Data structures in speech production

Mark Tatham
Katherine Morton
University of Essex, Colchester, UK
mark.tatham@btconnect.com
katherine.morton@btconnect.com

Computationally testable models in linguistics focus on declaring data structures and providing exemplar derivations. This paper outlines a comprehensive model of speech production which goes beyond derivations to show how actual instances of utterances can be formally characterised. Utterances contain a wealth of detail beyond the underlying utterance plan: some of this is a function of the mechanism itself (e.g. coarticulation) and some is the result of carefully supervised control. We develop the notion of managed or supervised speech production to enable the inclusion of EXPRESSIVE content in speech. Building on earlier work the Cognitive Phonetics Agent bridges the gap between the physical and cognitive processes in phonetics by controlling the way phonologically determined utterance plans are phonetically rendered in detail. The model is illustrated using different types of data structure which occur in speech, concentrating in particular on an XML characterisation of appropriate structures. We trace a simple utterance through from its phonological plan to a detailed intrinsic allophonic representation to show how stages in the model work.

1 Introduction

This paper outlines the basis of a model of human speech production that focuses on contributing to an understanding of detail present in an acoustic speech signal but which has so far eluded satisfactory explanation. Models of this kind should meet the criterion of computational adequacy in the sense that they are explicit, and lend themselves to formal declarative or procedural testing. Descriptive models which have no computational implementation are difficult to test and evaluate, and so do not help in establishing the theory they exemplify. In addition, most current models have gaps: it is usually not possible to trace a potential utterance through ALL its phonological and phonetic stages to arrive finally at an articulation or acoustic signal (Tatham, Morton & Lewis 1998, 2000).

Central to any testable model is a set of clear formalisms for the main data structures involved. Current phonological theory, for example, is inherently computational, and focuses primarily on data structures and derivations which cascade through these (Bird 2002). The declaration of data structures in speech production theory is our focus in this paper – not because they are a complete model in themselves but because they are a necessary foundation.

In this paper we use a declarative formalism for this purpose, adopting the technique prevalent in linguistics: characterise the knowledge base (the data structure system) first. The XML paradigm we mainly use has two important properties:

Journal of the International Phonetic Association (2003) 33/1
DOI:10.1017/S0025100303001142

- it has built in procedures for validating declared data structures – that is, it checks for coherence and accuracy, and
- it can be interpreted simply as though it were procedural.

The second of these is equivalent to characterising derivations in linguistics, rather than just enumerating a grammar. We say more about this property later and have incorporated it, though it is comparatively trivial and not central to our purpose here, which is to formulate the data structures themselves in a way that makes them amenable to casting light on where some of the detail present in the speech waveform comes from.

In our characterisation of data structures, two considerations are important:

- what are the elements needed within the data structure?
- what is the most appropriate formalism?

Here we give examples of a few of the main data structures we believe to be necessary, and explain the formalisms used.

But above all a computational model must be complete and coherent; if it is not, the computation will fail. By reason of its completeness such a model is inherently testable – it either runs or it does not: either way, weaknesses and areas needing more development will point toward establishing the basis for formal hypotheses for empirical investigation. If the model is complete it will run, but equally importantly the model must be correct to achieve the right output.

The theory of speech production assumes that an exhaustive and generalised description of speech production needs to exist separately from dynamic instantiations of any particular utterances. We think of the generalised characterisation as a static representation that provides the basis of a plan for a particular utterance. We propose a reasoning Cognitive Phonetic Agent (Tatham 1986c) which selects the appropriate structures from the static representation in order to render the utterance dynamically. The term DYNAMIC PHONETIC RENDERING will be explained. The layout of the model can be seen in figure 1. The Cognitive Phonetic Agent itself is described later.

We illustrate the model by taking a single sentence and tracing it through several phonological and phonetic processes of planning and rendering to arrive at a symbolic representation of how the utterance will finally be articulated. Along the way we show how the various data structures are manipulated. The processes we have selected for illustration exemplify problem areas in the theory, and give us the opportunity to discuss the various formalisms used. In particular we have concentrated on the theory's overall prosodic framework and the way we have introduced a first approximation to modelling expression in speech production.

2 Scope of the model

The theory of speech production we are dealing with depends on the idea that speakers know in general about the processes used in formulating utterance plans (phonology) and about what is involved in rendering utterance plans (phonetics). This aspect of the model is developed on what we call its STATIC PLANE. The word PLANE is used because it is useful to think of the model as having more than two dimensions, involving more than one plane. We label this plane STATIC because it is a fixed and simultaneous characterisation of everything which, from a linguistics point of view, contributes to the planning and rendering of ALL utterances in the language; it is equivalent to the 'grammar' in generative linguistics.

Parallel to the model's static plane is a DYNAMIC PLANE. It is here that the plan for any one utterance is developed by drawing on information held on the static plane. Because the static plane holds the information needed for formulating all utterances it must have the information necessary for developing any single utterance. The dynamic plane is like 'derivation' in linguistics, except that derivation is usually confined to instantiating

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

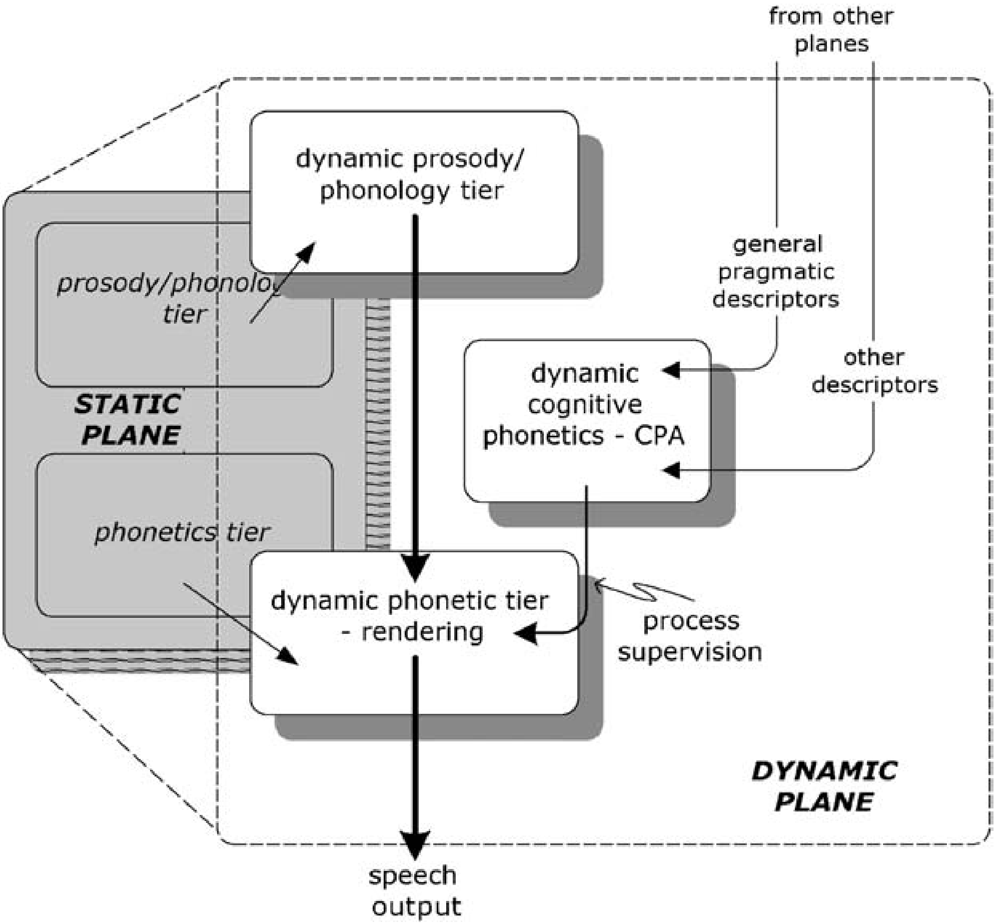


Figure 1 The multi-dimensional model showing the static plane BEHIND the dynamic plane. Each plane has a prosody/phonology tier and a phonetic rendering tier. Here the Cognitive Phonetic Agent is shown directly supervising phonetic rendering dependent on descriptors from other areas of the model.

the grammar. Our dynamic plane derives utterances by including grammar and ADDITIONAL information.

Figure 1 shows the relationship between the static and dynamic planes, together with, on the static plane, distinct sets of phonological and phonetic processes and, on the dynamic plane, an area on which a single utterance plan is developed (its unique phonology) and an area on which its plan is rendered (its unique phonetics). The model can trace the history of a single instantiation of a speech waveform, including factors other than the underlying grammar, rather than just enumerate the entire knowledge base supporting it.

But, if the dynamic development of the plan for a unique utterance and its subsequent rendering depend on information held on the model's static plan, it becomes necessary to set up the means to select and bring appropriate 'objects' and processes from the static plane to the dynamic areas. We do not think of the procedures for drawing on static information as purely automatic. They are developed here as informed and reasoned procedures which need special mechanisms. We call these mechanisms AGENTS, confining ourselves in this paper to some details of our Cognitive Phonetic Agent (CPA) whose supervisory role has been described elsewhere (Tatham 1994).

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

Some of the terminology used in characterising the CPA – including the term ‘agent’ itself – comes from computer science: but we confine ourselves strictly to speech theory in this paper, and do not digress into areas such as artificial intelligence or speech technology. A reasoning agent is simply a device that accepts data from a number of different sources and uses this information to arrive at an output. The output is said to be reasoned because it is not derived automatically but is the result of a process that evaluates the coming data potentially differently on each occasion.

One area of speech production theory in need of development is about accounting for expression. The general position taken by most researchers (Tatham & Morton 2003 in press) is that expressive speech is speech in which a listener can detect a particular MOOD, EMOTION or INTENTION. That is, besides what we might call the BASIC MEANING conveyed by the semantics of the actual words of the utterance and the way they are arranged syntactically, there is an additional element that enables listener reports like: ‘spoken angrily’ or ‘the speaker is happy’ or even ‘they like me’. Listener comments such as these refer to information separable from the words themselves.

Most speech production models have used two descriptive systems to handle plain semantic content and co-occurring expressive content. The first belongs to a purely linguistics domain, and the other originates in the bio-psychology (LeDoux 1996) and psychology (Frijda 1993) domains. To bring these into a unified characterisation we refer to the fact that the vocal tract configuration changes between different expressive modes, which is why expressive acoustic parameters like fundamental frequency average, f0 range and tempo all alter the SUBSEQUENT linguistic content.

In other words, a speaker attempts an utterance under certain altered physical conditions. These conditions, whether physically sourced (as in the tension associated with being very angry) or cognitively sourced (as in an attempt at persuasion) or both, DOMINATE the utterance in the speech production model we are developing here. The changed settings will dominate the WAY the speech is produced. The resulting changes to the physical act of speaking show up in the speech waveform. They are detectable and quantifiable – and perceived from the waveform. Thus listeners report invariant properties which correlate with expression within the acoustic signal.

If we think of the written version of an utterance we can see it can clearly be instantiated or read aloud as several different spoken versions ‘wrapped’ in different expressive content. The situation is analogous to scripting for drama:

John: *[sincerely]* I love her.

The actor playing John is to speak the words *I love her* sincerely, such that Mary (and indeed the audience) is to be in no doubt as to his intentions. The convention makes provision for such alternatives as:

John: *[angrily]* I love her.

and

John: *[dreamily]* I love her.

or, in the general case,

actor: [{mood, emotion, intention, . . .}] utterance

where {mood, emotion, intention, . . .} are types of expression.

This model of production (for that is what this drama scripting convention amounts to) makes a clear distinction between an utterance and the tone of voice in which it is to be produced. It is clear that *[emotion]* is a variable unconstrained by the spoken words, *[utterance]*, immediately following.

Translating the drama script analogy to our own general speech production perspective we might say that both the WHAT (the utterance itself) and the HOW (the tone of voice to be used)

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

are here represented separately as part of some underlying or abstract plan: they constitute the intention, and the intention is in two parts. What is in the script is a plan which underlies the acoustic signal the actor will produce. We say the plan is abstract because it is not itself an actual acoustic signal. The plan or script captures the INTENTION, rather than the actual signal, and intentions are abstractions.

In our own dynamic model of speech production an abstract intention or plan of an utterance has to be rendered. In the drama this is the job of the actor, and it is important to note that the rendering process is deliberately inexplicit in the above example: it is part of the actor's art. The rendering of a script – speaking utterances with appropriate tones of voice – requires artistic 'interpretation' of some kind. This is why we can point to the individuality of interpretations by different actors of the same passages in a play. Speech production modelling though is NOT about characterising an art. A speech production model must characterise explicitly how the two aspects of the plan are brought together to produce a composite soundwave that integrates the utterance and its expression.

Expression in speech could be characterised very abstractly in terms simply of generalisations and a speaker's potential for delivering expressive speech; but an understanding of what expression in speech IS and how it forms part of almost every utterance speakers make requires a focus on the INDIVIDUAL utterance and its subsequent rendering. We claim that utterances without expression can exist only in the abstract. A dynamically rendered utterance MUST have expression and the model must incorporate the means of explaining the utterance's expressive content. It falls to the Cognitive Phonetic Agent to supervise expressive content.

3 Cognitive Phonetics and supervision

We have argued before (Tatham 1986a) that there is a need to explain the manipulation of intrinsic physical processes in speech production. Such processes include coarticulation and, although they are intrinsic to the system and might at first glance be a-linguistic in origin, they can sometimes get involved in the linguistically motivated encoding process in very subtle ways. For example, simple or uncontrolled coarticulation effects are unplanned and are the result of mechanical and other constraints on juxtaposed articulations. But these effects can often be overridden, and when this happens they seem to be under cognitive control. We have grouped processes where cognition influences intrinsic physical processes under the heading of Cognitive Phonetics (Morton 1986, Tatham 1986b, Code & Ball 1988, Cawley & Green 1991) and distinguished these processes from truly phonological cognitive processes performed uniquely on phonological objects. Thus Cognitive Phonetics and the cognitive processes of phonology are defined in terms of the domain of the objects on which they operate: phonological processes manipulate phonological objects, Cognitive Phonetic processes manipulate phonetic objects.

Following early work, the principles of Cognitive Phonetic Theory were extended to account for what we felt had to be a managerial role for the group of processes involved. We noted that the precision of controlled physical processes varies enormously – not just because of intrinsic factors but because of deliberate tightening or relaxing of the precision of articulation. This varying precision turned out to be principled. Thus we introduced the idea of supervision in speech production (Tatham 1995). Supervision involves a pre-determined level of ACCURACY and a deliberate attempt to maintain that level of accuracy. All the functions of a control system are invoked, in particular feedback and the response to on-going success in maintaining the required level of precision.

The current model invokes a Cognitive Phonetic Agent (the CPA) – a device dedicated to supervising the overall rendering process. The CPA takes its instructions from a number of different sources (figure 1), and manages the rendering process to produce the desired articulation or acoustic signal goal. The signal in turn reflects underlying generalisations

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

about speech, but at the same time reflects this continuously varying precision and the accompanying expressive content.

4 Some data structure details

Central to any computational model is a clear representation of the various data structures involved. In this section we highlight some of the main data structures of the speech production model, and illustrate their organisation with some examples. A characterisation of data structures begins with their general case. In this model, the general case is represented on the static plane.

Much of the model is formalised in XML – a declarative language designed for characterising hierarchically structured data, and to make the characterisation suitable for subsequent procedural processing. A data structure is set out as an XML-schema which indicates in a formal way its most general case, including all constraints on its content. The rules governing XML-schema are complex, and a good way of approaching what XML can do is to focus on a very simple example which we are familiar with from a more traditional approach in phonology.

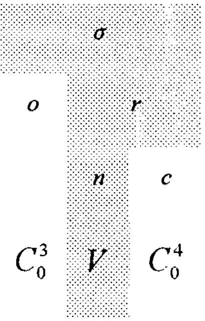
A simple declarative data structure: the syllable

The most useful hierarchical model of an English syllable in traditional metrical phonology (Hayes 1995) looks like this:

$$\begin{array}{lll} \sigma & \rightarrow & \text{onset} + \text{rhyme} \\ \text{rhyme} & \rightarrow & \text{nucleus} + \text{coda} \\ \text{onset} & \rightarrow & C_0^3 \\ \text{nucleus} & \rightarrow & V \\ \text{coda} & \rightarrow & C_0^4 \end{array}$$

where σ is a syllable, C_0^3 means a number of consonants from zero to three, and C_0^4 a number of consonants from zero to four; V is the vowel.

Its tabular or graphical form is as follows:



where σ is a syllable, o is the onset, r the rhyme, n the nucleus and c the coda, C_0^3 means a number of consonants from zero to three, and C_0^4 a number of consonants from zero to four, V is the vowel. The lowest elements, C_0^3, V and C_0^4, are subject to phonotactic constraints that will prevent sequences such as [mr . . .], [psl . . .] or [. . . tnft].

The shading in the table represents the part of the internal derivation which in traditional terms is not optional – that is, must produce a surface element. This is the direct path $\sigma \rightarrow r \rightarrow n \rightarrow V$. The alternative is to regard everything as not optional, but indicating that the onset C_0^3 and the coda C_0^4 have a zero instantiation option that is not to be regarded as a null element. The approach taken will depend on whether some surface detail in the eventual phonetic rendering is dependent on the influence of an underlying element formerly considered optional.

For our development work in XML we use the integrated development environment created by Altova GmbH and Altova Inc (Altova 1998–2001). Expressed as an XML-schema (equivalent to the grammar) the syllable data structure looks like this:

```
<?xmlversion = "1.0"encoding = "UTF − 8"?>
<xs : schema>
    <xs : element name = "syllable">
      <xs : annotation>
        <xs : documentation > prosodic object </xs : documentation>
      </xs : annotation>
      <xs : complexType>
        <xs : sequence>
           <xs : element name = "onset">
             <xs : complexType>
               <xs : sequence>
                 <xs : element name = "consonant"type = "xs : anySimpleType"
                                  minOccurs = "0" maxOccurs = "3"/>
               </xs : sequence>
             </xs : complexType>
           </xs : element>
           <xs : element name = "rhyme">
             <xs : complexType>
               <xs : sequence>
                 <xs : element name = "nucleus">
                   <xs : complexType>
                     <xs : sequence>
                       <xs : element name = "vowel"type = "xs : anySimpleType"/>
                     </xs : sequence>
                   </xs : complexType>
                 </xs : element>
                 <xs : element name = "coda">
                   <xs : complexType>
                     <xs : sequence>
                 <xs : element name = "consonant"type = "xs : anySimpleType"
                                  minOccurs = "0" maxOccurs = "4"/>
                     </xs : sequence>
                   </xs : complexType>
                 </xs : element>
               </xs : sequence>
             </xs : complexType>
           </xs : element>
         </xs : sequence>
       </xs : complexType>
     </xs : element>
</xs : schema>
```

At first glance this structure looks complex, but in fact it is based on a simple system of hierarchically nested elements, complete with attributes. A schema is equivalent to a tree diagram in linguistics: it captures structural generalisations but does not, except by implication, detail a particular instantiation of an object. However, in our computational model it is necessary to provide specific utterance instantiations. These are formatted as an XML object which must be validated against the corresponding XML-schema. So, for example, the XML characterisation of a PARTICULAR syllable must be tested against the XML-schema characterisation of syllables IN GENERAL and shown to be valid. The XML code always points to its validating XML-schema (see line 2 in the following code, where the schema *syllable.xsd* is referenced). The first line of the code has to name the XML version (in fact the only version currently available) as well as the basis of the character encoding.

An instantiation (equivalent to a derivation in linguistics) deriving from the general case is easier to follow than its corresponding XML-schema. The example below is the coding for the mono-syllabic word *streets* /striːts/. The first line declares the version of XML being used, and the second line indicates that the instance conforms to the general syllable XML-schema (called *syllable.xsd*) which we saw above.

```
<?xmlversion = "1.0"encoding = "UTF – 16"?>
<syllableSchemaLocation = "./syllable.xsd">
   <onset>
      <consonant> str </consonant>
   </onset>
   <rhyme>
      <nucleus>
         <vowel> iː </vowel>
      </nucleus>
      <coda>
          <consonant> ts </consonant>
      </coda>
   </rhyme>
</syllable>
```

where <element> means ‘start of an element’, and </element> means ‘end of an element’ – these can be collapsed to <element/> (= ‘there is an element which starts and ends here’) if necessary.

In this XML model of the syllable /striːts/ we find the three-consonant sequence /str/ identified as the onset, the vowel /iː/ identified as the nucleus within the rhyme. The two-consonant sequence /ts/ is identified as the coda, also within the rhyme. The nucleus and coda stand in a logical AND relationship – that is the coda logically follows the nucleus and must exist (even if it includes zero consonants).

In procedural pseudo-code the instantiation would look like this:

```
syllable : string;
onset : string;
rhyme : string;
coda : string;
nucleus : string;
  {
        syllable = onset + rhyme;
        rhyme = nucleus + coda;
        if syllable = ‘streets’ then
          {
            onset = ‘str’;
            nucleus = ‘iː’;
            coda = ‘ts’;
          }
  }
```

However, procedural coding implies a PERFORMANCE act of instantiation rather than a DERIVATION based on the declared grammar. The two are not equivalent. The distinction is clearly made in linguistics where the GRAMMAR (the declaration) is contrasted with a

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

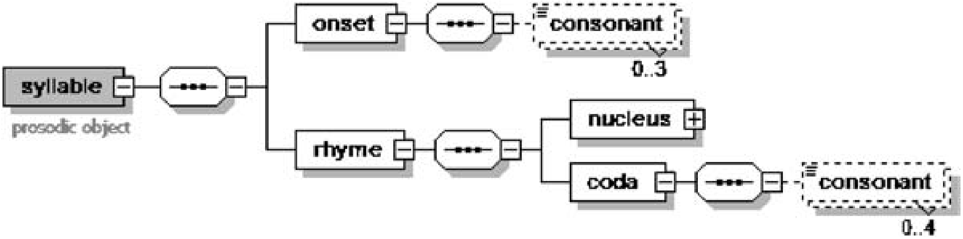


Figure 2 Tree diagram of *syllable.xsd*. The dotted consonant elements indicate that they include a zero option as well as a sequence of up to 3 or 4. The nodes linking elements indicate SEQUENCE rather than CHOICE – that is, the descendent elements are in a logical AND relationship rather than an OR relationship. Thus if we have an onset descendent from element *syllable*, we must also have a rhyme element. Below each consonant element is an annotation indicating the minimum and maximum number of occurrences of the consonant object.

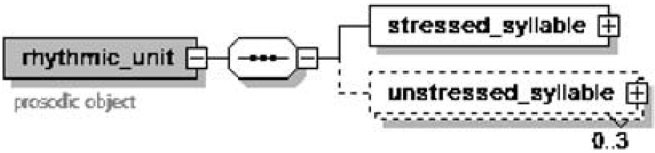


Figure 3 Collapsed tree diagram of *rhythmic_unit.xsd*. The immediate descendents of the rhythmic_unit element are, in sequence, a stressed syllable followed by zero or up to three unstressed syllables (the occurrence of 4 or more unstressed syllables is rare). The unstressed syllable is dotted to indicate that it may not be present.

DERIVATION (an idealised exemplar instantiation of the grammar); a performance act is an instantiation on a particular occasion and is not necessarily idealised. We give examples of procedural pseudo-coding elsewhere in this paper, but only for the sake of comparison with the procedural paradigm – this code does not form part of our declarative model of data structures in speech production theory.

Although one purpose of the XML-schema is to capture the overall general structure of a syllable, it is also used to validate any proposed instantiation as a check on conformity. A validation parse of the model for /striːts/ shows it conforms to the XML-schema *syllable.xsd*. The parse is performed by traversing the tree structure of the XML declaration and checking that each node is a valid instantiation of the general case. At the lowest level, what constitutes a valid consonant, valid vowel, and valid sequences within the elements onset, nucleus and coda are defined externally to this schema, and are givens.

In the overall computational model of speech production the validating parse is important because it enables identification of the various nodes in an instantiation and establishes their relationship with each other – that is, it characterises their context. This in turn enables subsequent processing to operate properly on the correct object. As a very simple example, in our monosyllabic word *streets* we may need to render phonetically the onset /t/ quite differently from the coda /t/; this would reflect their position or context within the syllable. In turn the position of the syllable within the wider prosodic structure of the entire utterance would also enter into the detailed rendering of the various segments within the syllable. We shall see later that the dominant framework entering into the detail of phonetic rendering of the entire utterance is its prosodic structure.

It is useful to view an XML-schema graphically, and this is the format used from now on in this article. Figure 2 is the tree diagram associated with *syllable.xsd*, while figure 3 shows the tree diagram associated with *rhythmic_unit.xsd* – the XML-schema for rhythmic units. To avoid confusion over units of rhythm we do not use the term FOOT here. Elsewhere (Tatham

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

Table 1 Results for durations in ms for 267 rhythmic units.

Rhythmic unit	Mean	Median	Standard deviation	Lowest value	Highest value	Count	Coefficient of variation
1 syllable	354.5	366	111.5	177	673	63	31.5
2 syllables	436.7	432	125.7	183	768	119	28.8
3 syllables	497.3	487.5	110.4	267	781	74	22.2
4 syllables	594	590	69.2	480	702	11	11.6

& Morton 2001) we distinguish between the traditional term FOOT (which indicates a general abstract unit of rhythm to which listeners and speakers are sensitive) and the term RHYTHMIC UNIT to indicate a quantifiable instantiation of FOOT. In this paper we shall use only RHYTHMIC UNIT.

In the prosodic hierarchy rhythmic units dominate syllables and impose constraints on their occurrence. They are basically trochee-like in character, and in our case are defined as SYLLABIC trochees ($\sigma_{\text{stressed}} + \sigma_{\text{unstressed}}$), though Hayes (1995: 89ff. and elsewhere) and others argue in favour of MORAIC trochees – but see Hammond (1999) defending English as basically syllabic in nature; this is not an issue in this paper.

Figure 4 shows the expanded tree for *rhythmic_unit.xsd* – right down to the terminal elements: vowels and consonants. Each syllable is structured according to the syllable schema *syllable.xsd* and descendent to the rhythmic_unit.

A predictive procedure using the basic rhythm unit

In a recent experiment (Tatham & Morton 2002) we investigated the timing of rhythm units in a small database of read speech. Most rhythmic units in the data had two syllables (stressed + unstressed), and the mean duration for this type was 437 ms. Other rhythmic units in the data had from one to four syllables, each beginning with a stressed syllable and followed by up to three unstressed syllables according to the data structure validated by *rhythmic_unit.xsd*. The durations are shown in table 1.

Using this finding as our starting point we are now in a position to begin building a simple predictive procedural model of rhythmic unit duration as an example of how data structures enter into the procedural 'flow' of the model. A predictive model is one which is derived from some experimental data, but which generalises the results to predict what will be measured in some new data. It should be borne in mind, though, that the focus of this present paper is the data structures themselves, rather than procedures that they might be involved in. It seems to us that this focus is a prerequisite to building a fully interpretive model. XML data structure declarations are validated by their associated schema and used in the model in a way analogous to the built-in procedures of a text-layout browser for interpreting HTML code or those to be found often in associated CSS (cascading style sheet) files. The procedure illustrated here constitutes a tiny fragment of the speech production interpreter – the device responsible for taking the validated XML data structures and making them work in a real speech production environment.

We use this stressed + unstressed unit type as the basic rhythm unit. Our model focuses on rhythm unit ratios, and calculates the following rhythm unit durations from the starting point of a basic rhythm unit to which is assigned a value *L*:

```
basic_rhythm_unit = L;
        {
        if one_syllable_unit then L = L − (L*20/100);
        if two_syllable_unit then L = L;
        if three_syllable_unit then L = L + (L*15/100);
        if four_syllable_unit then L = L + (L*35/100);
        }
```

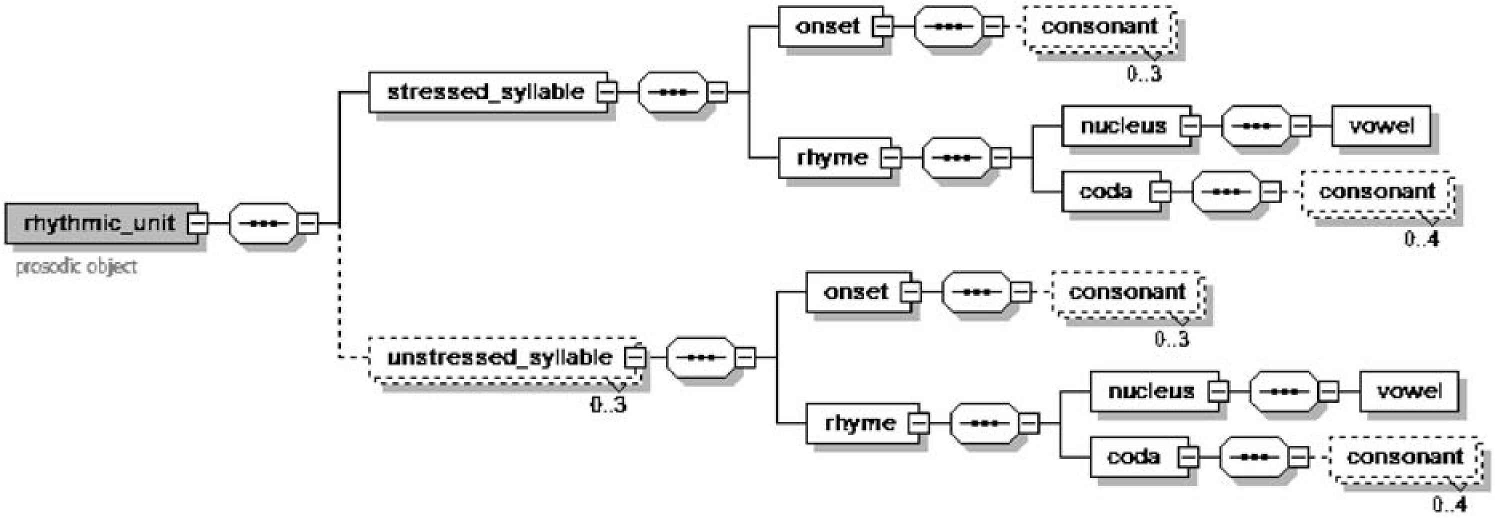


Figure 4 Expanded tree diagram of *rhythmic_unit.xsd*. The sequence 'stressed syllable followed by zero to 3 unstressed syllables' is expanded to include the data structure associated with syllables. Elements that can have a zero presence are enclosed in a dotted box.

That is, the ratio is:

81.2 : 100 : 113.9 : 136.1

or, simplified:

80 : 100 : 115 : 135

We use these formulae, which describe not only the unit lengths, but also their relationship to each other, as the basis for the predictive model. The model generalises by using ratios rather than absolute units.

As part of the research reported in the article cited we note that predictions assigning a value to L different from that in the original data tend to be less reliable in correlation with how much the new value differs. This is entirely to be expected, and simply highlights the need to investigate what other factors influence rhythm during rates of delivery differing from the norm.

Speech production is a system that operates predictably within certain limits and becomes less predictable and less stable the more those limits are approached. Part of the research area called PHONETICS is determining just what these limits are. The same is true of the perceptual system. It, too, appears fairly stable within certain limits. The major function of the Cognitive Phonetics Agent in our model is either to pull the system back within these limits when it seems to be straying outside them, or to use what it knows of the possibilities of speaker/listener collaboration to stretch these limits. This is partly why we call the human speech production system DYNAMIC, and certainly why we call it INTELLIGENT and REASONING. Earlier ideas dwelt too much on automaticity in phonetics (Chomsky & Halle 1968: chapter 7). The terms ‘intelligent’ and ‘reasoning’ have been carefully chosen to reflect active cognitive processes, and have nothing to do here with another field of study, Artificial Intelligence. The model conforms strictly to the theory of human speech production.

The prosodic framework for utterances

Utterance plans in our computational model are derived within an overall prosodic framework which itself is declared in XML. Syllables are the basic units of prosody and, as we saw from figures 2 and 3, they relate sequentially *via* a dominant rhythmic unit. In turn rhythmic units (of which there must be at least one) form a sequence within an accent group, and accent groups form a sequence within intonational phrases.

The overall framework is a prosodic one since the phonetic rendering of utterances depends on their prosodic structure, including the detail of the structure of the syllable – the basic unit of the prosody. The idea is not novel (see Firth 1948 on the prosodic structure, and Kahn 1976 and Gussenhoven 1986 on the structure of syllables, in particular the phenomenon of ambisyllabicity and detailed phonetic rendering). In earlier publications (Lewis & Tatham 1991, Morton & Tatham 1995) we reported our adoption of a prosodic framework within which some of the segmental properties of speech could be modelled (cf. also a similar framework in Keating & Shattuck-Hufnagel 2002). One example of a positive gain was that the prosodic properties of a declarative utterance, like the rallentando effect usually occurring toward the end, predict segmental effects – for example, weaker coarticulatory formant bending.

In a traditional notation the most general case of the prosodic framework is:

$$IP \rightarrow AG\ (AG\ \ldots) \rightarrow (\ldots\rho)\ \rho\ (\rho\ \ldots) \rightarrow \sigma\ (\sigma\ \ldots) \rightarrow (o)\ r \rightarrow n\ (c)$$

> where IP = intonational phrase, AG = accent group, ρ = rhythmic unit, σ = syllable. Syllables take the traditional *onset* + *rhyme (→ nucleus* + *coda)* form. Optional elements are bracketed.

Figure 5 is a graphic representation of the XML-schema *prosodic_framework.xsd* down to the syllable structure elements. The initial dominant element is the intonational phrase, below which there must be at least one accent group (with the upper limit open). In turn the

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

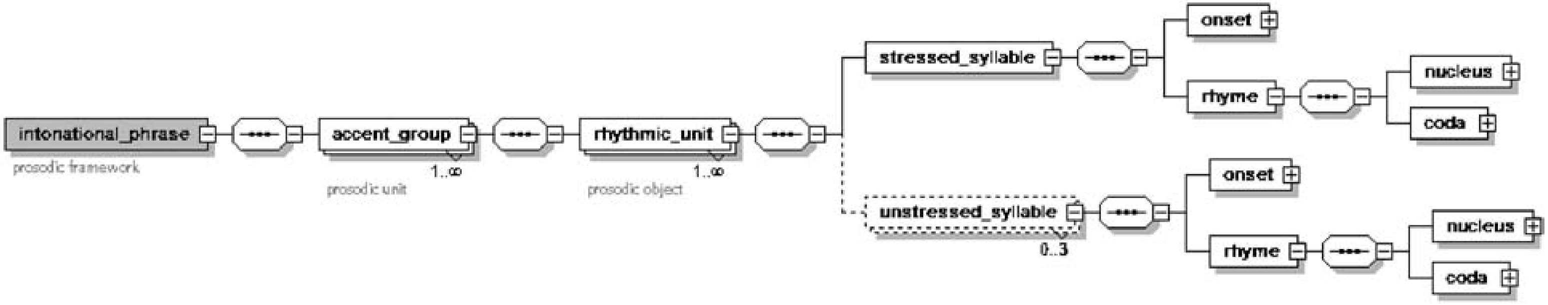


Figure 5 The prosodic framework modelled as an XML-schema. The figure shows the graphical representation of the schema down to the general syllable structure elements of the tree.

accent group dominates at least one rhythmic unit. As before any XML instantiation of this general case schema can be validated against the schema and appropriately parsed for further processing.

Utterances are planned WITHIN this prosodic framework; thus prosody is not something which is added to utterances at some lower level. The reason for this is that detail of actual utterances is dominated by prosodic structure (e.g. the effects of stress and rhythm on vowel quality, the effects of ambisyllabicity on plosive release (Ogden et al. 2000)), and interactions between elements within the structure. We give some examples later of prosodically dominated processes operating on segmental elements (see the section 'Phonetic rendering on the dynamic plane's phonetic tier'). Meanwhile, let us continue with some more examples of data structures within the computational speech production model.

Below syllable level – segment gesture data structures

The word SEGMENT on its own is used in the model for elements at the phonological level; at the phonetic level we speak of SEGMENT GESTURES. In recent times the term GESTURE has been associated with the Articulatory Phonology speech production model (Browman and Goldstein 1986) but the term had been used extensively by previous writers (cf. its use in Paget 1930 and Abercrombie 1967).

The model adopts the established classification of two types of segment gesture: consonant and vowel (Ladefoged 2001). We express the data structures for these types in terms of the object-oriented paradigm. This particular paradigm has been chosen because the data structure associated with segment gestures is essentially flat – that is, it does not have the deep hierarchical composition of the earlier prosodic framework examples.

In Tatham & Morton (1988) we discuss various computational approaches to data structures and data manipulation in speech production modelling, and conclude that an object-oriented representation is particularly appropriate for speech gestures. An essential feature of the object-oriented approach is that properties and procedures (called 'methods') are attached to individual objects. Thus, if objects are grouped according to type or class, these properties will be transparently seen to be shared without separate repetition for each individual object. A procedural approach could achieve the same objective using explicit linking, but class membership and property inheritance are more elegantly (and practically) captured in the object-oriented paradigm.

Segment gestures are objects with a general structure identifying a number of parameters and indicating their computational type. The parameters specify phonetic articulatory goals and use a terminology which is for the most part familiar in articulatory phonetics. The computational types are markers for the appropriate attributes of each goal. Notice that many parameters are doubled up; thus, we have 'place1_oral' and 'place2_oral'. This is to allow for specifying the structure of BI-PHASAL and BI-POLAR gestures. A bi-phasal gesture is one which has a sequence of two constriction goals (e.g. [t] with a stop phase followed by a release phase), and a bi-polar gesture is one with two place goals (e.g. [ai] – a diphthong with different start and end points in the vowel space). The robustness parameter is an indicator of how resistant the gesture is to externally derived factors such as coarticulation. It is arguable that this choice of parameters is an oversimplification of the phonetic facts – but we stress we are dealing with a first approximation model focusing on data structure: detailed representation comes later.

The general case for a consonant gesture of 14 parameters is:

```
consonant_gesture = object
   type:                string;
   robustness:          real;
   place1_non_oral:   string;
   place2_non_oral:   string;
   place1_oral:         real;
   place2_oral:         real;
   place1_extraoral:    string;
   place2_extraoral:    string;
   constriction1:       real;
   constriction2:       real;
   round:               real;
   nasal:               real;
   glottis1:            real;
   glottis2:            real;
end;
```

And a vowel gesture is an object with the following general structure of 10 parameters:

```
vowel_gesture = object
   type :               string;
   robustness :         real;
   place1_oral :        real;
   place2_oral :        real;
   constriction1 :      real;
   constriction2 :      real;
   round :              real;
   nasal :              real;
   glottis1 :           real;
   glottis2 :           real;
end;
```

where 'real' is a real number, either 0 to +1, or −1 through 0 to −1. String is a text string like *lips*, etc. These are data types in the traditional sense.

Vowel gestures can have only an oral place (that is, their traditional front/back place must be in the vowel space). But consonant gestures can also have a non-oral or 'extra-oral' place: lips, lips/teeth, teeth, velum, pharynx, glottis. We find it useful to identify those CONSONANT gestures that have a VOWEL-space affinity – we feel that vectors like: [t] → [s] → [ʃ] → [i] → [ɛ] → [æ] are important conceptually although it appears that we have switched from consonant to vowel midway. In traditional terms the start place for this vector is the alveolar ridge, a value of 0.8 (see table 3 below for more detail on the meanings of these values). It is helpful to think of the vector as a line on a traditional vowel chart moving from the alveolar ridge down to the spot used to mark [æ]. This line passes through all the segments in the sequence, which could be regarded as zones on the vector, or categories through which the vector moves.

The segment gestures of English on this vector all share place 0.8 and have, respectively, constrictions with values: 1 → 0.8 → 0.7 → 0.5 → 0.3 → 0.1. The constriction 1 is important because it actually signifies 'beyond the palate'; the use of the concept BEYOND guarantees the required contact pressure for the plosive, whereas 0.9 which is for taps signifies 'AT the palate' with insufficient contact pressure to hold back the air stream other than momentarily – a condition never tested because the contact time is always too small (except for rolls where the contact pressure is low and the time is long). A bilabial roll, for example, would be 'type: bi-polar (place focus), place1: lips, place2: lips, constriction1: 0.9, constriction2: 0.9', whereas a bilabial stop would be 'type: bi-phasal (constriction focus), place1: lips, place2: lips, constriction1: 1, constriction2: 0', and a bilabial tap would be 'type: bi-phasal (constriction focus), place1: lips, place2: lips, constriction1: 0.9, constriction2: 0'.

These general data structures are declared on the phonetic tier of the static plane. This is where they are also instantiated to give us the general set of segments available for the language. Here is an example of two default instantiated consonant objects, [t] and [r] before being moved to the dynamic plane:

```
var
   [t]: consonant_gesture;

[t].type             = bi-phasal
[t].robustness       = 0.5
[t].place1_oral      = 0.8
[t].place2_oral      = 0.8
[t].constriction1    = 1
[t].constriction2    = 0
[t].round            = 0
[t].nasal            = 0
[t].glottis1         = 0
[t].glottis2         = 0
```

where [t] is an object of type consonant, thereby inheriting the set of variables defined as being those which characterise a consonant object: viz. {type, robustness, place1_oral, place2_oral, constriction1, constriction2, round, nasal, glottis1, glottis2}, and these variables are instantiated for [t] as listed. The variables for consonant objects which do not appear here are not applicable to this particular segment, but are present (with no value) for the sake of completeness and to indicate their applicability to OTHER segments. The '0' is the indicator. Note that the default instantiation of [t] is NOT aspirated (the two phases are stop + release); aspiration is the delayed onset of vocal cord vibration in a following vowel and not a property of [t].

```
var
   [r]: consonant_gesture;

[r].type             = bi-phasal
[r].robustness       = 0.9
[r].place1_oral      = 0.5
[r].place2_oral      = 0.6
[r].constriction1    = 0.2
[r].constriction2    = 0.3
[r].round            = 0.1
[r].nasal            = 0
[r].glottis1         = 0.8
[r].glottis2         = 0.8
```

where [r] is an object of type consonant, inheriting the characteristics of a consonant object, and these are instantiated as shown for [r]. As before irrelevant parameters are not listed. [r] is of type bi-phasal because there is constriction change during the segment.

And similarly, two vowels: [ɒ] and [aɪ]:

```
var
   [ɒ]: vowel_gesture;

[ɒ].type             = uni-polar
[ɒ].robustness       = 0.9
[ɒ].place1_oral      = 0.1
[ɒ].constriction1    = 0.2
[ɒ].round            = 0.2
[ɒ].nasal            = 0
[ɒ].glottis1         = 0.9
```

where [ɒ] is an object of type vowel, inheriting vowel prototype parameters, properly instantiated here for [ɒ]. Notice that short monophthongs (as in this example) are of type uni-polar, but long monothongs are of type bi-polar allowing for characterisation of their almost universal tendency to diphthongise slightly.

```
var
   [aɪ] : vowel_gesture;

[aɪ].type            = bi-polar
[aɪ].robustness      = 0.9
[aɪ].place1_oral     = 0.5
[aɪ].place2_oral     = 0.7
[aɪ].constriction1   = 0.1
[aɪ].constriction2   = 0.4
[aɪ].round           = 0.1
[aɪ].nasal           = 0
[aɪ].glottis1        = 0.9
[aɪ].glottis2        = 0.9
```

where [aɪ] is an instantiation of the vowel prototype with parameters appropriately assigned. Diphthongs are of type bi-polar to enable place shift to be described. ‘Vowelness’, that is, what constitutes a vowel, is inherited from the general case characterisation, with appropriate values assigned for this particular vowel.

Note that the parameters and values we have used here, and throughout this paper, are no more than hypotheses; the main point is to illustrate the framework itself. Experimental evidence may point the way toward additional parameters or extensions of the ones we suggest, for example, the inclusion of a tri-polar type of vowel to account for sounds which some researchers have classified as triphthongs. Our suggestions are based on English, but clearly what happens in other languages will alter the detail of the data structures.

Objects also have methods – procedures which the object embodies. That is, these static descriptions just discussed are enhanced with functional information that the object knows about its own behaviour. We do not discuss this property of phonetic or phonological objects here except to say that the model allows for segment gesture behaviour to originate from within the gesture as well as to be determined from outside. An example would be a function which picked up a low value constriction requirement (say, tongue very low) and brought in jaw dropping to assist tongue movement – thus the object would know that tongue lowering and jaw dropping are in a functional relationship with each other. All object oriented systems have this property – a major characteristic distinguishing them from simple procedural systems. We originally applied the object oriented paradigm to a computational characterisation of Action Theory’s coordinative structures (Fowler 1980) because the coordinative structure model fitted the paradigm remarkably well (Tatham & Morton 1988).

Additionally the object oriented paradigm permits INHERITANCE in a similar way to the declarative XML paradigm used here to characterise the prosodic framework for speech production. Properties associated with a higher node or parent declaration are inherited by lower nodes or child declarations. For example, an instantiation of *consonant_gesture*, say [t], is said to inherit its parent properties, that is, all those assigned to the dominant general case, *consonant_gesture*.

5 The Cognitive Phonetic Agent and phonetic rendering

The Cognitive Phonetic Agent (CPA) works to supervise phonetic rendering, making sure that the output of the reasoned rendering processes is optimal. To do this the CPA needs various pieces of information to decide what constitutes an optimal rendering and how to achieve it on any one occasion.

An optimal rendering is one which achieves the goal of promoting a good percept in the listener’s mind. The perceptual system is such that there is not just a single rendering which is optimal. There is a range of renderings all of which can be accommodated by the listener, by a process of ‘repair’, in arriving at the right percept. The range describes a bell shaped curve with the effectiveness of repair lessening toward its edges. The CPA is aware of this because it understands how the repair process works and what its limitations are – IT INCORPORATES A MODEL OF THE REPAIR PROCESS.

The essential property of rendering we need to focus on is that it is an active dynamic process which brings additional information and data to developing an articulation from the basic utterance plan. The rendering process has more than the phonological plan as its input. In addition there is a supervisory input bringing considerations of expressive content to the rendering process.

The term ‘render’ is derived from the field of computer graphics in which a rendering process takes a simple wire frame model of a 3D object and paints it with colour and texture, and provides an illuminating light source together with appropriate shadows. The rendered object derives from the basic wire frame plan by the addition of graphical ‘expression’ involving interpretation of the plan in the light of such expressive demands. We use the term ‘render’ in a similar fashion: a basic utterance plan is rendered with spoken expression to derive an articulation from which the original plan can be perceived but which also triggers in the listener perceptual correlates of the added expressive content. ‘Render’ is also a term used in the theatre; scripts are rendered in a particular way by actors.

6 An example derivation

In this example derivation we trace a short utterance through events on the dynamic plane of the speech production model. The DYNAMIC plane is where CPA-driven algorithmic processes can occur, and this contrasts with the STATIC plane which is reserved for groupings of simple descriptive processes akin to expected descriptions in the usual phonological and phonetic components of a grammar.

The model begins with an entry point to the speech production algorithm. Here the unique future utterance enters the system in the form of a REQUIREMENT UTTERANCE – an object to be spoken. There are four main procedures on the dynamic plane, shared between the prosodic/phonological tier (concerned with formulating the utterance plan) and the phonetic tier (concerned with rendering the plan):

```
{
   input (requirement_utterance);
   formulate_plan (requirement_utterance);
   render_plan (requirement_utterance);
   output (requirement_utterance);
}
```

The four main actions to be performed on the utterance specify that

- it must be input into the speech production algorithm [phonological tier],
- a plan for speaking it has to be formulated [phonological tier],
- the plan has to be rendered [phonetic tier], and
- the result has to be output [phonetic tier].

The required utterance which is input to the speech production dynamic plane (phonological tier) originates higher up in the system – it is equivalent to a string written down and which has to be spoken out aloud; in itself it has no sound shape other than a very minimal representation of some underlying phonological properties, just sufficient to enable the subsequent planning and rendering procedures which are part of the speech production process we are modelling to be performed.

So what does the required utterance look like? We declare the structure of an utterance again using XML since what we need is a hierarchically structured declaration to reflect the composition of the data structure. A specific utterance representation takes the form of an

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

XML structure, but the general representation of all utterances takes the form of an XML-schema. Thus there would be a file called *utterance.xsd* specifying what any utterance must look like, and several files of which one would be a particular utterance called *utterance.xml* where *utterance* is the name of the actual utterance.

Here is a sample utterance:

<sentence> bʌt wɒt s ðə fʊl prais </sentence>   *[But what's the full price?]*

The highest level in the hierarchical description of this utterance declares the sentence domain. Other syntactic marking is present as necessary for subsequent phonological marking or analysis; though this is omitted in the example for the sake of clarity and because it is not central to the discussion here. The utterance is immediately assigned the abstract prosodic framework (see section 4 above, 'The prosodic framework for utterances'), which forms the basis, or WRAPPER, for all subsequent phonological and phonetic processing down to the level of motor control. An overview of phonological encoding and what the process might mean (movement from morphemic to phonological representation) is discussed by Keating (2000).

Assignment to the abstract prosodic framework

In its simplest most general form the abstract prosodic framework looks like this:

<prosody><utterance><IP><AG/+></IP></utterance></prosody>

This means that on this occasion the utterance contained within the prosodic framework consists of an intonational phrase (IP) within which there is at least one accent group (AG). The notation used here for the accent group tag uses the '+' to indicate that there can be more than one AG, and the '/' following the element's name indicates that when instantiated the element will open and close containing other elements. For example <AG/> can expand to

<AG><rhythmic_unit/></AG>.

But the data structure has to have elements assigned within it as part of its instantiation to a particular utterance. Intonational phrases and accent groups are assigned to utterances in the usual way (Ogden et al. 2000). In this abstract representation of an utterance stress levels (of which we have for the purposes of this exposition just two: stressed and unstressed) are either pre-assigned to lexical items or procedurally assigned where patterning is regular. Syllable based rhythmic units are also assigned procedurally as shown in this simple pseudo-code:

```
procedure insert_rhythmic_unit_boundaries;
  {
    for utterance = 1 to i
         for syllable = 1 to j
             if syllable = stressed then
                insert rhythm_unit_boundary before syllable;
         next syllable;
    next utterance;
  }
```

Below is the highest level declaration of our example sentence *But what's the full price?* It has been assigned to the ABSTRACT prosodic framework. The framework for the utterance is dominated by <IP/>, and this forms the widest domain and container for the remaining prosodic units. This utterance consists of three sequenced (AND-ed) accent groups, the first of which has one (abstract) rhythmic unit with two syllables (one stressed, one unstressed), the second a rhythmic unit with two syllables (one stressed, one unstressed), and the third two rhythmic units each with a single stressed syllable. The notation is explained below the declaration.

```
<utterance>
      <IP>
         <AG>
            <rhythmic_unit>
               <syllable stressed="1"> $ </syllable>
               <syllable stressed="0"> bʌt </syllable>
            </rhythmic_unit>
         </AG>
         <AG>
            <rhythmic_unit>
               <syllable stressed="1"> wɒt s </syllable>
               <syllable stressed = "0"> ðə </syllable>
            </rhythmic_unit>
         </AG>
         <AG>
            <rhythmic_unit>
               <syllable stressed="2"> fʊl </syllable>
            </rhythmic_unit>
            <rhythmic_unit>
               <syllable stressed="1"> praɪs </syllable>
            </rhythmic_unit>
         </AG>
      </IP>
</utterance>
```

where

<IP/> is an intonational phrase: the domain of an intonation contour
<AG/> is an accent group: an intonation unit
<rhythmic_unit/> is a rhythmic unit: an abstract unit of rhythm
<syllable> is a syllable – the lowest coherent unit (node) of prosody, and in this system, the lowest coherent unit of prosodic phonology – that is, phonology within a prosodic framework
<syllable stressed="1"> is a stressed syllable, <stressed="1"> is an attribute of <syllable>
<syllable stressed="0"> is an unstressed syllable
<syllable stressed="2"> is an nuclear stressed syllable
$ is an empty stressed syllable to cater for rhythmic units at the start of an utterance with an apparently missing stressed syllable (Tatham and Morton 2001 and 2002), where hanging syllables in rhythmic structure are explained and discussed. The condition is that each rhythmic unit must start with a stressed syllable; if it does not it is necessary to insert an empty stressed syllable).

The sample derivation is not intended to show SYLLABIFICATION in action; the procedures which operate in the syllabification process reside on the static plane. It would complicate our illustration to include these procedures, but their description would provide the following XML code:

```
<syllable stressed = "0">
         <onset> b </onset>
         <rhyme>
            <nucleus> ʌ </nucleus>
```

```
        <coda> t </coda>
      </rhyme>
</syllable>

<syllable stressed = "1">
    <onset> w </onset>
      <rhyme>
        <nucleus> ɒ </nucleus>
        <coda> ʦ </coda>
      </rhyme>
</syllable>

<syllable stressed = "0">
    <onset> ð </onset>
      <rhyme>
        <nucleus> ə </nucleus>
        <coda> 0 </coda>
      </rhyme>
</syllable>

<syllable stressed = "2">
    <onset> f </onset>
      <rhyme>
        <nucleus> ʊ </nucleus>
        <coda> l </coda>
      </rhyme>
</syllable>

<syllable stressed = "1">
    <onset> pr </onset>
      <rhyme>
        <nucleus> aɪ </nucleus>
        <coda> s </coda>
      </rhyme>
</syllable>
```

Phonotactic constraints mentioned earlier under the declarative description of a general syllable in English operate to constrain segment sequences in the onset, nucleus and coda elements to prevent sequences such as [mr. . .], [psl. . .] or [. . . tnft].

IP (the intonational phrase) is the widest intonational domain treated here, and may often correspond to the syntactic domain SENTENCE. Within the IP domain are accent group (AG) sub-domains – the domains of pitch accents. Notice though that there are no fully predictable IP or AG instantiation types (particular intonation contours) from the basic syntactic structure of utterances, except on a statistical basis. Rather, prediction comes from the yet larger framework of EXPRESSION, and takes in communicative aspects of language extending in principle beyond the utterance. However, for the moment, let us just indicate this as

<EXPRESSION><utterance><IP><AG/+></IP></utterance></EXPRESSION>,

where <EXPRESSION/> might be instantiated, for example, as <*tactful/*> or <*forthright/*> or some similar expression declaration. It seems reasonable to us to assume that expression would have the major influence (*via* the CPA) on the intonation contour type to be associated with how this particular utterance is to be spoken. Because expression seems to pervade all nodal processes of the utterance it is right that it should be located on a higher wrapper node. Expression as a ‘way of talking’ often changes during a communicative exchange, and our data structure must and does take care of this – enabling the node’s varying content to influence intonation contour type and moment of change appearing lower in the structure. (There is no space here to argue the case for having the node <EXPRESSION/> dominating the utterance; the term was however explained at the beginning of this paper. Let us just say that for the moment this arrangement seems to us to account for the data more satisfactorily than other arrangements. A single expression type usually spreads, for example, over one or more utterances rather than simply over part of an utterance.)

Ambisyllabicity

As discussed earlier <syllable/> is itself hierarchically organised into units of <onset/> and <rhyme/>, with <rhyme/> organised as <nucleus/> and <coda/>. Note that there is scope for the <coda/> content of one syllable spanning the <onset/> content of a following syllable – the phenomenon usually called AMBISYLLABICITY. When the phenomenon does occur it is important in determining some of the detail in subsequent rendering processes, for example, whether a voiceless plosive is followed by aspiration or not. However, Huckvale (1999) has noted a problem with representing ambisyllabicity in XML because the strict component hierarchy constraint in XML syntax forces element duplication by preventing membership of two branches of the tree by a single leaf – the very meaning of ambisyllabicity. We do not feel this to be a particularly serious constraint because element duplication is sometimes felt and reported by native speakers. It is true, though, that the phenomenon, called by us ELEMENT SPANNING, is badly catered for in XML.

Ambisyllabicity (Kahn 1976, Gussenhoven 1986) occurs when the coda of a syllable and the onset of the following syllable overlap in the sense that there is no real certainty as to which syllable a particular element belongs. Take a polysyllabic word like *maker*. There is evidence of native speaker hesitation here when asked about syllable boundaries. Speakers can say that the word consists of two syllables, but some will hesitate when asked where the boundary is: some will report a structure like /meɪ.kə/ and others will say /meɪk.ə/. A linguist might say that /meɪ.kə/ is essentially a phonologically motivated structural description, while /meɪk.ə/ is a morphologically motivated description. Ambisyllabicity as a notion in phonology captures this ambivalence of the /k/ by assigning it to BOTH the coda of the first syllable and to the onset of the second – this is where the strict component hierarchy constraint is violated. Ambisyllabicity is the default solution for this kind of data, and holds so long as constraints on syllable structure are not violated – for example, *makeshift* can only be /meɪk.ʃɪft/ – the sequence /...kʃ.../ is not permitted in either a coda or an onset.

In our model we assign an attribute *span* to the element *coda*, where span is either true or untrue (0 or 1 respectively). If span is true then the consonant or consonant cluster in the coda can also be the onset of the next syllable in an ambisyllabic arrangement. Researchers are unclear at to the DIRECTION of ambisyllabicity. By this we mean that we have chosen to say that a coda consonant, consonant cluster or partial consonant cluster can span to a following onset (a left to right direction), but is it the case that we might equally have spoken of an onset consonant spanning to the previous coda (a right to left direction)?

Furthermore we feel that the right place to assign span is on the coda itself rather than on the lower consonant – our feeling is that it is the coda which overlaps rather than the lower consonant node. We keep this idea even when only part of a consonant cluster may be ambisyllabic, as in *selfish*, for example. We shall see later that ambisyllabicity has consequences for the rendering process where syllabic boundaries might be aligned 'to best predict allophonic variation' (Gimson, revised Cruttenden 2001: 52). Gimson also uses phonological allophonic rules to assign syllable boundaries. So, for example, a word like *metal* would be /mɛt.əl/ or /mɛt.təl/, rather than /mɛ.təl/ to satisfy the observations that the word is pronounced in some accents as /mɛʔ.əl/ (/mɛ.ʔəl/ is not possible), that monosyllabic words cannot end in /ɛ/, and that this is a vowel which is shortened before a voiceless consonant in the same syllable.

We characterise this particular word as having its first syllable ending in a spanning coda, that is, as ending in an ambisyllabic coda. Thus we modify the basic syllable model to assign to the element <coda/> an attribute *span* which can take a Boolean value of 0 or 1: <coda span=(0 | 1)/> (meaning either 'this is a coda with no ambisyllabic element', or 'this is a coda with an ambisyllabic element').

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

Phonological rules used on the dynamic plane

Once the utterance has been given a prosodic envelope it is possible to proceed through the phonological rules. Tiers on the dynamic plane are like blackboards under the control of the devices we see as reasoning agents. We have already mentioned the CPA – the Cognitive Phonetics Agent. The prosodic/phonological reasoning agent equivalent to the CPA has two main functions:

- first, to scan the STATIC prosodic/phonological tier to locate rules which fit the prosodic and phonological context descriptors of the utterance; and then to import such rules and apply them – this whole process to be performed iteratively until no context descriptors remain;
- to invoke its supervisory capacity to manage the whole procedure, in order to use the prosodic/phonological processes appropriately, depending on pragmatic (Morton 1992) and other inputs, provided these constrain phonological processes.

So, the reasoning supervisor agent selects appropriate processes to apply to the utterance requirement. The procedure is not straightforward because there are additional inputs specifying conditions like attitude and emotion. Thus it is unlikely that two utterance plans for a single utterance requirement would be the same. The reason for this is that at this point the supervisor agent exercises the role of systematically managing variations that are collectively perceived as ‘expression’. We work on the assumption that no speech is without expression, and that expression is a continuous variable. This paper is not about modelling the detail of expressive content in speech, but it is often reported that underlying categories of expression, like ‘angry’, ‘happy’, etc., are instantiated in the speech waveform in a continuous way (Tatham & Morton 2003 in press).

A straightforward planning process would result in an expressionless plan – and, if it is the case that plans are never expressionless, a means must be established for interacting with the planning process to derive plans with expressive content. In our model the supervisor is responsible for varying the plan to include expression. The supervisor knows what expression is needed by carefully weighing up competing inputs sourced in pragmatics (Morton 1992, Morton & Tatham 1995), stylistic and other such tiers and planes. For the moment we are only discussing prosodic/phonological effects – but clearly phonetic rendering also includes variability conveying expressive content (see figure 1 above).

As an example consider that the utterance *But what’s the full price?* is to be spoken with authority. Authoritative style might, among other things, call for a positive release for the /t/ in *what’s*. The speaker we have in mind might normally have an unreleased /t/ in this word, dissolving through affrication into the following /s/. We could argue whether the surface unreleased [t̚] is actually a coarticulated version of the released [t] in this instance (in which case a released [t] would be a cognitive phonetically constrained phenomenon), but we feel that here there is clear choice between released and unreleased /t/ at the phonological level. This is even more obviously the case when the /t/ occurs in true final position: speakers of several accents of English have a clear phonological choice between /wɒt/ and /wɒt̚/, though one form will dominate in some accents for some speakers and the other in other accents. One particular accent at least, Cockney English, will usually substitute /ʔ/ in both final and pre-/s/ positions (Wells 1982), lending weight to the argument that the alternation is phonologically determined since [ʔ] is not a coarticulated allophone of [t]. However this does not mean to say that some measure of coarticulation is not overlaid on phonological choices: for us coarticulation is all-pervasive – it’s a question of teasing out the choice element from the final result and representing extrinsic events of choice consistently on the phonological tier, and intrinsic events on the phonetic tier (Ladefoged 1971, Tatham 1971). Thus, in a more traditional notation:

$$/t/ \Rightarrow \begin{bmatrix} t \\ bi-phasal \end{bmatrix} \Big/ \begin{bmatrix} coda \\ -/s/ \end{bmatrix}$$

That is, voiceless alveolar stops have positive release when in the syllable coda and preceding an /s/ which is also in the coda. We include the rule here for the sake of completeness: in fact, the bi-phasal condition is the default (see table 2 below for an example).

As pseudo-code, the above derivation looks like this:

```
coda : string;
{
    if coda [j] = t and if coda [j + 1] = s
    then t = t_bi-phasal
}
```

We feel, though, that phonetically there is no corresponding positive release for the final [t] in the word [bət̚], so we need to obtain from the static plane and import to the dynamic plane's prosodic/phonological tier a rule like

$$/t/ \Rightarrow \begin{bmatrix} t \\ uni-phasal \end{bmatrix} \Big/ \begin{bmatrix} coda \\ -\# \end{bmatrix}$$

That is, voiceless alveolar stops are unreleased when at the end of a word.

As pseudo-code, the above derivation looks like this:

```
coda : string;
{
    if coda [j] = t and if coda [j + 1] = #
    then t = t_uni-phasal
}
```

So, part of sounding authoritative might involve a CAREFUL style. Suppose the speaker's normal accent is a casual Estuary English, then ordinarily the following rule might apply to the /l/ at the end of the word /fʊl/

$$/l/ \Rightarrow \begin{bmatrix} l \\ +vocalised \end{bmatrix} \Big/ V - \begin{Bmatrix} \# \\ C\# \end{Bmatrix}$$

> where *l* to the left of the arrow is the underlying phonological segment, *V* stands for any underlying vowel and *C* for any consonant; while # stands for word boundary.

That is, an underlying *l* is planned as vocalised 'l', or /ʊ/, in syllable codas either immediately before the boundary or before some other consonant. Examples are *wall*, *melt*.
However a careful style would call for a planning rule

$$/l/ \Rightarrow \begin{bmatrix} l \\ +velarised \end{bmatrix} \Big/ V - \begin{Bmatrix} \# \\ C\# \end{Bmatrix}$$

That is, a velarised ('dark') 'l' or /ɫ/ (sometimes written /l_w/) is to be used in the plan at the end of a word or before final consonants, rather than the vocalised alternative.

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

As pseudo-code, the above derivations collapse to

```
coda : string;
{
   if style = casual then
      {
         if coda [j] = l and if coda [j − 1] = V and
         (if coda [j + 1] = # or if (coda [j + 1] = C and coda, [j + 2] = #))
         then $l = l_{vocalised}$
      }
   if style = careful then
      {
         if coda [j] = l and if coda [j − 1] = V and
         (if coda [j + 1] = # or if (coda [j + 1] = C and coda [j + 2] = #))
         then $l = l_{velarised}$
      }
}
```

For the moment it does not matter whether we regard this as the selection of an alternative rule or as tightening an existing rule, the point is that there has been an informed and supervised act of CHOICE operating. The choice is dependent on considerations peripheral to or outside the usual prosodic/phonological planning processes. Here we are adding a dynamic, intelligent and choice-oriented planning agent to our model of human speech production and to the usual more automatic set of procedures.

But our example authoritative style also involves speaking with more precision (Tatham & Morton 1980) – a matter for phonetic rendering. So far we have a phonological plan looking like this (detail of syllable composition has been omitted here):

```
<utterance E = "authoritative">
   <IP>
      <AG>
         <rhythmic_unit>
            <syllable stressed = "1"> $ </syllable>
            <syllable stressed = "0"> bət </syllable>
         </rhythmic_unit>
      </AG>
      <AG>
         <rhythmic_unit>
            <syllable stressed = "1"> wɒt s  </syllable>
            <syllable stressed = "0"> ðə </syllable>
         </rhythmic_unit>
      </AG>
      <AG>
         <rhythmic_unit>
            <syllable stressed = "2"> fʊɫ ></syllable>
         </rhythmic_unit>
         <rhythmic_unit>
            <syllable stressed = "1"> praɪs </syllable>
         </rhythmic_unit>
      </AG>
   </IP>
</utterance>
```

To summarise: the root *utterance* has now been given an attribute set *E* of which one member is authoritative. The attribute system allows for the names of other expressions to be included, for example: <utterance *E*="happy">. The utterance plan for *But what's the full price?* spoken carefully now has three accent groups within a single intonational phrase. The first two accent groups each contain one rhythmic unit and the third contains two rhythmic units. The syllables

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

are: /.bət̚./ (with reduced vowel and unreleased /t/), /.wɒts./ (with a positively released /t/, /.ðə./ (with reduced vowel), /.fʊɫ./ (with velarised /l/), and /.praɪs./. Tonic or sentence stress falls on /.fʊɫ./ – '.' means 'syllable boundary'. Some choices of surface variant have been determined by the pragmatic consideration that the utterance is to be spoken authoritatively.

Phonetic rendering on the dynamic plane's phonetic tier

Our example sentence *But what's the full price?* began with a very abstract phonological representation which in more traditional terms would have been called phonemic: /$ bʌt wɒt s ðə fʊl praɪs/. The final phonological representation, still within the prosodic framework, is the utterance plan expressed here in what, in the same terminology, would have been called extrinsic allophones: /$ bət̚ | wɒt̚s ðə| ˈfʊɫ | praɪs/ (vertical lines mark rhythmic unit boundaries). A final rendering, still using traditional notation to express intrinsic allophones, includes all coarticulatory effects – that is, phonetically contextually determined variation: [$ ˤ bə̊t̚ | wɒ̹ɫ s ðə | ˈf̹ʊɫ | pˤraɪs]. This section discusses how some of these allophones are derived during the rendering process.

Firstly, some notes on the principal coarticulation effects:

1. The [t] in the coda of [bət̚] is not ambisyllabic because in this accent (Estuary English; and also most North American English accents) it has release failure: we observe that only a fully released [t] (as in Standard British English) is ambisyllabic. Hence the onset of [wɒt̚s] is not [tw], as it would be in Standard British English, and thus would have resulted in vocal cord vibration failure at the start of the [w]. Hence the single onset segment [w] does NOT have appreciable vocal cord vibration failure. Compare this with the start of onset of [pˤraɪs], which DOES have vocal cord vibration failure (VOT). Similarly the onset of [ðə] has no ambisyllabic consonant – so no appreciable vibration failure at the start of [ð].
2. The [ɒ] of [wɒ̹t̠̚s] has some perseverative lip-rounding from the preceding [w], and the [f] of [fʊɫ] has some anticipatory lip-rounding from the following [ʊ].
3. The [t̚] of [wɒ̹t̠̚s] is somewhat retracted following the back vowel [ɒ].

In general the coarticulatory effects present in the phonetic rendering of this utterance can be divided into those that have an aerodynamic basis and those that have a mechanical basis.

1. Aerodynamically induced coarticulation

- vocal cord vibration failure:
  here symbolised [ˤ b] – utterance onset [b]
- partial vocal cord vibration failure:
  here symbolised [ə̊] – unstressed inter-voiceless stop [ə]
- partial vocal cord vibration failure:
  here symbolised [ˤr] following syllable initial [p] (VOT)

In pseudo-code:

```
utterance :                 string;
syllable :                  string;
plosives_voiced:            array of plosive;
plosives_voiceless:         array of plosive;
liquids:                    array of consonant;
segment_word-initial:       plosive;
plosives_voiced           = [b, d, g];
plosives_voiceless        = [p, t, k];
liquids                   = [l, r];
ə:                          vowel;
```

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

Table 2 Utterance plan data structure expressed as a feature matrix.

	b	ə	tˀ	w	ɒ	t	s	ð	ə	ˈf	ʊ	ɫ	p	r	aɪ	s
type	bi-phasal	uni-polar	uni-polar	bi-phasal	uni-polar	uni-phasal	uni-polar	uni-polar	uni-polar	uni-polar	uni-polar	bi-polar	bi-phasal	bi-polar	bi-polar	uni-polar
rob.	0.3	0.7	0.5	0.9	0.9	0.5	0.7	0.9	0.7	0.9	0.9	0.9	0.9	0.9	0.9	0.7
place1	lips	0.5	0.8*	lips	0.1	0.8*	0.8*	teeth	0.5	lips/ teeth	0.2	0.2	lips	0.5	0.5	0.8*
place2	lips											0.2	lips	0.6	0.7	
constr1	1	0.3	1	0.5	0.2*	1	0.8	0.7	0.3	0.8	0.4*	0.4	1*	0.2	0.1	0.8
constr2	0			0.2								0.4	0	0.3	0.4	
round	0	0	0	0.8	0.2	0	0.1	0	0	0	0.7	0.3	0	0.1	0.1	0.1
nasal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
glottis1	0.8*	0.7*	0	0.8	0.9	0	0	0.8*	0.7*	0*	0.9	0.8	0,0	0.8*	0.9	0
glottis2	0.1			0.8*								0.8*		0.8	0.9	

```
{
    if segment$_{\texttt{word-initial}}$ in[plosives$_{\texttt{voiced}}$] then vc_vibration $< 1$;
      if utterance [j] $=$ ə and utterance [j $-$ 1] in plosives$_{\texttt{voiceless}}$
        and utterance [j $+$ 1] in plosives$_{\texttt{voiceless}}$ then vc_vibration$_{ə}$ $=$ 0;
      if syllable [2] in liquids and syllable [1] in plosives$_{\texttt{voiceless}}$
        then vc_vibration$_{\texttt{syllable[2]}}$ $< 1$;
}
```

2. Mechanically induced coarticulation
- lip rounding:
  [ɒ] after syllable initial [w]
- retraction:
  [tˀ] following [ɒ]

In pseudo-code

```
{
     if syllable[2] $=$ ɒ and syllable[1] $=$ w
       then syllable[2] $=$ ɒ$_{\texttt{rounded}}$;
     if syllable[j] in plosives and syllable[j $-$ 1] $=$ ɒ
       then syllable[j] $=$ syllable[j]$_{\texttt{retracted}}$;
}
```

In table 2 above is the utterance plan (extrinsic allophonic representation) data structure expressed as a feature matrix after it has had some physical detail added at the start of the phonetics tier. The phonetic information has come from the phonetics tier on the static plane. The following have been added as the processes move from segment to segment gesture:

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

Table 3 The relationship between our numerical system and the traditional phonetic description.

Place	lips, lips/teeth, 0.9 [teeth], 0.8 [front palate], 0.5 [mid palate], 0.1 [back palate], velum, glottis
Constriction	1 [stop], 0.9 [flap, tap], 0.8 [close fricative], 0.7 [open fricative], 0.5 [high], 0.3 [mid], 0.1 [low]
Glottis	vowels: 1 (except [ə]: 0.7); liquids and semi-vowels: 0.8; voiced fricatives: 0.8; voiced plosives 0.8

Robustness	stressed vowels: 0.9 (none); [ə]: 0.7 (voice);
	[p]: 0.9; other voiceless plosives: 0.5 (place);
	[b]: 0.7 (voice); other voiced plosives 0.3 (voice and place);
	lips and lips/teeth voiceless fricatives: voiceless: 0.9 (none); voiced: 0.7 (voice);
	oral fricatives: voiceless: 0.7 (place) 0.3 (voiced);
	semi-vowels: lips-teeth: 0.9; [l], [r] 0.9.
	(items in brackets are the vulnerable parameters)

- segment gesture type (uni- or bi-phasal (focuses on constriction), uni- or bi-polar (focuses on place))
- robustness (an index of how vulnerable the segment gesture is to constraints such as coarticulation)
- place details
- constriction details
- roundness and nasality details
- glottal details

(Note that bi-phasal and bi-polar gestures require two values for place, constriction and glottal parameters.)

Table 2 does not include a complete set of parameters. We show those that we have found most useful in developing the computational model. To a certain extent the idea of a place/constriction centred characterisation comes from Browman & Goldstein (1986), but with different labels. Notice the distinction between place within the oral cavity (i.e. within the vowel place range – numerically indexed) and outside the oral cavity (place labels used).

Table 3 shows the relationship between our numerical system and a more traditional phonetic description using labels. The numerical values are system defaults, and these are the values that the extrinsic allophonic plan finds – in this sense they underlie the rendering process. In a more traditional approach a surface phonological string entering the phonetics would find the usual phonetic labels.

Initial values assigned to the robustness parameter are also included, and all values are arguable: these are our first approximation working hypotheses.

In the extrinsic allophonic plan for this utterance there are eight candidate segments for coarticulation:

- [ʿb] – vocal cord vibration fails throughout the stop (aerodynamic failure: supraglottal pressure too high)
- [ə̊] – partial vocal cord vibration failure (common in unstressed vowels)
- [p̹] – rounded (coarticulates with preceding rounded [w])
- [ť] – retracted (coarticulates with preceding back [ɒ])
- [f̹] – rounded (coarticulates with the following rounded [ʊ])
- [t̹] – rounded (coarticulates with preceding rounded [ʊ])
- [ʿr] – vocal cord vibration failure at start (aerodynamic failure: supraglottal pressure instability and pressure too high immediately following voiceless plosive [p] release)
- [aɪ] – the second pole ([ɪ]) has greater than usual constriction – coarticulates with following [s]

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

Table 4 [$ ˁbə̊t̚ | wɒ̧t̠̚s ðə | ˈƒʊł̡ | pˁraɪs] in matrix form.

	b	ə	t̚	w	ɒ	t	s	ð	ə	ˈƒ	ʊ	ł	p	r	aɪ	s
type	bi-phasal	uni-polar	uni-polar	bi-phasal	uni-polar	uni-phasal	uni-polar	uni-polar	uni-polar	uni-polar	uni-polar	bi-polar	bi-phasal	bi-polar	bi-polar	uni-polar
rob	0.3	0.7	0.5	0.9	0.9	0.5	0.7	0.9	0.7	0.9	0.9	0.9	0.9	0.9	0.9	0.7
place1	lips	0.5	0.8*	lips	0.1	0.8 →0.6*	0.8*	teeth	0.5	lips/ teeth	0.2	0.2	lips	0.5	0.5	0.8*
place2	lips											0.2	lips	0.6	0.7	
constr1	1	0.3	1	0.5	0.2*	1	0.8	0.7	0.3	0.8	0.4*	0.4	1*	0.2	0.1	0.8
constr2	0			0.2								0.4	0	0.3	0.4 →0.5	
round	0	0	0	0.8	0.2 →0.5	0	0.1	0	0	0 →0.3	0.7	0.3 →0.5	0	0.1	0.1	0.1
nasal	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
glottis1	0.8 →0,1*	0.7 →0,2*	0	0.8	0.9	0	0	0.8*	0.7*	0*	0.9	0.8	0,0	0.8 →0*	0.9	0
glottis2	0.1			0.8*								0.8*		0.8	0.9	

At this point it is the job of the CPA to predict the normal intrinsic allophonic outcome of dynamically applying the appropriate coarticulatory rules as found on the phonetic tier's static plane. In traditional notation the coarticulated string would be: [$ ˁbə̊t̚ | wɒ̧t̠̚s ðə | ˈƒʊł̡ | pˁraɪs], and in matrix form would look like table 4. In some cells however we find by experiment that the predicted value gives way to a new value (highlighted). These are instances of CPA supervision to constrain normal coarticulatory processes. In the outline model presented in this paper this table represents the final gestural specification.

The CPA deals with coarticulation

Coarticulation is the occurrence of mechanical, aerodynamic and other effects on segments that 'distort' the utterance plan. The utterance plan is not descriptive, but prescriptive in the sense that it can specify only what an utterance would look like if the subsequent coarticulatory effects did not occur. But by definition they MUST occur. The default condition is that distortions to acoustic signals which take it away from the prescribed ideal are not relevant because the listener's perceptual system 'repairs' them back to the intended, abstract ideal – which is what the listener perceives. This is an example of collaboration between speaker and listener. But under certain conditions these distortions DO matter and ambiguity or 'mishearing' can occur. Many researchers have observed that when ambiguity is likely the speaker seems to minimise the distortions (Morton 1986). In this model the CPA is the mechanism responsible for detecting or predicting adverse distortion and dealing with it (Tatham 1986b).

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

The CPA needs to address several questions concerning the predicted coarticulatory effects on the extrinsic allophonic utterance plan. Are any of these coarticulation effects likely to cause:

1. semantic ambiguity? – i.e. do any of the effects lead to phonemic changes that cannot be repaired by general semantic context and which may result in ambiguity of meaning of the entire utterance? – NO.
2. phonological ambiguity? – i.e. do any of the effects change the local meaning of any words? – NO.

These two questions are relevant to ANY utterance. The answer YES on any occasion would prompt the CPA to attempt to minimise the ambiguities by constraining the predicted coarticulation effects. It would do this by supervising motor control to increase precision in the appropriate areas of the utterance. There is a general principle here: *Precision is lax so long as there is no predicted ambiguity*.

But now the CPA must deal with other input considerations:

1. Are there any intentional effects to be brought into the utterance? – on this occasion we have decided that the utterance is to be spoken carefully. The CPA knows that 'carefully' – one of a number of possibilities it recognises – means a deliberateness of utterance reflecting increased precision and a somewhat slower tempo throughout the period in which authority is to be shown, i.e. the entire utterance in this case.
2. Are there any emotive effects for this utterance? – on this occasion, NO.

So the reasoning process conducted by the CPA continues: there are no serious ambiguities likely to be created by coarticulatory effects, but the utterance is to be spoken with generally increased precision and a reduction of the default tempo. Reduction of tempo itself will reduce the extent of coarticulation since the phenomenon is tempo dependent. An increase in tempo correlates with increased coarticulation or failure of parameters depending on their robustness, but the increase may not eliminate coarticulation altogether. Increased precision of speech does not necessarily imply much reduction of coarticulation throughout the utterance. For us it would mean more careful supervision of the motor control – which may or may not result in reduced degrees of coarticulation. Coarticulation is a complex phenomenon: segment 'targets' are more likely to be hit with increased supervision of precision, but 'edge blending' (where a segment blends into the adjacent one) is never eliminated even in the slowest continuous speech.

The only effect we noted earlier was the tendency for the [t] of [wɒ̰ṯ̚s] to be unreleased (because it preceded a same-place fricative) – a phonological rule in this speaker's accent. Authoritative speech will sometimes call for a change of accent (a phonological adjustment), or increased precision at the phonetic level (perhaps to SIMULATEchange of accent). Note that some accents are often considered to be spoken with more care or precision than others. In English it may be the case that Received Pronunciation is more carefully supervised than, say, Estuary English. We do not know if this is the case but suspect that it is not – what is happening is that the extrinsic allophonic representation in the utterance plan is different, perhaps fuller, giving the IMPRESSION of less coarticulation. But in this particular example we know that a full release of the [t] in this word can be managed at the phonetic level and can simulate a more careful accent normally handled at the phonological level. That is, this is not actually a local accent change (that would have been done earlier in the phonology), but an adjustment to the phonetics to simulate a phonological process. There is no doubt that there are a number of effects which can be both phonetic and phonological on this basis. Lateral or nasal releases of stops, for example, can often be similarly negated in favour of a 'regular' release, giving rise to a perceived effect of more careful speech (e.g. [bɒtˡɫ] – laterally released [t] into the syllabic [ɫ] – vs. [bɒtɫ] – regular release into the syllabic [ɫ]; note, though, that if the [l] has been vocalised to [lᶷ] then [tˡ] is not a possibility). (We use the superscript lower case 'l' as a diacritic on [t] to indicate lateral release: [tˡ].)

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

Within phonology rule ordering is often critical to ensure a correct derivation, and the way in which we distinguish phonology and phonetics assumes that phonology, and therefore utterance planning, occurs before rendering. For the same kind of reason that rule ordering is important in phonology the ordering of 'effects' is important in phonetics. A simple example would be that precision of articulation of a plosive will clearly influence formant bending in surrounding sounds, so a characterisation of formant bending comes after any decision regarding precision in the articulation of what causes the formant bending. Similarly decisions as to rate of delivery made by the CPA will change some aspects of coarticulatory effects – these are therefore characterised after describing the CPA action.

7 Conclusion

We have presented an outline of a model of speech production which is specifically designed to account for a number of observations about speech and speakers. The model is fully computational, with particular attention paid to the choice of suitable paradigms for representing the different data structures involved. The model is multi-dimensional in the sense that it can be approached as a static representation of the inherent features of speech production in much the same way as early transformationalists approached the characterisation of syntax, or it can be seen as a dynamic system characterising the processes involved in time-governed production of individual utterances involving detailed properties such as expression.

We have incorporated the idea of supervision, particularly in the area of phonetic rendering of utterance plans. Phonetic rendering is a complex set of procedures involving a balance between the basic requirements of the utterance plan and a number of incoming pragmatic and other constraints. To achieve this we develop the idea of agent, particularly the Cognitive Phonetic Agent operating at the phonetic rendering level. In this model of human speech production this agent is a reasoning device able to evaluate competing requirements to optimise the balance between processes.

The paper has illustrated the model by dwelling on several data structures and showing how appropriate computational paradigms characterise them. A simple utterance has been traced through from its abstract phonemic representation to a fairly detailed intrinsic allophonic representation as an example to show how some of the stages in the computational model work.

Although by no means exhaustive or even completely accurate we feel that the model is currently coherent enough to be tested against samples of real speech. The errors and hypotheses generated by the testing procedure should feed naturally into an iterative process of refinement of the model.

References

Abercrombie, D. (1967). *Elements of General Phonetics*. Edinburgh: Edinburgh University Press.

Altova GmbH & Altova Inc. (1998–2001). *XML-Spy Integrated Development Environment*. Address: Vienna, Rodolfplatz 13a/9.

Bird, S. (2002). Computational phonology. *Oxford International Encyclopedia of Linguistics* (2nd edn.). Oxford: Oxford University Press.

Browman, C. P. & Goldstein, L. M. (1986). Towards an articulatory phonology. In Ewan, C. & Anderson, J. (eds.), *Phonology Yearbook* **3**, 219–252. Cambridge: Cambridge University Press.

Cawley, G. C. & Green, A. D. P. (1991). The application of neural networks to cognitive phonetic modelling. *Proceedings of the I.E.E. International Conference on Artificial Neural Networks*, 280–284. Bournemouth, UK: IEE.

Channon, R. & Shockey, L. (eds.) (1986). *In Honor of Ilse Lehiste*. Dordrecht: Foris.

Chomsky, N. & Halle, M. (1968). *The Sound Pattern of English*. New York: Harper and Row.

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

CODE, C. & BALL, M. J. (1988). Apraxia of speech: the case for a cognitive phonetics. In Ball, M. J. (ed.), *Theoretical Linguistics and Disordered Language*, (152–167). London: Croom Helm.

CRUTTENDEN, A. (2001). *Gimson's Pronunciation of English*. London: Arnold. [See also Gimson, A. C. (1962). *An Introduction to the Pronunciation of English*. London: Edward Arnold – the first edition prior to Cruttenden's revision.]

FIRTH, J. R. (1948). Sounds and prosodies. *Transactions of the Philological Society*, 127–152. [Reproduced in Jones, W. E. & Laver, J. (eds.), *Phonetics in Linguistics: A Book of Readings*, 47–65. London: Longman.]

FOWLER, C. A. (1980). Coarticulation and theories of extrinsic timing control. *Journal of Phonetics* **8**, 113–133.

FRIJDA, N. H. (1993). The place of appraisal in emotion. *Cognition and Emotion* **7**, 357–388.

GIMSON, A. C. See Cruttenden (2001).

GUSSENHOVEN, C. (1986). English plosive allophones and ambisyllabicity. *Gramma* **10**, 119–141. Amsterdam: University of Amsterdam.

HAMMOND, M. (1999). *Phonology of English: A Prosodic Optimality-theoretic Approach*. Oxford: Oxford University Press.

HAYES, B. (1995). *Metrical Stress Theory*. Chicago, IL: University of Chicago Press.

HUCKVALE, M. (1999). Representation and processing of linguistic structures for an all-prosodic synthesis system using XML. *Proceedings of EuroSpeech '99*, 1847–1850. Budapest: European Speech Communication Association.

KAHN, D. (1976). *Syllable-based Generalizations in English Phonology.* Ph.D. dissertation, MIT. [Published 1980, New York: Garland.]

KEATING, P. (2000). A phonetician's view of phonological encoding. Talk presented at Laboratory Phonology 7, Nijmegen, June 2000.

KEATING, P. & SHATTUCK-HUFNAGEL, S. (2002). A prosodic view of word form encoding for speech production. *Working Papers in Phonetics* **101**, 112–156. Los Angeles, CA: University of California.

LADEFOGED, P. (1971). *Preliminaries to Linguistic Phonetics*. Chicago, IL: University of Chicago Press.

LADEFOGED, P. (2001). *Vowels and Consonants*. Oxford: Blackwell.

LEDOUX, J. (1996). *The Emotional Brain*. New York: Simon and Schuster.

LEWIS, E. & TATHAM, M. (1991). SPRUCE – a new text-to-speech synthesis system. *Proceedings of EuroSpeech '91*, 976–981. Genoa: European Speech Communication Association.

MORTON, K. (1986). Cognitive phonetics: some of the evidence. In Channon & Shockey (eds.), 191–194.

MORTON, K. (1992). Pragmatic phonetics. In Ainsworth, W. A. (ed.), *Advances in Speech, Hearing and Language Processing*, 17–55. London: JAI Press.

MORTON, K. & TATHAM, M. (1995). Pragmatic effects in speech synthesis. *Proceedings of EuroSpeech '95*, 1819–1822. Madrid: European Speech Communication Association.

OGDEN, R., HAWKINS, S., HOUSE, J., HUCKVALE, M., LOCAL, J., CARTER, P., DANKOVIČOVÁ, J. & HEID, S. (2000). ProSynth: an integrated prosodic approach to device-independent, natural-sounding speech synthesis. *Computer Speech and Language* **14**, 177–210.

PAGET, R. (1930). *Human Speech*. London: Kegan Paul, Trench and Tubner & New York: Harcourt, Brace.

TATHAM, M. (1971). Classifying allophones. *Language and Speech* **14**, 140–145.

TATHAM, M. (1986a). Towards a cognitive phonetics. *Journal of Phonetics* **12**, 37–47.

TATHAM, M. (1986b). Cognitive phonetics: some of the theory. In Channon & Shockey (eds.), 271–276.

TATHAM, M. (1986c). The problem of capturing linguistic and phonetic knowledge. In Lawrence, R. (ed.), *Proceedings of the Institute of Acoustics* **8**, 443–450. St Albans: Institute of Acoustics.

TATHAM, M. (1994). The supervision of speech production: an issue in speech theory. In Lawrence, R. (ed.), *Proceedings of the Institute of Acoustics* **16**, 171–182. St Albans: Institute of Acoustics.

TATHAM, M. (1995). The supervision of speech production. In Sorin, C., Mariani, J., Meloni, H. & Schoentgen, J. (eds.), *Levels in Speech Communication: Relations and Interactions*, 115–125. Amsterdam: Elsevier.

TATHAM, M. & MORTON, K. (1980). Precision. *Occasional Papers* **23**, 104–116. Colchester: University of Essex, Linguistics Department.

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press

TATHAM, M. & MORTON, K. (1988). Knowledge representation and speech production/perception modelling in an artificial intelligence environment. In Ainsworth, W. A. & Holmes, J. N. (eds.), *Proceedings of Speech '88* (7th FASE Symposium), 1053–1060. Edinburgh: Institute of Acoustics.

TATHAM, M. & MORTON, K. (2001). Intrinsic and adjusted unit length in English rhythm synthesis. In Lawrence, R. (ed.), *Proceedings of the Institute of Acoustics* **23**, 189–200. St Albans: Institute of Acoustics.

TATHAM, M. & MORTON, K. (2002). Computational modelling of speech production: English rhythm. In Braun, A. & Herbert, R. (eds.), *Phonetics and its Applications: Festschrift for Jens-Peter Köster on the Occasion of his 60th Birthday*, 383–405. Stuttgart: Steiner.

TATHAM, M. & MORTON, K. (2003 in press). *Expression in Speech: Natural and Synthetic*. Oxford: Oxford University Press.

TATHAM, M., MORTON, K. & LEWIS, E. (1998). Assignment of intonation in a high-level speech synthesiser. In Lawrence, R. (ed.), *Proceedings of the Institute of Acoustics* **20**, 255–262. St Albans: Institute of Acoustics.

TATHAM, M., MORTON, K. & LEWIS, E. (2000). SPRUCE: speech synthesis for dialogue systems. In Taylor, M. M., Néel, F. & Bouwhuis, D. G. (eds.), *The Structure of Multimodal Dialogue* II, 271–292. Amsterdam: John Benjamins.

WELLS, J. C. (1982). *Accents of English*. Cambridge: Cambridge University Press.

https://doi.org/10.1017/S0025100303001142 Published online by Cambridge University Press