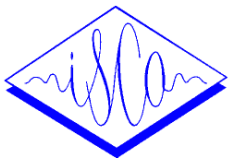


The ESCA Workshop
on Speech Synthesis
Autrans, France
September 25-28, 1990

A MODULAR APPROACH TO MULTI-DIALECT AND MULTI-LANGUAGE SPEECH SYNTHESIS USING THE DELTA SYSTEM

Susan R. Hertz

Eloquent Technology, Inc.
24 Highgate Circle, Ithaca, New York, 14850

and

Dept. of Modern Languages and Linguistics
Cornell University, Ithaca, New York 14853

ABSTRACT

This paper describes a *modular approach* to rule-based speech synthesis that we have been using for English dialects and have begun to extend to multiple languages. In the modular approach, a single program, divided into language-universal, language-specific/dialect-universal, and dialect-specific rule modules, is used to synthesize a number of dialects or languages. The approach is based on a new "multi-stream" phonetic model and is implemented in Delta, a fourth-generation programming language designed for developing synthesis rules based on such models. Besides the theoretical significance of such an approach, it results in more cost-effective development and implementation of synthesis rule sets than possible by writing separate programs or by modularizing within general-purpose languages such as C.

1. INTRODUCTION

This paper gives an overview of the modular approach as applied to the synthesis of American English dialects. While the paper focuses on American dialects, with specific examples taken from General American (GA) and a Black English dialect (BE) spoken in Richmond, Virginia, only the text-to-phone rules are specific to American English; the general phone-to-speech strategy applies to all languages.

2. THE DELTA UTTERANCE REPRESENTATION

Central to the modular approach is the *delta*, a multi-stream utterance representation upon which the Delta language is built (Hertz, 1988a; Hertz, 1988b). Below are the final deltas for GA and BE *pie* that would result from applying the appropriate rule sets. Subsequent sections explain the various parts of these deltas and show how the values are inserted by the particular modules.

(1) GA:

```
text:        | p  |      |       i       |      e      |
syllable:    |                  stress1                  |
diaphoneme:  | p  |      |              ai             |
phone:       | p  |      |        a      |      | i    |
formant_1:   |400 |      |700            |      |300   |
formant_2:   |1000|      |1200|      |1400|      |1900  |
aspiration:  |0   |60    |0                            |
voicing:     |0           |55                           |
transition:  |    |trans|     |trans|     |trans|      |
millisec:    |75  |70   |0    |117  |0    |120  |30    |
```

(2) BE:

```
text:        | p  |      |i  | e|
syllable:    |    stress1      |
diaphoneme:  | p  |      | ai  |
phone:       | p  |      | ah  |
formant_1:   |900 |      |920  |
formant_2:   |1700|      |1760 |
aspiration:  |0   |60    |0    |
voicing:     |0           |55   |
transition:  |    |trans|      |
millisec:    |75  |70   |200  |
```

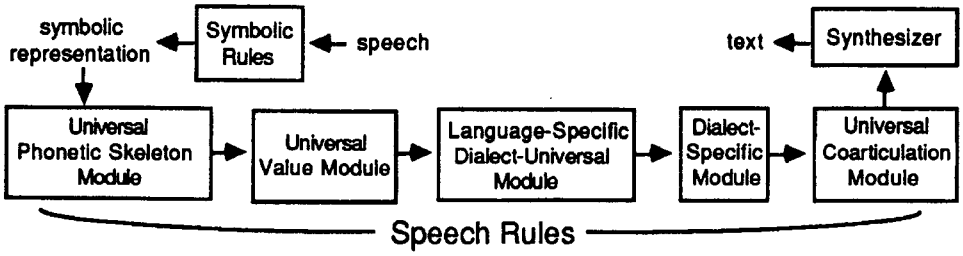
Each horizontal line of a delta is called a *stream*, the number and function of which are defined in accordance with a model selected by the linguist. In our model, streams represent abstract linguistic structure (here, `syllable` down to `phone`), acoustic structure (`formant_1` down to `transition`), and timing (`millisec`). Each stream consists of a sequence of *tokens* (e.g., `p`, `i`, and `e` in the `text` stream), separated by vertical bars called *sync marks*, which coordinate tokens across streams. Although not shown, tokens may have associated user-defined features that describe their acoustic and physiological properties (e.g., `consonant` or `vowel`, `labial`,

alveolar, or palatal). In this paper, only selected streams are shown; a complete delta for synthesis would include, among others, a stream for each of the synthesizer parameters.

3. THE MODULAR APPROACH

The modular approach can be diagrammed as follows.

(3)



The delta begins with only the text stream filled in, representing ordinary spelling. The *symbolic rules* operate first, filling in the abstract linguistic streams (syllable etc.). Then, using the output of the symbolic rules, the *speech rules*, on which this paper focuses, fill in the acoustic streams for aspiration (dB), voicing (dB), formants (Hz), and transitions (described below), as well as their timing (millisec), from which actual values for the synthesizer are generated. The synthesizer parameter values represent steady state values held for the associated number of milliseconds (sometimes 0) and the transitions represent linear interpolations.

3.1 The Symbolic Rules

The symbolic rules are specific to a language. For American English, they comprise dialect-universal and dialect-specific modules. The diaphoneme stream for American English (see delta (1) above) contains underlying "phonemes" common to all dialects (cf. Troike, 1971), and the phone stream contains the phones particular to the dialect being synthesized. The dialect-universal symbolic module fills in the diaphoneme stream, while the dialect-specific modules map from diaphonemes to phones. These dialect-specific rules, for example, would realize the underlying diaphoneme ai in *pie* as the two phones a and i in GA, but as a single phone, represented by us as ah, in BE, as shown in deltas (1) and (2).

3.2 The Speech Rules

The speech rules contain language-universal, language-specific/dialect-universal, and dialect-specific modules.

First, the *universal phonetic skeleton module* creates the language-universal structure for each phone, comprising, for example, two target "slots" for each formant, positioned precisely at the edges of the phones. Explicit transitions, represented by trans tokens in the transition stream, are positioned between the slots and between phones, as illustrated for the second formant in the delta fragment below for BE *pie*. The GAP tokens hold places to be filled later with the appropriate values.

```
(4) phone:      |       p        |      |       ah       |
    formant_2:  |GAP |     |GAP |      |GAP |     |GAP |
    aspiration: |GAP             |GAP   |GAP             |
    voicing:    |GAP             |GAP   |GAP             |
    transition: |    |trans|     |trans|    |trans|     |
    millisec:   |GAP |GAP  |GAP  |GAP  |GAP | GAP |GAP  |
```

Our experience (see Section 4) suggests that the two-slot formant structure will be adequate for all phones in all languages. Explicit transitions, not incorporated into phones, are crucial to the division of rules into universal and specific modules. They also result in more natural and powerful rules for the timing of many acoustic events (Hertz, 1988b; Hertz, in press).

The *universal value module* fills in the universal phonetic skeleton with acoustic values common to all or most languages, as illustrated below for our BE example:

```
(5) phone:       |       p          |      |       ah       |
    formant_2:   |1000 |      |1000 |      |GAP |       |GAP |
    aspiration:  |0                 |60    |0                |
    voicing:     |0                 |0     |55               |
    transition:  |      |trans|     |trans|     |trans|     |
    millisec:    |GAP   |GAP  |GAP  |GAP  |GAP  |GAP  |GAP  |
```

Here, the `formant_2` values are filled in for `p` and the `voicing` for both phones and their intervening transition (although some of these values will be modified by later rules, as discussed below for the second formant of `p`). Also, since `p` would have the feature `aspirated` (assigned by the symbolic rules), the token `60` (dB) in the `aspiration` stream is aligned with the transition between the stop and the following vowel, reflecting our observation for all languages we have studied that aspiration is coordinated with the transition following the "aspirated segment" (Hertz, 1988b; Hertz, in press). The degree (i.e., length) of aspiration, however, is language-specific, as discussed below, so the aspirated transition is not given a duration at this stage. In contrast, our studies suggest that between many other kinds of phones the transition durations will be universal.

The *language-specific/dialect-universal* module adds the values common to a group of dialects in the language. For English, these values include the formant values and durations of most consonants and the durations of transitions not specified by the universal value module. For example, the aspirated transition between `p` and `ah` in BE is given the duration 70:

```
(6) phone:       |       p          |      |       ah       |
    aspiration:  |0                 |60    |0                |
    transition:  |     |trans|      |trans|     |trans|     |
    millisec:    |0    |75   |0     |70   |GAP  |GAP  |GAP  |
```

The *dialect-specific module* replaces selected `GAP` tokens with dialect-specific values. For English, these include primarily formant values and durations for vowels, as illustrated below for the second formant of BE *pie*:

```
(7) phone:       |       p          |      |       ah       |
    formant_2:   |1000 |      |1000 |      |1760 |       |1760 |
    aspiration:  |0                 |60    |0                |
    voicing:     |0                 |0     |60               |
    transition:  |      |trans|     |trans|     |trans|     |
    millisec:    |0     |75   |0    |70   |0    |200  |0    |
```

In the last major step, the *universal coarticulation module* makes language-universal contextual modifications, such as the raising of a relatively low formant value in a consonant before a relatively high formant value in a following vowel. For example, the second formant target of BE `p` is changed from 1000 Hz to 1700 Hz before `ah`.

Finally, any remaining `GAP` placeholder tokens are removed, and all identical adjacent values in a stream are collapsed into single tokens, producing the final deltas, as shown in examples (1) and (2) above. From the final deltas, the system generates a set of acoustic values at five ms intervals for a version of the Klatt synthesizer (Klatt, 1980), which generates a speech waveform.

4. IMPLEMENTATION

Both the symbolic and speech rules are implemented in Delta, and compiled into C programs that can be run on many popular computer systems. The following Delta fragment of the dialect-specific module for GA, for example, expresses the fact that in GA all of the phones in a syllable nucleus are lengthened before a voiced obstruent, but by different degrees depending on the type of phone. Details of the Delta notation are beyond the scope of this paper, but the comment lines, beginning with double colons, should provide a general feel for the rules.

```
(8)  :: constrain patterns to operate in syllables:
          fence %syllable;
     :: for each syllable nucleus token that precedes a phone that is
     :: a voiced stop (within the same syllable)...
          forall [%nucleus _^begnuc <> !^endnuc [%phone <voiced stop>]] ->
     :: lengthen each phone within the nucleus by the appropriate
     :: amount, depending on the type of phone:
          forall [%phone _^begphon <> !endphon] from ^begnuc to ^endnuc ->
           if
            [_^begphon <vowel> ]   ->  dur(^begphon...^endphon) *= 1.7;
            [_^begphon <lateral>] ->  dur(^begphon...^endphon) *= 1.4;
            ...
           fi;
```

Note that this rule lengthens the phones, but not the intervening transitions, an approach justified in Hertz (in press). BE has a similar rule, but the degrees of lengthening are quite different. Also, in BE the inter-phone transitions lengthen as well, but not nearly as much as the phones.

5. STATE OF DEVELOPMENT

As of this writing, we are modularizing a symbolic rule set previously developed for GA (Hertz, 1981; McCormick and Hertz, 1989), to serve as a front end to our English speech rules. We have already tested the feasibility of the modular approach to speech rules by interactively synthesizing (with Delta's accompanying interactive development environment, DeltaTools) a number of representative utterances in seven American English dialects, and in Hebrew, German, French, Chinese, Japanese, and Hindi. On this basis, we have developed preliminary universal phonetic skeleton and value modules and a dialect-universal module for English. We are now developing dialect-specific rules for GA (based on earlier rules in SRS (Hertz, 1982)) and BE, with work soon to begin on the five other English dialects, and on German and Japanese (an extension of work by Hertz (1980) and Hertz *et al.*, (1983)). While the rule sets are not yet complete, we are encouraged by preliminary results. Tests show our interactively synthesized utterances to be highly intelligible, and development of the rules is proceeding rapidly, owing to the linguistic orientation of the Delta language and to the modularity of our approach.

ACKNOWLEDGEMENTS

The multi-dialect project has been supported in part by Contract RS89071002 from the U.S. Dept. of Education and by Grant 1R43DC00758-01 from the NIH, and the multi-language project just beginning by grant RSD 89174 from the New York State Science and Technology Foundation. Many thanks to Dr. David J. Lewis for his editorial assistance on this paper.

REFERENCES

Hertz, S. R. (1980) Multi-language speech synthesis—a search for synthesis universals, *JASA* **67**, S13.

Hertz, S. R. (1981) SRS text-to-phoneme rules: a three-level rule strategy, *Proc. IEEE ICASSP*, 102-105.

Hertz, S. R. (1982) From text to speech with SRS, *JASA*, **72**, 1155-1170.

Hertz, S. R. (1988a) Delta: flexible solutions to tough problems in speech synthesis by rule, *The Official Proceedings of Speech Tech 88*, New York: Media Dimensions Inc.

Hertz, S. R. (1988b) The Delta programming language: an integrated approach to non-linear phonology, phonetics, and speech synthesis, *Working Papers of the Cornell Phonetics Lab.* **2**, 69-122. Also to appear in *Papers in Laboratory Phonology I: Between the Grammar and the Physics of Speech* (J. Kingston and M. Beckman, editors) New York: Cambridge University Press.

Hertz, S. R. (in press). Streams, phones, and transitions: toward a new phonological and phonetic model of formant timing, to appear in *J. of Phonetics* **19**.

Hertz, S. R. and Beckman, M. (1983) A look at the SRS synthesis rules for Japanese, *Proc. ICASSP* **83**, 1336-1339.

Klatt, D. (1980) Software for a cascade/parallel formant synthesizer," *JASA* **82**, 737-793.

McCormick, S. and Hertz, S. (1989) A new approach to English text-to-phoneme conversion using Delta, version 2. *JASA* **85**, S124.

Troike, R. (1971). Overall pattern and generative phonology," *Readings in American Dialectrology*, (B. Allen and N. Underwood, editors) New York: Meredith Corporation, pp. 324-342.