. In this paper we describe the use of XML within the context of speech technology software, with a particular focus on speech recognition. We present a framework, based on the model of *Time Map Phonology* (Carson-Berndsen, 1998), for the development and testing of phonological well-formedness constraints for generic speech technology applications. Furthermore, we illustrate how the use of a syllable lexicon, specified in terms of phonological features, and marked-up in XML, contributes to both speech recognition and synthesis. In the following sections three inter-connected systems are discussed. The first, the Language Independent Phonotactic System, LIPS, a syllable recognition application based on *Time Map Phonology* and a significant departure from current ASR technology, is described. The second system, Realising Enforced Feature-based Lexical Entries in XML, REFLEX, is outlined and finally, the third system, Transducing Recognised Entities via XML, T-REX, is discussed. All three systems build on earlier work on generic speech tools (Carson-Berndsen, 1999; Carson-Berndsen & Walsh, 2000a).

2. The *Time Map* Model

This paper focuses on representing speech utterances in terms of non-segmental phonology, such as autosegmental phonology (Goldsmith, 1990), where utterances are represented in terms of tiers of autonomous features (autosegments) which can spread across a number of sounds. The advantage of this approach is that coarticulation can be modelled by allowing features to overlap. The *Time Map* model (Carson-Berndsen, 1998, 2000) builds on this autosegmental approach by allowing multilinear representations of autonomous features to be interpreted by an event-based computational linguistic model of phonology. The *Time Map* model employs a phonotactic automaton (finite-state representation of the permissible combinations of sounds in a language), and axioms of event logic, to interpret multilinear feature representations. Indeed, much recent research (e.g. Ali et al., 1999; Chang, Greenberg & Wester, 2001) has focused on extracting similar features to those used in our model. Figure 1 below, illustrates a mulitlinear feature-based representation of the syllable [So:n][1].

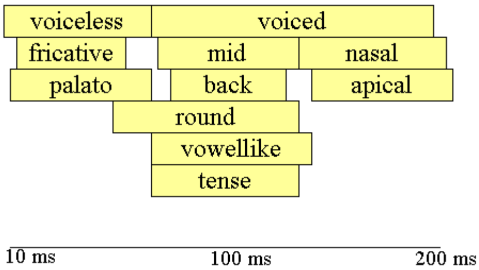


Figure 1. Multilinear representation of [So:n]

Two temporal domains are distinguished by the *Time Map* model. The first, *absolute* (signal) time, considers features as events with temporal endpoints. The second, *relative* time, considers only the temporal relations of overlap and precedence as salient. Input to the model is in absolute time. Parsing, however, is performed in the relative time domain using only the overlap and precedence relations, and is guided by the phonotactic automaton which imposes top-down constraints on the relations that can occur in a particular language. The construction of the phonotactic automaton and the actual parsing process is carried out by LIPS.

3. LIPS

LIPS is the generic framework for the *Time Map* model. It incorporates finite-state methodology which enables users to construct their own phonotactic automata for any language by means of a graphical user interface. Furthermore, LIPS employs an event logic, enabling it to map from absolute time to relative time, and in a novel approach to ASR, carry out parsing on the phonological feature level. The system is comprised of two principal components, the network generator and the parser, outlined in the following subsections.

3.1. The Network Generator

The network generator interface allows users to build their own phonotactic automata. Users input node values and select from a list of feature overlap relations those that a given arc is to represent. These relations can be selected from a default list of IPA-like features or the user can specify their own set. In this way LIPS is feature-set independent. The network generator constructs feature-based networks and parsing takes place at the feature level. Once the user has completed the network specification, the system generates an XML representation of the phonotactic automaton. An automaton representing a small subsection of the phonotactics of English is illustrated in Figure 2. It is clear from this automaton that English permits an [S] followed by a [r] in syllable-initial position, but not the other way around.

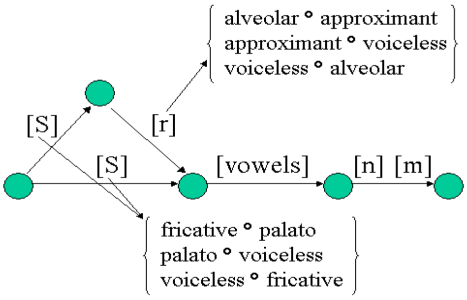
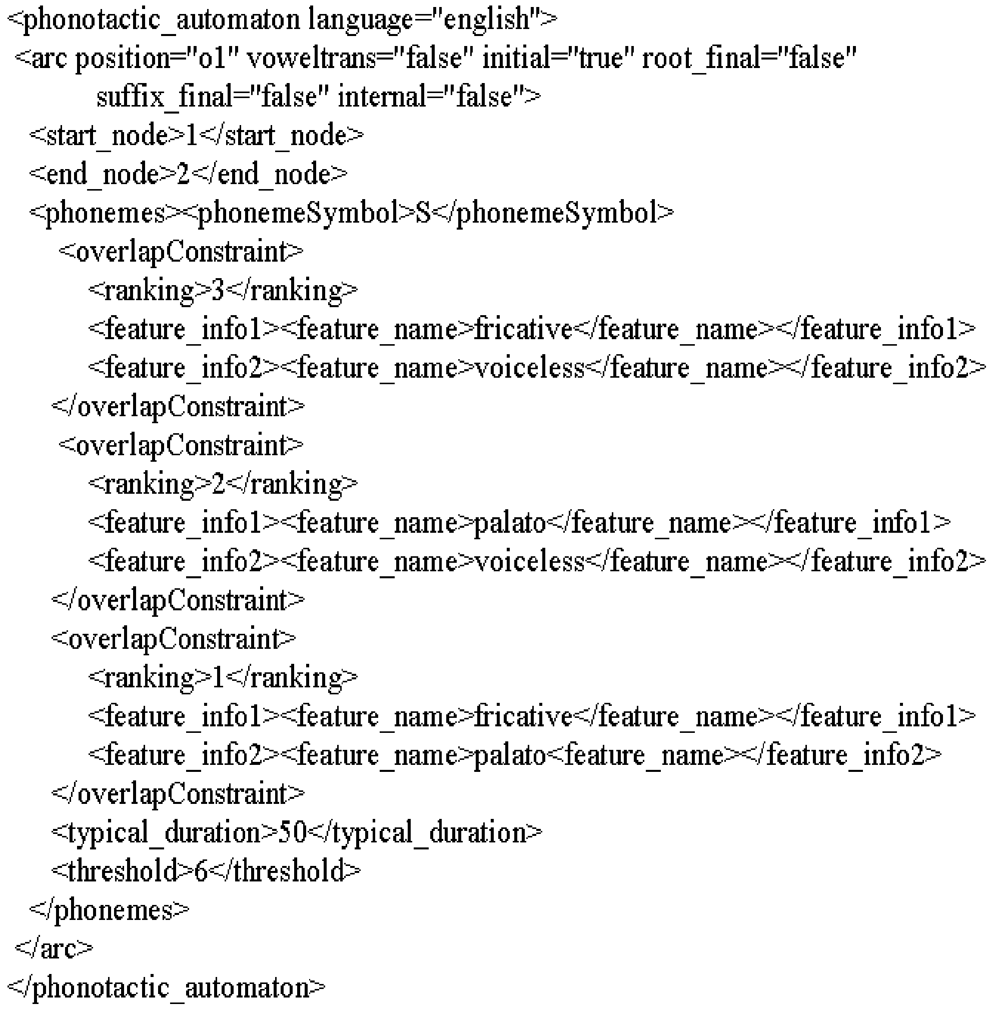


Figure 2. Phonotactic automaton

[1] All phonemes are specified in SAMPA notation.



```
<phonotactic_automaton language="english">
 <arc position="o1" voweltrans="false" initial="true" root_final="false"
          suffix_final="false" internal="false">
  <start_node>1</start_node>
  <end_node>2</end_node>
  <phonemes><phonemeSymbol>S</phonemeSymbol>
     <overlapConstraint>
        <ranking>3</ranking>
        <feature_info1><feature_name>fricative</feature_name></feature_info1>
        <feature_info2><feature_name>voiceless</feature_name></feature_info2>
    </overlapConstraint>
     <overlapConstraint>
        <ranking>2</ranking>
        <feature_info1><feature_name>palato</feature_name></feature_info1>
        <feature_info2><feature_name>voiceless</feature_name></feature_info2>
    </overlapConstraint>
    <overlapConstraint>
        <ranking>1</ranking>
        <feature_info1><feature_name>fricative</feature_name></feature_info1>
        <feature_info2><feature_name>palato<feature_name></feature_info2>
    </overlapConstraint>
    <typical_duration>50</typical_duration>
    <threshold>6</threshold>
  </phonemes>
 </arc>
</phonotactic_automaton>
```

Figure 3. XML representation of subsection of phonotactic automaton for English.

Figure 3 illustrates a subsection of the XML representation of the English phonotactics output by the network generator. A single arc with a single phoneme, [S], and its overlap constraints, is shown.

The motivation for generating an XML representation for our phonotactic automata is that XML enables us to specify a well-defined, easy to interpret, portable template, without compromising the generic nature of the network generator. That is to say the user can still specify a phonotactic automaton independent of any language or feature-set. The generated phonotactic automaton is then used to guide the second principal component of the system, the parser.

3.2 The Parser

LIPS employs a top-down and breadth-first parsing strategy and is best explained through exemplification.

Purely for the purposes of describing how the parsing procedure takes place, we return to the phonotactic automaton of Figure 2, which of course represents only a very small subsection of English. This automaton will recognise such syllables as *shum*, *shim*, *shem*, *shown*, *shrun*, *shran* etc., some being actual lexicalised syllables of English and others being phonotactically well-formed, potential, syllables of English. For our example we take the multilinear representation of the utterance [So:n] as depicted in Figure 4 as our input to the parser.

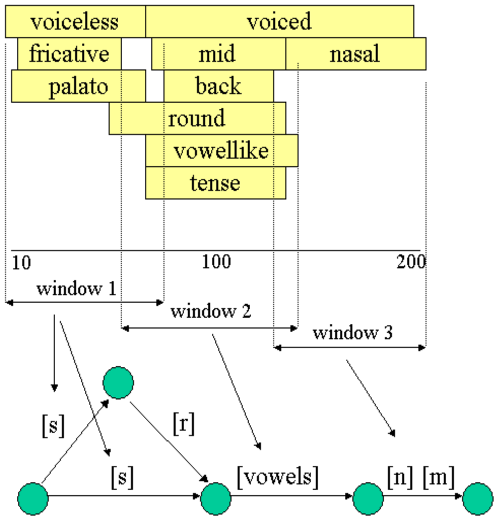


Figure 4. Interaction between the input and the automaton.

At the beginning of the parsing process the phonotactic automaton is anticipating a [S] sound, that is it requires three temporal overlap constraints to be satisfied, the feature *voiceless* must overlap the feature *fricative*, the feature *palato* must overlap the feature *voiceless*, and the feature *fricative* must overlap the feature *palato*. A variable window is applied over the input utterance and the features within the window are examined to see if they satisfy the overlap constraints. As can be seen from Figure 4 the three features are indeed present and all overlap in time. Thus the [S] is recognised and the two arcs bearing the [S] symbol are traversed and the window moves on. At this point then the automaton is anticipating either an [r] or a vowel sound. In a similar fashion the contents of the new window are examined and in the case of our example the vowel [o:] is recognised (the [r] is rejected). The vowel transition is traversed, the window moves on, and the automaton is expecting an [n] or an [m]. For full details of the parsing process see Carson-Berndsen & Walsh (2000b). Output from LIPS is then fed through the REFLEX system to determine if actual or potential syllables have been found.

4. REFLEX

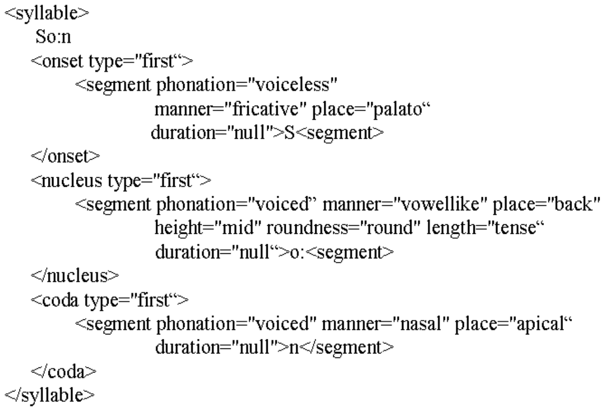
REFLEX is a generic, language independent application, which allows for the rapid design and construction of syllable lexicons, for any language. One of the main focuses of other research working on broadening the scope of the lexicon across languages, has been in the development of multilingual lexicons. One such project, PolyLex (Cahill & Gazdar, 1999), captures commonalities across related languages using hierarchical inheritance mechanisms. One of the main concerns of the work presented here however, is to provide generic, reusable, tools which facilitate the development and testing of phonological systems, rather than the creation of such multilingual lexicons.

Work on phonological features and lexical description has either been within this multilingual context (Tiberius & Evans, 2000) or has concentrated on using a feature-based lexicon for comparison with features extracted from a sound signal (Reetz, 2000). By removing reference to specific languages and concentrating on providing mechanisms for lexical generation, REFLEX can generate a syllable lexicon for any language that can be adequately represented in a phonetic notation.

Furthermore, the decision to use XML to represent the output data means that it is readily available for use and manipulation by other outside systems with minimal effort. All background processing is completely hidden; one deals only with the marked-up output, from which idiosyncratic user-required structures can be rapidly generated.

The REFLEX system outputs a feature-based syllable lexicon. This lexicon is a valid XML document, meaning that it conforms to the given REFLEX Document Type Definition (DTD). The DTD stipulates the structure, order and number of XML element tags and attributes, modelling all potential syllable structures (e.g. V, CV, CVC etc).

An example of a typical lexical entry, in this case corresponding to the multilinear representation specified in Figure 5, [So:n] is given below.



```
<syllable>
    So:n
   <onset type="first“>
         <segment phonation="voiceless"
                    manner="fricative" place="palato“
                    duration="null">S<segment>
   </onset>
   <nucleus type="first“>
         <segment phonation="voiced” manner="vowellike" place="back"
                    height="mid" roundness="round" length="tense“
                    duration="null“>o:<segment>
   </nucleus>
   <coda type="first“>
         <segment phonation="voiced" manner="nasal" place="apical“
                    duration="null">n</segment>
   </coda>
</syllable>
```

Figure 5. Typical lexical entry in XML

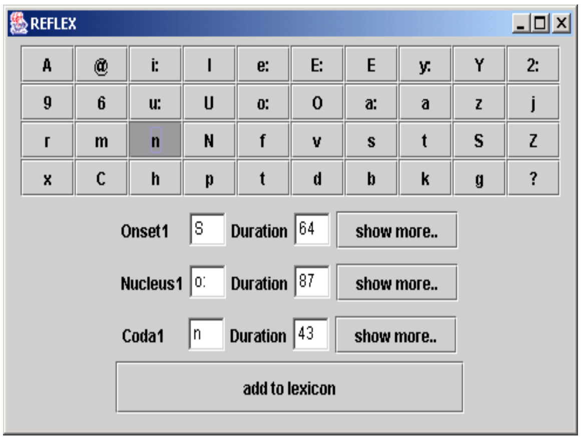
The syllable element shown has four children, described as follows:

1) A text child, in this case *So:n*, the SAMPA representation of the entire syllable. 2) An *<onset>* element whose attribute list denotes its position within the syllable, i.e.*<onset type=”first”>*, *<onset type=”second”>* etc. 3) Nucleus and 4) coda elements are similarly defined.

Each of the syllable's elements, *<onset>*, *<nucleus>* and *<coda>*, may have only one child element, *<segment>*, which tags the given phoneme. Its attribute list describes the phonemes specification in terms of phonological features. It also has a duration attribute, which is derived from corpus analysis.

<segment phonation=”voiced”
  manner=”nasal” place=”apical”
 duration=”null”>n</segment>

REFLEX provides two methods by which syllables can be added to the lexicon. The first, requires users to specify an input file of monosyllables represented in a phonetic notation, in this case SAMPA. The second, enables the user to specify syllables, in terms of phonemes, position, and if desired, a typical duration, by means of a GUI illustrated below in Figure 6.



Onset1 S Duration 64
Nucleus1 o: Duration 87
Coda1 n Duration 43

Figure 6. REFLEX lexicographer interface

Regardless of the input option chosen, new entries are added to the lexicon via a background process. REFLEX makes use of DATR, a non-monotonic inheritance based lexical representation language (Evans & Gazdar, 1996) to carry out this process. DATR is used to quickly and comprehensively define the phonological feature descriptions for a given language. For a greater understanding of how this can be achieved see Cahill, Carson-Berndsen & Gazdar (2000). Using DATR's inference mechanisms, REFLEX manipulates the output into a valid XML document, creating a sophisticated phonological feature-based lexicon, shown in Figure 5.

All syllable elements are enclosed within the root <*lexicon*> tag, whose sole attribute specifies the lexicon's language.

```
<lexicon language=”English”>
          <syllable>…</syllable>
               :
           <syllable>…</syllable>
</lexicon>
```

The REFLEX lexicon is a versatile tool that has a number of potential applications within the domain of speech technology systems. The following sub-sections illustrate how this syllable lexicon, by virtue of its being marked up in XML, can contribute to both speech recognition and synthesis.

4.1 LIPS and REFLEX

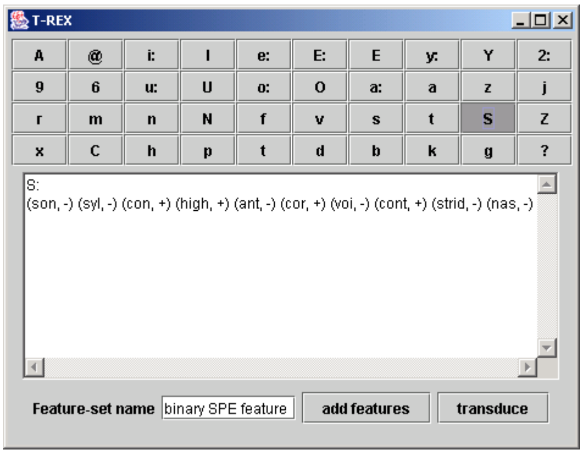
By allowing feature overlap constraints to be relaxed in the case of underspecified input, LIPS can produce a number of candidate syllables. In Figure 4 above, at the final transition, the automaton is expecting either an [m] or an [n]. The input, however, is underspecified, no feature distinguishing between [m] or [n], or indeed any voiced nasal, is present. By allowing the overlap constraints for the [m] and the [n] to be relaxed, LIPS can consider both [So:n] and [So:m] to be candidate syllables for the utterance. Both candidate syllables are well-formed, adhering to the phonotactics of English, however only one, [So:n], is an actual syllable of English. Thus at this point a lexicon providing good coverage of the language should reject [So:m] and accept [So:n]. In order to achieve this, REFLEX makes use of the XPath specification (a means for locating nodes in an XML document) and formulates a query before applying it to the syllable lexicon.[2] In the example given, REFLEX searches the document, checking the value of the text child of each syllable element, against each candidate syllable output by LIPS. Any successful matches returned are therefore not only well-formed, but are deemed to be actual syllables. Thus at this point, the lexicon is searched and the syllable [So:n] is recognised. The granularity of the REFLEX search capability is such, that it can be extended to the feature level. Users can search the lexicon for syllables that contain a number of specific features in certain positions, e.g. search for syllables that contain a voiced, labial, plosive in the first onset. Again, REFLEX forms an XPath expression and queries the lexicon, returning all matches. REFLEX also functions as a knowledge source for the T-REX system. This system is responsible for mapping output from the lexicon into syllable representations using different feature sets, e.g. features from other phonologies, and is discussed below in the context of speech synthesis.

5. T-REX

The role of this module is to enable lexicographers and speech scientists etc. to generate, via a transduction process, syllable lexicons based on different phonological feature sets. The default feature set employed by REFLEX is based on IPA-like features. However, T-REX provides a GUI that permits lexicographers to define phoneme to feature attribute mappings. Given this functionality T-REX operates as a testbed for investigating the merits of different feature sets in the context of speech synthesis. Different lexicons are generated by associating new feature sets with the same phonetic alphabet (SAMPA) via a GUI. The new lexicon is then transduced by T-REX which maps all syllable entries from the default lexicon (with IPA-like features) to the new lexicon,

[2] The full W3C XPath specification can be found at http://www.w3c.org/TR/xpath

applying the features input by the user, to their associated phonemes. In order to exemplify this we return to our sample syllable, [So:n]. Figure 2 above shows the lexical representation, using IPA-like features, for [So:n]. Figure 7 below shows new features being associated with the phoneme [S].



A	@	i:	I	e:	E:	E	y:	Y	2:
9	6	u:	U	o:	O	a:	a	z	j
r	m	n	N	f	v	s	t	S	Z
x	C	h	p	t	d	b	k	g	?

```
S:
(son, -) (syl, -) (con, +) (high, +) (ant, -) (cor, +) (voi, -) (cont, +) (strid, -) (nas, -)
```

Feature-set name: binary SPE feature

Figure 7. GUI for T-REX

Similarly, new features are associated with the remaining phonemes, [o:] and [n], and indeed the rest of the SAMPA alphabet. On completion the user initiates the transduction process and a new lexicon is produced. The XML representation of the phoneme [S], in the new lexicon, is depicted in Figure 8. Note how the feature attributes differ from those in the default lexicon.

```
<onset type="first">
    <segment son="-" syl="-" con="+" high="+" ant="-"
                  cor="+" voi="-" cont="+"
                  strid="+" nas="-">S</segment>
</onset>
```

Figure 8. Phoneme with transduced features

The advantages of this transduction capability are that numerous lexicons can be rapidly developed and used to investigate the appropriateness of specific formal models of phonological representation for the purposes of speech synthesis. Furthermore, the same computational phonological model, i.e. the *Time Map* model, can be employed. Bohan et al (2001) describe how the phonotactic automaton is used to generate a multilinear event representation of overlap and precedence constraints for an utterance, which is then mapped to control parameters of the *HLsyn* (Sensimetrics Corporation) synthesis engine. Different feature sets can be evaluated by assessing how they influence the various control parameters of the *HLsyn* engine and the quality of the synthesised speech.

6. Conclusion

This paper has described how the use of XML together with a computational phonological model can contribute significantly to the tasks of speech recognition, speech synthesis and lexicon development. Phonotactic automata and multilinear representations were introduced and the interpretation of these representations was discussed. Three robust, well-defined systems, LIPS, REFLEX, and T-REX, were outlined. These systems offer generic structures coupled with the portability of XML. In doing so, they enable users to recognise speech, synthesise speech, and develop lexicons for different languages using different feature sets while maintaining a common interface. The generic and portable nature of these systems means that languages with significantly different phonologies are supported. In addition, languages which, to date, have received little attention with respect to speech technology are equally provided for.

Ongoing projects include work on Irish, which has a notably different phonology from English and on developing phonotactic automata and phonological lexicons for other languages. Furthermore, the models are being extended to include phonemegrapheme mappings based on the contexts defined by the phonotactic automata.

7. Bibliography

Ali, A.M..A.; J. Van der Spiegel; P. Mueller; G. Haentjaens & J. Berman (1999): An Acoustic-Phonetic Feature-Based System for Automatic Phoneme Recognition in Continuous Speech. In: *IEEE International Symposium on Circuits and Systems (ISCAS-99)*, III-118-III-121, 1999.

Bohan, A.; E. Creedon, , J. Carson-Berndsen & F. Cummins (2001): Application of a Computational Model of Phonology to Speech Synthesis. In: *Proceedings of AICS2001*, Maynooth, September 2001.

Cahill, L. & G. Gazdar (1999). The PolyLex architecture: multilingual lexicons for related languages. *Traitement Automatique des Langues,* 40(2), 5-23.

Cahill, L.; J. Carson-Berndsen & G. Gazdar (2000), Phonology-based Lexical Knowledge Representation. In: F. van Eynde & D. Gibbon (eds.) *Lexicon Development for Speech and Language Processing,* Kluwer Academic Publishers, Dordrecht.

Carson-Berndsen, J. (1998): *Time Map Phonology: Finite State Models and Event Logics in Speech Recognition*. Kluwer Academic Publishers, Dordrecht.

Carson-Berndsen, J. (1999): A Generic Lexicon Tool for Word Model Definition in Multimodal Applications. *Proceedings of EUROSPEECH 99, 6th European Conference on Speech Communication and Technology*, Budapest, September 1999.

Carson-Berndsen, J. (2000): Finite State Models, Event Logics and Statistics in Speech Recognition, In: Gazdar, G.; K. Sparck Jones & R. Needham (eds.): *Computers, Language and Speech: Integrating formal theories and statistical data*. Philosophical Transactions of the Royal Society, Series A, 358(1770), 1255-1266.

Carson-Berndsen, J. & M. Walsh (2000a): Generic techniques for multilingual speech technology applications, *Proceedings of the 7th Conference on Automatic Natural Language Processing*, Lausanne, Switzerland, 61-70.

Carson-Berndsen, J. & M. Walsh (2000b): Interpreting Multilinear Representations in Speech. In: *Proceedings of the Eight International Conference on Speech Science and Technology*, Canberra, December 2000.

Chang, S.; S. Greenberg & M. Wester (2001): An Elitist Approach to Articulatory-Acoustic Feature Classification. In: *Proceedings of Eurospeech 2001*, Aalborg.

Evans, R & G. Gazdar (1996), DATR: A language for lexical knowledge representation. In: *Computational Linguistics* 22, 2, pp. 167-216.

Goldsmith, J. (1990): *Autosegmental and Metrical Phonology*. Basil Blackwell, Cambridge, MA.

Reetz, H. (2000) Underspecified Phonological Features for Lexical Access. In: *PHONUS 5*, pp. 161-173. Saarbrücken: Institute of Phonetics, University of the Saarland.

Tiberius, C. & R. Evans, 2000 "Phonological feature based Multilingual Lexical Description," *Proceedings of TALN 2000*, Geneva, Switzerland.