PANTOME: AN ARCHITECTURE FOR SPEECH AND NATURAL LANGUAGE PROCESSING

William Edmondson and Jon Iles

The University of Birmingham, School of Computer Science, Edgbaston, Birmingham. B15 2TT.

1. INTRODUCTION

The history of speech and natural language processing shows strong influences from engineering, in the form of pipe-line models and signal processing, but surprisingly little from cognitive science and linguistics. We believe it is necessary to demonstrate that coherent approaches can be formulated which allow the problem of processing natural language in a uniform manner to be considered, and that these can be well-informed by research in human processing of language. This is in contrast to many current approaches where examination of small isolated sub-topics within natural language processing is the norm.

In this paper we outline a suitable architecture for the unification of many fields of natural language processing research and discuss how we can demonstrate the validity of our claims by implementation. The architecture we have developed is called Pantome, and the discussion below begins with Pantome and its origins. The second topic discussed is a new speech synthesis strategy which can be driven within the Pantome architecture. This topic is presented fully in a related paper [13] so its treatment here is restricted to the details of speech synthesis driven from within Pantome. The paper ends with a discussion of the use of Pantome in a fully integrated system.

2. PANTOME

2.1 PHONOLOGICAL ORIGINS OF THE ARCHITECTURE

In a series of papers Edmondson [3, 4, 5, 6, 7, 8, 9, 10] has argued that the conventional formalisms of non-linear, or auto-segmental, phonology are too restricted. The feature geometry approach of Clements [1] is more restricted than the major alternative, the 'bottle-brush' approach of Hayes [11] (see figure 1 for illustrations), but the restriction is deeper than their disagreement. The flaw is that in most cases the formalisms are assumed to be inherently speech specific; the only behaviour to be accounted for is the production of sequences of speech segments.

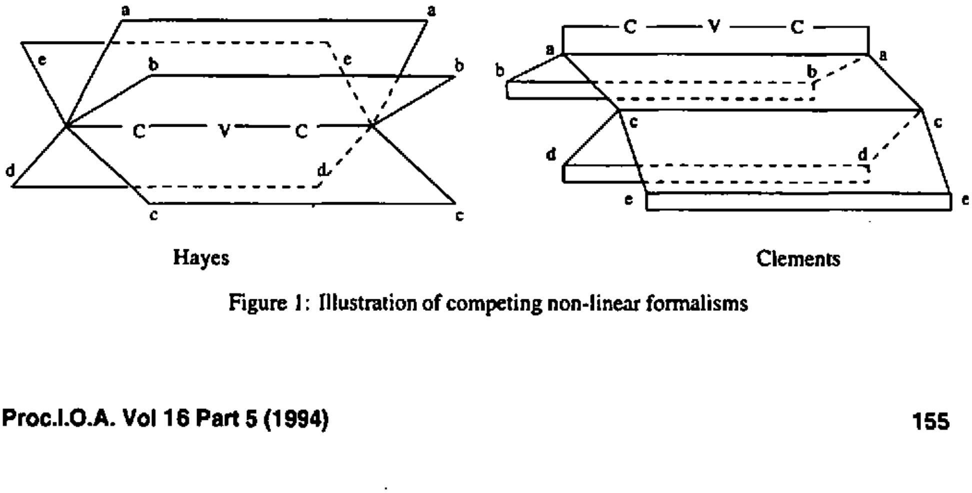

Figure 1: Illustration of competing non-linear formalisms

Attempts to generalise the formalisms have been made, but these are flawed in the same general way. For example, application of work in non-linear phonology to sign language behaviour does not, of course, make use of the notion of speech segment, but it does rely upon the notion of linguistic, articulatory, segmentation. In signed language 'phonology' there is little agreement about the most suitable approach (cf. Edmondson [5, 10], Perlmutter [15], Sandler [16], Wilbur [17]), but nonetheless many authors view the notion of segment as unquestionable. These generalizations thus only yield a differently special purpose formalism, not a *general* formalism.

2.2 COGNITIVE BASIS OF THE PANTOME ARCHITECTURE

It can be argued [4] that the real significance of the non-linear approach is its value in accounting for the assembly of sequentially organised behaviour from atemporal cognitive precursors - the 'linearization process' as it is known. The process is the same, inherently, whether the behaviour is speech, sign, or interaction with a computer (cf. Cypher [2], Edmondson [6]). Segments are the product of this process, not the underlying units of behaviour (cf. Kaye [14]). Removal of the emphasis on the speech segment in the non-linear formalisms can un-couple the linearization process from speech. The approach we have taken does this, and it yields a formal architecture which is more general and may be applied to any cognitive activity. This architecture is called Pantome.

In the current project we are confining our interests to natural language; an example will help to clarify the utility of our approach. Consider speech communication between a speaker and a listener. The speaker has a desire to communicate a concept to the listener. This concept may be thought of as an inherently atemporal collection of entities. Between the point at which the concept is conceived, and the point at which it is expressed as speech, a transformation takes place in terms of the representation of the concept. It changes from being atemporal, to being a collection of temporally organized sub-entities, in this case a series of articulatory gestures. This is the process of linearization, moving from the atemporal to the temporal.

For the listener, the reverse of the process described above is true. On receipt of the speech waveform, this temporally organised series of acoustic features is processed in such a way that the end result is a version of the concept the speaker was attempting to communicate. Here we see the process of linearization reversed; it is de-linearization. While these activities are occurring there are undoubtedly other tasks to consider. The speaker will be monitoring the ambient noise in the surroundings where the conversation is taking place, the body language of the listener and the "difficulty" of the concepts that are being communicated. All of these factors will have a positive feedback effect and the speaker will adjust his or her speech accordingly; for example the precision of articulation or rate of speech may be modified. The listener will also be engaged in similar tasks, for example paying more attention to facial features and lip movements if the ambient noise level increases. These activities are concurrent with the speech itself.

Current approaches to speech recognition and text-to-speech conversion can generally be thought of as being "pipe-line" models: processing is carried out in sequential stages (MITalk is a well-established example). Often the input representation is transformed stage by stage into the required output representation by way of rewrite rules, or similar operations. These pipe-line architectures provide little scope for integration of the many disparate sources of information that we see are involved in human speech communication. There is even less scope for the integration of the synthesis and recognition processes even though they appear to be so closely interlinked in human speakers. It is in fact very difficult to say that existing systems are currently modelling any of the processes involved in human speech communication, apart from the simple surface results of speaking and listening. If the eventual goal of current speech synthesis and recognition technology is to match human performance, then we believe it to be essential that more than a passing account must be taken of human behaviour. We believe that the Pantome architecture is the only proposal so far capable of supporting close integration of speech production and recognition, for example, and thus demonstrating human-like processing behaviour. The potential of Pantome will become clear - our 'worked example' is text-to-speech conversion, but generalization to speech recognition is not problematic.

2.3 PANTOME APPLIED TO NATURAL LANGUAGE PROCESSING

Pantome is the name we have given to the architecture based on the generalized non-linear formalism. The key elements of this architecture are illustrated in figure 2. It is worth noting at this point that the architecture does not operate in a pipe-lined manner. All operations can be carried out in parallel, and the architecture has been conceived to support this approach.

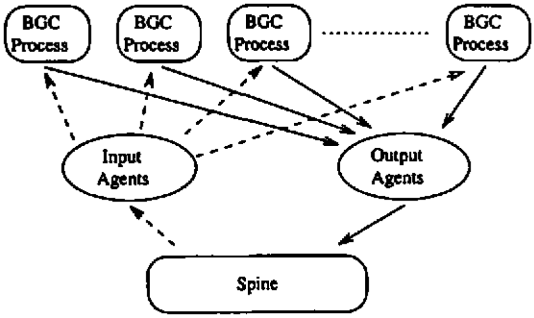

Figure 2: Key elements of the Pantome architecture

The spine is the central data structure: this corresponds closely to the notation used by Hayes in figure 1. This data structure consists of a number of data items known as "segments". The idea of a segment is distinct from the notion of a speech segment. In the context of Pantome a segment can used to represent any given item of data. For example in the case of a text-to-speech system (described below, and in [13]) there may be letter segments, word segments, syllable segments and so on. This is illustrated in figure 3. Part (a) of figure 3 illustrates an extract from the spine structure containing letter segments, word segments, syllable segments and phoneme segments. The inter-relationships between these segments are shown in full.

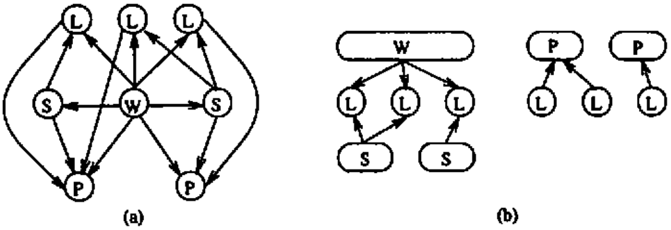

Figure 3: Two views of an extract from the spine structure

To simplify figure 3 slightly, part (b) illustrates the same structure in two parts. In reality this data structure can be considered three dimensional; figure 4 gives a perspective view of a similar structure. The spheres in this figure represent individual segments; the solid lines indicate the inter-relationships between them.

The input and output agents represented in figure 2 handle reading and writing segments from and to the spine data structure. All references to the spine structure are dealt by these agents. They provide a standard interface to the data structure, and also handle all potential consistency problems within the data structure that may arise due to the parallel nature of its operation. The flow of data within the architecture is represented by the arrows connecting the various elements together.

Adding data to the spine is represented as the solid lines, reading data from the spine is indicated by the dashed lines. The agents themselves have some degree of autonomy; they are supplied with a set of domain specific rules that allow them to complete parts of the structure automatically when new segments are added.

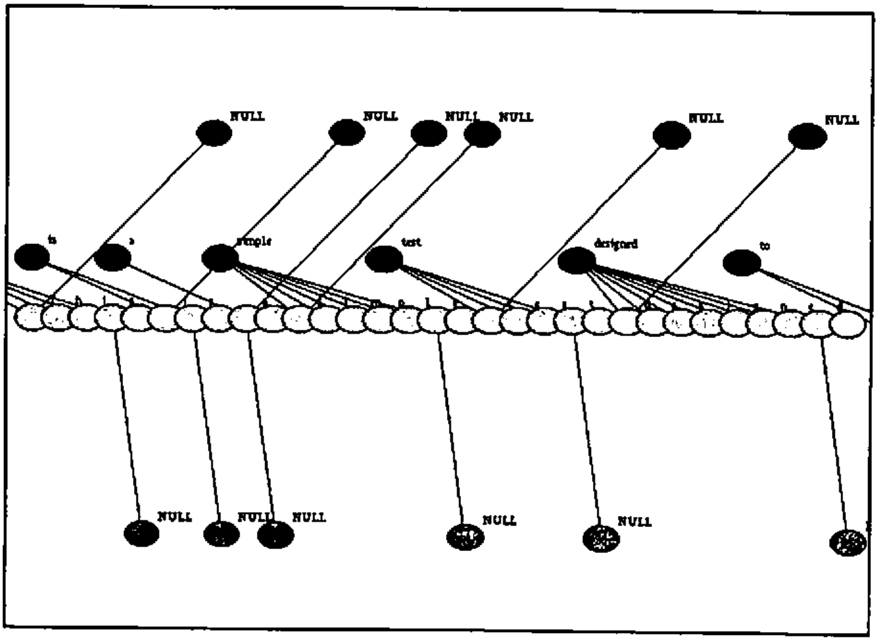

Figure 4: Perspective view of an extract from the spine structure

The final elements of the Pantome architecture are the Background Context Processes (illustrated as "BGC Processes" in figure 2). These are a collection of heterogeneous processes that provide all of the domain dependent processing required to convert the general purpose Pantome architecture to a domain specific application. In our text-to-speech conversion example, we would expect to see BGC processes dealing with letter-to-sound rules, dictionary lookup, morphological analysis, syllabification and so on. Each of these processes will be attempting to add more detail to the structure by adding new segments of particular types and defining their relationships with the existing structure. The Pantome architecture places no restriction on the type or number of processes that are attached as BGC processes. This enables information from disparate heterogeneous sources to be combined in one structure. The architecture allows this structure to be viewed from any perspective and at any level of detail, thus providing for whatever level of contextual detail is required.

Importantly, the Pantome architecture is bi-directional. Its parallel nature allows it to construct structures that represent input and output simultaneously, and allows these structure to be inter-related when required. This bi-directional nature allows common "knowledge" resources to be shared between input and output tasks thus closely integrating the two processes. The architecture also has the facility to re-evaluate parts of the structure when new information arrives, and selectively update the parts of the structure that are affected by the new data.

Two further points need to be noted here: a) Pantome is unlike any previous modelling architecture, and its development over several years has revealed no flaws in the underpinning theory; b) a prototype of this architecture has already been successfully developed [12, 13]. Pantome is ideally suited as an architecture for natural language processing in a human-like manner, and we can illustrate this point with the example of text-to-speech conversion.

3. TEXT-TO-SPEECH CONVERSION

The production of speech appears to be the production of sequences of discrete entities - the segments of speech, or phones as they are called. This self-evident notion is flawed, as can readily be appreciated from the fact that the articulators employed in speaking do not jointly make sequences of separable movements. What actually happens instead is that the articulators make independently controlled but co-ordinated and continuously variable movements the effects of which are perceived as segments. This observation is vital to successful work in natural language processing in a human-like manner. Two points follow from the observation.

The first point to note is that converting text to speech is not the simple process of converting one string of symbols into another using a look-up table. This is obvious anyway; such systems do not work well. The point here is that they cannot work. What is required instead is understanding of the text, at some level, and then synthesis of the speech from that abstract representation. The architecture of Pantome is designed to facilitate this.

The second point is that the synthesis philosophy used must not be a simple segment driven system - speech is not like that, and furthermore such a system would undermine the requirement for avoiding character to phoneme conversion. We take this point first, in the next subsection, before returning to the architecture of Pantome in text-to-speech conversion.

3.1 FEATURE DRIVEN FORMANT SYNTHESIS

This sub-section describes our approach to solving the problem of producing a synthesis strategy that will provide high quality synthetic speech, capable of exhibiting all of the attributes of natural speech. Our approach has been to consider a hybrid strategy between a formant synthesis technique and articulatory control. The formant synthesis technique can provide high quality synthetic speech, indistinguishable from natural speech given appropriate control parameters. It is infinitely variable and language independent.

Articulatory approaches in principle allow control of synthetic speech in an intuitive manner that simplifies specification of the attributes of natural speech (e.g. raise the tongue, round the lips etc.) but problems remain with ensuring that control is managed in a way which blends with other (e.g. linguistic) requirements. We have called our approach "feature driven formant synthesis" (FDFS) as we are using a set of *linguistic features* to provide control over formant synthesis. These features are, in effect, quasi-articulatory descriptors. The use of linguistic features means that work in linguistics and cognitive science can be incorporated directly, to ensure human-like processing. Additionally, such a feature based approach is quite unlike the conventional segmental approach and is, furthermore, very suitable for the parallel control environment provided via Pantome because the architecture provides independent control of the linguistic features, in a time-varying manner.

One of the benefits of our approach is that we can control the speech synthesis process in an intuitive manner. Take as an example the problem of specifying coarticulation. With our approach we can examine coarticulation from the point of view of the effect that the articulation of the second phoneme has on the first. We can allow the articulation of the second phoneme to begin before the articulation of the first is complete, and calculate the effect that this second set of articulatory features has on the first. This "overlap" allows us to model anticipatory coarticulation in a way that is closer to the physiological reality of coarticulation in natural speech. This is illustrated in figure 5.

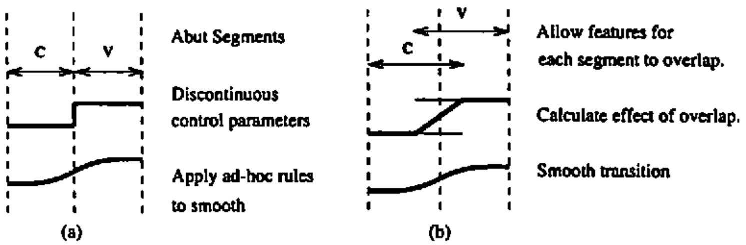

Figure 5: Coarticulation using (a) conventional synthesis and (b) FDFS

It should be clear that unlike conventional parameter control of formant synthesis, the adjustments are not ad-hoc interpolations in parameter space. The interpolations are managed in the quasi-articulatory space of feature values (and of course where appropriate these are not arbitrarily restricted to binary values). Attempts have been made to manage coarticulation or segment transitions more sensitively, but these reduce to management of the formant parameter values, not the phenomena which give rise to them.

The FDFS system requires, at base, the continuous specification of a set of feature values, for example tongue height, lip rounding and so on. Not only do these continuous-valued parameters allow well-known effects to be modelled (for example, coarticulation), they also provide the opportunity to model, directly effects such as precision of articulation [13]. Experiments with this technique are continuing, but already useful insights have been gained, notably that 'duration' may not directly be a controllable component of stressed pronunciation. Instead, we conjecture, duration is varied as a side-effect of specific control of articulatory precision (in short: precision needs time - time does not sanction precision).

3.2 CONTROL OF SYNTHESIS WITHIN PANTOME

In output mode Pantome builds a data structure from multifarious specifications in the Background Context. These specifications are linked into the segment sequence by the appropriate output agents and the result is a frame by frame set of quasi-articulatory feature specifications. To effect actual speech synthesis these feature specifications are applied to the synthesizer via a conversion mapping [12, 13] which yields the required control parameter values.

Within the Pantome architecture there is no specification of conventional speech segments unless these are generated in BGC as useful domains of feature value specification. For example, it could happen that a so-called 'segment boundary' emerges from the approximately co-incident transition of several feature values, with the following boundary likewise emergent (but not necessarily from transitions in the same features). In this way Pantome directly implements insights from non-linear phonology whilst actually being domain independent (different knowledge resources in BGC would permit synthesis of sign language, using a different synthesizer, of course).

3.3 TEXT INPUT AND TEXT-TO-SPEECH CONVERSION

Text input, just as feature specification on output, requires a way of showing how each moment of time is accounted for in terms of the various possible descriptions or specifications, at all the levels simultaneously. The non-linear model provides, in essence, a way of capturing the co-ordination of time varying activity at many different levels or scales. It is of no concern that on text input the specification process is driven from the bottom; the effect is still that of provision of specification details in the spine, in relation to abstract representations in BGC.

The bi-directional properties of Pantome permit the analysis of a text string in terms of many specifying details - attributes related to prosody, syntax, pragmatics etc., can all be used to re-represent the text string in a more arbitrary form. This arbitrary form, available in the BGC, is then available to drive synthesis.

In this model we see that the sequentially organized stream of events, or segments, is characterized in co-ordinated fashion by many different descriptors with different domains. These specifications are not processed in any sequence - the segments are characterized by a descriptor as soon as that descriptor is available. Text-to-speech synthesis is, therefore, the production, from text, of the specifications in the different domains. In consequence of the generalized non-linear approach to the production of speech we can argue that the control of a speech synthesizer of the sort described earlier - FDFS - is 'fully' linguistic. In principle *any* linguistic specification of speech production can be modelled in a non-linear way, and thus incorporated into the system, and this means that the representation derived from the text input can be as complex as desired.

4. INTEGRATION OF SPEECH RECOGNITION WITH TEXT-TO-SPEECH CONVERSION

The power of Pantome as a language processing architecture is simply revealed through consideration of what is required for incorporation of speech recognition (to text output) into a text-to-speech conversion system such as that described above.

Speech recognition to text requires only that signal processing software can derive quasi-articulatory parameters from the signal, and this is partly available, potentially, from the mapping transformation which converts feature space into synthesis parameter control space. If processing software can automatically derive synthesis parameters from speech (as is required for automatic copy synthesis, for example) then the existing BGC knowledge resources can handle the remaining work. Thus, the full power of Pantome is that minimal extra work is required to greatly extend functionality. The architecture is genuinely general purpose.

5. SUMMARY

We have shown that, in contrast to conventional pipe-line architectures, the architecture of Pantome can provide an inherently parallel basis for broadly integrative natural language processing. Text-to-speech synthesis is readily achieved with Pantome, and the feasibility of broadening functionality by integrating speech recognition to text seems clear (and has, in fact, been experimentally explored). We expect Pantome to become a significant tool for speech and natural language processing.

6. ACKNOWLEDGEMENTS

The work described in this paper has been supported in part by Apricot Computers Limited, a subsidiary of Mitsubishi Electric UK Limited, the Science and Engineering Research Council, and GPT Limited.

7. REFERENCES

[1] G.N. Clements. The geometry of phonological features. *Phonology Yearbook*, 2:225–252, 1985.

[2] A. Cypher. The structure of the user's activities. In D.A. Norman and S. Draper, editors, *User Centred System Design: New Perspectives on Human-Computer Interaction*. Lawrence Earlbaum Associates, 1986.

[3] W.H. Edmondson. Autosegmental phonology: The syllable and the sign. In B.T. Tervoort, editor, *Signs of Life. Proceedings of the Second European Congress on Sign Language Research*, pages 79–94, University of Amsterdam, 1986. Institute of General Linguistics.

[4] W.H. Edmondson. Issues in linearisation: Prolegomena for a general theory of communication. *Language and Communication*, 6(4):225–266, 1986.

[5] W.H. Edmondson. Segments in signed languages: Do they exist and does it matter? In W.H. Edmondson and F. Karlsson, editors, *SLR '87. Papers from The Fourth International Symposium on Sign Language Research*, pages 66–74, Hamburg, 1987. Signum-Press.

[6] W.H. Edmondson. Asynchronous parallelism in human behaviour: A cognitive science perspective on human-computer interaction. *Behaviour and Information Technology*, 8(1):3–12, 1989.

[7] W.H. Edmondson. Asynchronous parallelism in the formalism of non-linear phonology. In M.M. Taylor, F. Neel, and D.G. Bouwhuis, editors, *The Structure of Multimodal Dialogue*, pages 241–247. North-Holland, Amsterdam, 1989.

[8] W.H Edmondson. A non-concatenative account of classifier morphology in signed and spoken languages. In S. Prillwitz and T. Vollhaber, editors, *Current Trends in European Sign Language Research. Proceedings of the 3rd European Congress on Sign Language Research*, pages 187–202, 1990.

[9] W.H. Edmondson. General cognitive principles and the structure of behaviour. Paper presented at the Third International Colloquium on Cognitive Science, Donostia - San Sebastian., May 1993.

[10] W.H. Edmondson. Signs, syllables and segments: What do they mean? Invited contribution to the Workshop on Phonology and Morphology of Sign Language, Amsterdam, June 1993.

[11] B. Hayes. Dipthongisation and coindexing. *Phonology*, 7:31–71, 1990.

[12] J.P. Iles and W.H. Edmondson. The use of a non-linear model for text-to-speech conversion. In *Proceedings of the European Conference on Speech Technology - EUROSPEECH*, volume 2, pages 1467–1470. ESCA, September 1993.

[13] J.P. Iles and W.H. Edmondson. Feature driven formant synthesis. Appears in these proceedings, November 1994.

[14] J. Kaye. *Phonology: a cognitive view*. Lawrence Earlbaum Associates, New Jersey, 1989.

[15] D.M. Perlmutter. Sonority and syllable structure in American Sign Language. *Linguistic Inquiry*, 23(3):407–442, 1992.

[16] W. Sandler. *Sequentiality and simultaneity in American Sign Language phonology*. Foris, Dordrecht, 1989.

[17] R.B. Wilbur. Why syllables? what the notion means for ASL research. In *Theoretical issues in sign language research*, volume 1, chapter 4, pages 81–108. University of Chicago Press, Chicago, 1990.