Pantome: an integrative architecture for speech and natural language processing

William Edmondson and **Jon Iles**
School of Computer Science,
University of Birmingham,
Edgbaston, Birmingham.
B15 2TT. UK.
email: (whe or jpi)@cs.bham.ac.uk

Paul Mc Kevitt
Department of Computer Science,
Regent Court, 211 Portobello Street,
University of Sheffield, Sheffield.
S1 4DP. UK.
email: p.mckevitt@dcs.shef.ac.uk

Abstract

Speech and natural language researchers now see the need to demonstrate that coherent approaches are being formulated which allow the problem of processing natural language in a uniform manner to be considered. This is in contrast to many current approaches where examination of small isolated sub-topics within natural language processing is the norm. It is our belief that very little work of any significance has been done with the aim of unification in mind. In this paper we outline what we believe to be a suitable architecture for the unification of many fields of natural language processing research, an architecture which has been developed by generalizing the formalism of non-linear phonology. We show that this architecture, called Pantome, can handle in a unified way a range of pragmatic influences on speech.

1 Introduction

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 "objects". 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 temporally organized, in this case a series of articulatory gestures. This is the process of linearization, moving from the atemporal to the temporal [8, 25]. 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 situation of the interchange, as well as the intentions of both participants, contribute pragmatic factors which surface (in part) in various audible effects. 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.

Pragmatic contributions are not limited to management of 'signal quality' in the discourse, although this is very important. Significant pragmatic factors include contextualizing the discourse in the environment of the participants (which may exploit a variety of non-verbal procedures not further discussed here), and contextualizing fragments of the discourse within the overall speech activity. These frequently supra-sentential effects are provided semantically (reference and anaphor) and acoustically (prosody).

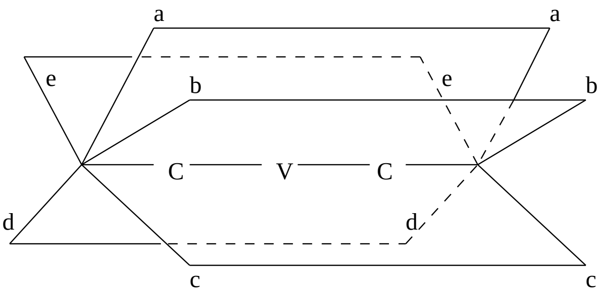
Speech synthesis from text, and also speech recognition-to-text, requires these pragmatic contributions to be utilized. The development of speech processing techniques which can incorporate pragmatic factors such as those described in [27] is a major step towards an integrated system. In what follows, we first outline the processing architecture of Pantome - developed from theoretical work in phonology and capable of managing the linearization and de-linearization required in human communication. We then show how this supports quasi-articulatory control of speech synthesis, in a manner which permits incorporation of pragmatic factors. The use of Pantome in a speech synthesis system automatically provides the basis for a speech recognition system, and this is outlined, demonstrating the integrative properties of the architecture.

2 The architecture

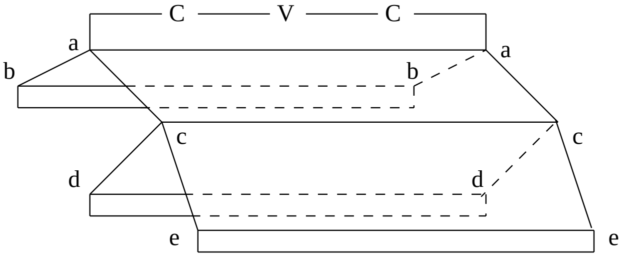
2.1 Previous Models

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. 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 one of the few proposals so far capable of supporting the closely integrated model of speech production and recognition that we have discussed above.



Hayes



Clements

Figure 1: Illustration of competing non-linear formalisms

2.2 Non-linear origins

In a series of papers Edmondson [7, 8, 9, 10, 11, 12, 13, 14] has argued that the conventional formalisms of non-linear, or auto-segmental, phonology are too restricted. The feature geometry approach of Clements [5] is more restricted than the major alternative, the 'bottle-brush' approach of Hayes [18] (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. For example, both formalisms illustrated in Figure 1 have a "spine" containing a sequence of consonants and vowels.

Attempts to generalise the formalisms have been made, but these are flawed in the same general way. 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 [9], [14], Perlmutter [34], Sandler [36], Wilbur [37]), but nonetheless many authors view the notion of segment as unquestionable.

It can be argued [8] that the general 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. The process is the same, inherently, whether the behaviour is speech, sign, or interaction with a computer (cf. Cypher [6], Edmondson [10]). Segments are the product of this process, not the underlying units of behaviour (cf. Kaye [24]). Removal of the emphasis on the speech segment in these formalisms has un-coupled the linearization process from speech - the approach is now more general and may be applied to any cognitive activity.

We restrict the discussion here to speech and natural language processing.

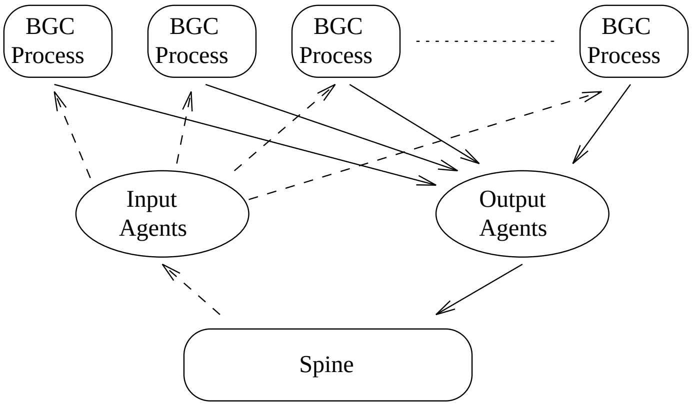


Figure 2: Key elements of the Pantome architecture

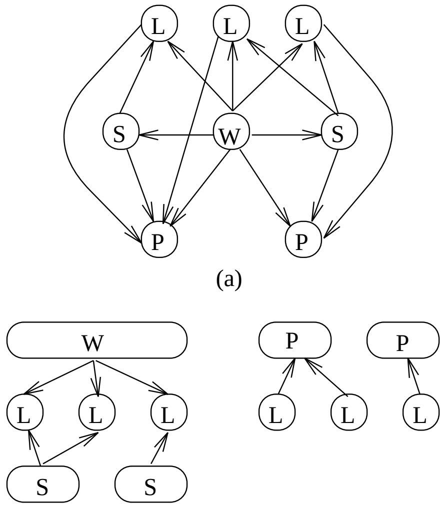
2.3 Pantome

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.

The spine is the central data structure: this corresponds directly with 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 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. The use of the term "segment" here is equivalent to the use of the term "agent" in [35].

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.



(a)

(b)

Figure 3: Two views of an extract from the spine

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.

To simplify Figure 3 (a) 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 representing the relationships between letters, words, word starts and word ends in a sentence. The spheres in Figure 4 represent individual segments; the lines indicate the inter-relationships between them.

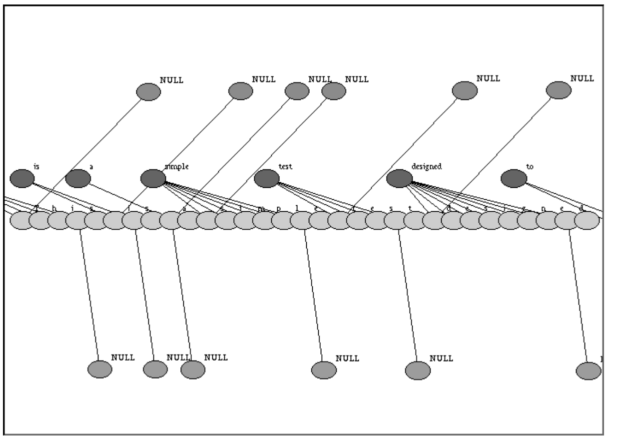


Figure 4: Perspective view of an extract from the spine

In the domain we are discussing here - combining speech with pragmatic factors - we would expect to see BGC processes dealing with letter-to-sound rules, dictionary lookup, morphological analysis, syllabification and also the interrelation of prosodic factors (stress, pitch contours) with the structures of reference in the discourse. 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.

Finally, this 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 integrating closely 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 version of this architecture has been successfully implemented in a prototype text-to-speech system [22].

2.4 Pantome in use

The architecture has been developed and implemented for use, initially, in a text-to-speech system which provides quasi-articulatory control of speech synthesis. In this configuration output agents (see Figure 2) control the spinal specifications of speech production in linguistically and prosodically useful terms, for example tongue height, lip rounding, rhythm, and articulatory precision.

This work is described elsewhere [22, 23]. Suffice to say here that a prototype synthesis system is nearly complete; Pantome is available, relevant BGC processes exist (e.g. for morphology and lexical retrieval), quasi-articulatory synthesis is well advanced. A major component which remains to be supplied concerns pragmatic factors which surface acoustically. Given the architecture into which these factors must be incorporated it is possible to specify the pragmatic components as BGC processes.

The traditional approach to natural-language processing in AI has been to use rules, or grammars, to dictate the global behaviour of a system which analyses incoming natural-language sentences. Many of the approaches use grammars of English to parse sentences into structures called *parse trees.* In Rowe and Mc Kevitt's approach [35], the individual words of a sentence act as low-level agents which have their own rules of behaviour. These rules provide three types of information: syntactic information on structural constraints, semantic information on meaning constraints, and pragmatic information on usage constraints. These components can be extracted from the existing framework [35] and reconfigured to suit the architecture of Pantome, thus providing the demonstration of integration sought here.

3 Integrating pragmatics with Pantome

A theory of intention analysis (see [27]) has been proposed as a model, in part, of the coherence of natural-language dialogue. A central principle of the theory is that coherence of natural-language dialogue can be modelled by analysing sequences of intention. The theory has been incorporated within a computational model in the form of a computer program called the Operating System CONsultant (OSCON) (see Guthrie et al. [17], Mc Kevitt [26, 28, 27], Mc Kevitt and Wilks [32], and Mc Kevitt et al. [29, 31, 30]. OSCON, which is written in Quintus Prolog, understands, and answers in English, English queries about computer operating systems.

The computational model has the ability to analyse sequences of intention. The *analysis of intention* has at least two properties: (1) that it is possible to recognise intention, and (2) that it is possible to represent intention. The syntax, semantics and pragmatics of natural-language utterances can be used for intention recognition. Intention sequences in natural-language dialogue can be represented by what we call *intention graphs.* Intention graphs represent frequencies of occurrence of intention pairs in a given natural-language dialogue. An ordering of intentions based on *satisfaction* exists, and when used in conjunction with intention sequences, indicates the *local*[1] and *global* degree of expertise of a speaker in a dialogue.

The architecture of the OSCON system consists of six basic modules and two extension modules. There are at least two arguments for modularising any system: (1) it is much easier to update the system at any point, and (2) it is easier to map the system over to another domain. The six basic modules in OSCON are as follows:

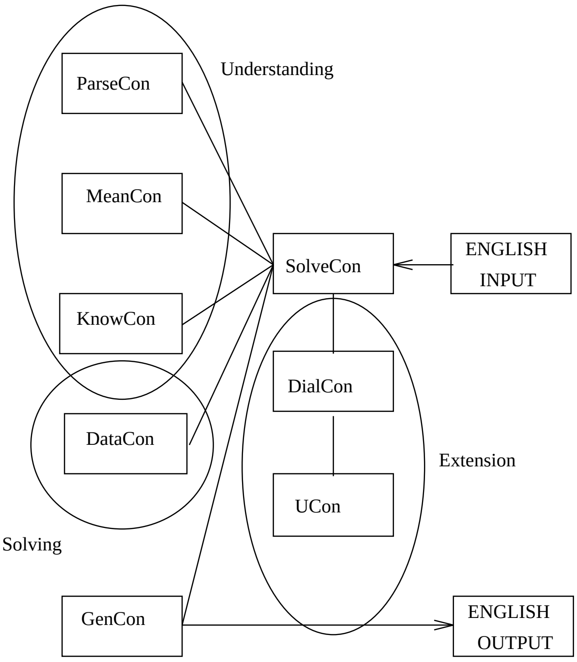


Figure 5: Architecture of the Operating System CONsultant (OSCON) system

1. ParseCon: natural-language syntactic grammar parser which detects query-type,
2. MeanCon: a natural-language semantic grammar (see Brown et al. [3], and Burton [4]) which determines query meaning.
3. KnowCon: a knowledge representation, containing information on natural-language verbs, for understanding.
4. DataCon: a knowledge representation for containing information about operating system commands.
5. SolveCon: a solver for resolving query representations against knowledge base representations.
6. GenCon: a natural-language generator for generating answers in English.

These six modules are satisfactory if user queries are treated independently, or in a context-free manner. However, the following two extension modules are necessary for dialogue-modelling and user-modelling: (1) DialCon: a dialogue modelling component which uses an intention matrix to track intention sequences in a dialogue, and (2) UCon: a user-modeller which computes levels of user-satisfaction from the intention matrix and provides information for both context-sensitive and user-sensitive natural-language generation. A diagram of OSCON's architecture is shown in Figure 5.

ParseCon consists of a set of Prolog predicates which read natural-language input and determine the

---

[1] By *local* expertise we wish to stress the fact that sometimes experts can act as novices in areas of a domain which they do not know well.

type of query being asked, or intention type presented, by the user. For each type of query there are tests for characteristic ways that people might utter that query. ParseCon uses a semantic grammar, in the Definite Clause Grammar (DCG) [2] formalism of Prolog.

MeanCon consists of predicates which check queries for important information. There are predicates which check for mentioned (1) command names (e.g. "ls", "more"), (2) command-effect specifications (e.g "see a file"), and (3) concepts, or objects (e.g. "file", "directory"). In case (2) there are specific types of information searched for: (1) **verb** specifying action (e.g. "see", "remove"), (2) **object** of action (e.g. "file"), (3) **modifier** of object (e.g. "contents"), and (4) **location** of object (e.g. "screen"). MeanCon also checks for option verbs (e.g "number") and option verb objects (e.g. "lines"). MeanCon contains a dictionary of English words that define categories such as "person", "modifier", "article", "quantifier" and "prepositions".

KnowCon consists of a set of data files to represent knowledge about the domain language used for understanding English queries. Data files here contain information about English verbs which denote types of command or action. Examples of categories of action are:

1. creating
2. screenlisting
3. printerlisting
4. sending
5. transferring
6. removing

KnowCon also contains grammar rules for operating system objects like "date", "file" and "directory". The grammar rules encode characteristic ways in which people talk about the objects in English.

DataCon consists of a set of data files defining detailed information about operating system commands. This information is stored for the UNIX and MS-DOS Operating Systems. The data for UNIX is split among seven files about commands:

1. preconditions
2. effects
3. syntax
4. names
5. precondition options
6. effect options
7. name options

The first four files contain basic data about commands while the last three contain data for options. For MS-DOS, data is contained in just four files which are similar, in spirit, to the first four here.

SolveCon is a solver which constructs and matches representations of user queries (called *Formal Queries*) against the knowledge base, DataCon, and produces an instantiated Formal Query which serves as an answer for the query. SolveCon is the heart, or driver, of the OSCON program because it contains the information for mapping English sentences into instantiated formal queries. It contains a set of complex rules which call other OSCON modules to determine (1) query type, (2) intention type, and (3) the instantiated Formal Query for that query. The determination of intention type is a two stage process where natural-language queries are first mapped into query types, and then into intention types. SolveCon also checks for repetitions by comparing the propositional content, or topic, of the current intention against that of the previous.

GenCon is the natural-language generator for OSCON and maps instantiated information from SolveCon into English answers. Here, there are algorithms for printing out (1) preconditions, (2) effects (or postconditions), and (3) syntax of commands. Also, there are predicates for printing out examples of the use of commands and command compositions. The type of query asked by the user determines the information presented in English to the user.

DialCon is the dialogue modeller for OSCON which updates the *intention matrix* representing intention pair frequencies in the dialogue. Matrix update is conducted by locating the relevant cell in the matrix which needs to be updated, and increases its count by 1. DialCon indexes the cell in the matrix by pairing the current intention type with the previous.

UCon is the user-modelling component of OSCON. UCon derives a binary measure of user expertise, *expert* and *novice*. UCon applies a user-modelling function to the intention matrix to determine levels of user *satisfaction* and *dissatisfaction*. Initially, the user is assumed to be an expert. Subsequent changes in the levels of satisfaction and dissatisfaction will result in changes in the level of user expertise. Such information is used by GenCon to generate context-sensitive and user-sensitive natural-language responses. A detailed analysis of how the system can modify its natural language responses is given elsewhere (see Mc Kevitt [27]). We will not discuss details of processing within components of the OSCON system. These can be found in [27].

The OSCON system has been used to test the what we call the *Intention-Computer* hypothesis: that the analysis of intention facilitates effective natural-language dialogue between different types of people and a computer. OSCON provides positive evidence for this hypothesis (see Mc Kevitt [27], Mc Kevitt et al. [29, 31, 30]).

It is our intention to incorporate segments for intention analysis from not only the natural language pro-

---

[2] Definite Clause Grammars (DCG's) were first developed by Pereira and Warren [33] as a tool to be used in Prolog for natural-language processing.

cessing side but also from the speech processing side into the Pantome architecture in order to enable higher level intention processing to occur. This requires that the components of teh OSCON system be reconfigured so that, for example, SolveCon is not central (cf the discussion of vowels and consonants in the spine), and Ucon and DialCon are not pipe-lined from SolveCon bu instead connect directly to the spine. The advantage of this revision to the architecture of OSCON, so that it conforms to the structure of Pantome, is that by placing the various components ('Xcon') in the BGC their interrelatedness to each other, *and to other components* of, say, a morphological nature, is strengthened, but this is done via the sequentially organized segments in the spine.

4 Speech recognition

Speech recognition is another example of an application area that will benefit from utilization of the Pantome architecture. The development of Pantome has been implemented in parallel with an example application: text-to-speech conversion [21, 22, 23, 19, 16, 15, 20]. The diagram in figure 6 illustrates the typical text-to-speech conversion architecture found in the majority of modern implementations. This is a pipeline, data is passed from stage-to-stage with each step in the process being isolated from its neighbours and unable to contribute or benefit from direct communication. The Pantome architecture has been designed to overcome this and to allow the exploitation of the interrelatedness of the components, as described above.

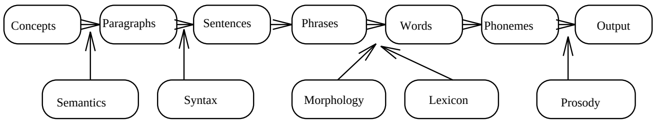


Figure 6: Architecture of a text-to-speech system

As the diagram in figure 7 illustrates, the typical architecture employed when implementing the speech recognition task is a pipe-line. This immediately raises the question of how Pantome can be beneficially employed in the implementation of such a system.

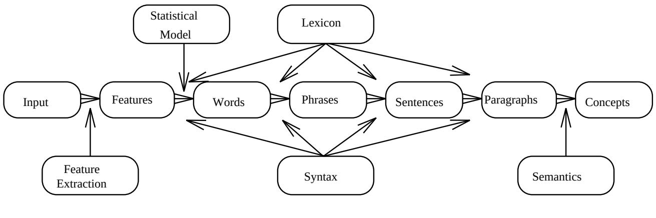


Figure 7: Architecture of a speech recognition system

The pipe-line architecture generally implies the repeated application of "re-write" rules, which waste much of the potential of the complex structures that underpins the surface structure. Utilizing a non-linear model that allows this structure to be represented for processing at all levels is a novel and potentially valuable solution to this problem.

5 Conclusion

We believe it will be possible to incorporate into the Pantome model segments for natural language processing from levels of syntax to levels of pragmatics in the framework of Rowe and Mc Kevitt []. In particular we briefly described how intentions are processed in the OSCON system and we hope that processing of other forms of pragmatics such as beliefs (see Ballim and Wilks [1, 2] and Wilks and Ballim [38]) can be integrated.

References

[1] A. Ballim and Y. Wilks. Stereotypical belief and dynamic agent modelling. In *Proceedings of the second international conference on user modelling*, 1990.

[2] A. Ballim and Y. Wilks. *Artificial Believers*. Lawrence Earlbaum Associates, Hillside, New Jersey, 1991.

[3] J.S. Brown, R.R. Burton, and A.G. Bell. SOPHIE: a step towards creating a reactive learning environment. *International Journal of Man-Machine studies*, 7:675–696, 1975.

[4] R. Burton. Semantic grammar: An engineering technique for constructing natural-language understanding systems. Technical Report 3453, Bolt, Beranek and Newman, Cambridge, MA, USA, 1976.

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

[6] 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.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] W.H. Edmondson and J.P. Iles. A non-linear architecture for speech and natural language processing. To appear in the Proceedings of the 1994 International Conference on Spoken Language Processing.

[16] W.H. Edmondson and J.P. Iles. Pantome: an architecture for speech and natural language processing. To appear in the Proceedings of the Institute of Acoustics 1994 Autumn Conference on Speech and Hearing.

[17] L. Guthrie, P. Mc Kevitt, and Y. Wilks. OSCON: An operating system consultant. In *Proceedings of the Fourth Annual Rocky Mountain Conference on Artificial Intelligence (RMCAI-89)*, pages 103–113, Denver, Colorado, USA, June 1989.

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

[19] J.P. Iles and W.H. Edmondson. Feature driven formant synthesis. To appear in the Proceedings of the Institute of Acoustics 1994 Autumn Conference on Speech and Hearing.

[20] J.P. Iles and W.H. Edmondson. Quasi-articulatory formant synthesis. To appear in the Proceedings of the 1994 International Conference on Spoken Language Processing.

[21] J.P. Iles and W.H. Edmondson. Control of speech synthesis using phonetic features. In R. Lawrence, editor, *Proceedings of the Institute of Acoustics Autumn Conference on Speech and Hearing*, volume 14, pages 369–373. Institute of Acoustics, December 1992.

[22] 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.

[23] J.P. Iles and W.H. Edmondson. Feature driven formant synthesis. Accepted for poster presention at the second ESCA/IEEE workshop on speech synthesis, 1994.

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

[25] W.J.M. Levelt. *Speaking: from intention to articulation.* MIT Press, 1989.

[26] P. Mc Kevitt. Building embedded representations of queries about unix. Technical Report MCCS-86-72, Computing Research Laboratory, New Mexico State University, Department 3CRL, Box 30001, New Mexico State University, Las Cruces, USA., 1986.

[27] P. Mc Kevitt. *Analysing coherence of intention in natural-language dialogue.* PhD thesis, Department of Computer Science, University of Exeter, Exeter, UK, September 1991.

[28] P. Mc Kevitt. Principles and practice in an operating system consultant. In D. Partridge, editor, *Artificial Intelligence and Software Engineering*, volume 1. Ablex Publishing Corporation, New York, 1991.

[29] P. Mc Kevitt, D. Partridge, and Y. Wilks. Analysing coherence of intention in natural language dialogue. Technical Report 227, Department of Computer Science, University of Exeter, Exeter. EX4 4PT. UK, 1992.

[30] P. Mc Kevitt, D. Partridge, and Y. Wilks. Experimenting with intention in natural language dialogue. Technical Report 234, Department of Computer Science, University of Exeter, Exeter. EX4 4PT. UK, 1992.

[31] P. Mc Kevitt, D. Partridge, and Y. Wilks. Why machines should analyse intention in natural language dialogue. Technical Report 233, Department of Computer Science, University of Exeter, Exeter. EX4 4PT. UK, 1992.

[32] P. Mc Kevitt and Y. Wilks. Transfer semantics in an operating system consultant: the formalization of actions involving object transfer. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence (IJCAI-87)*, volume 1, pages 569–575, August 1987.

[33] F. Pereira and D. Warren. Definite clause grammars for language analysis – a survey of the formalism and a comparison with augmented transition networks. *Artificial Intelligence*, 13:231–278, 1980.

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

[35] J. Rowe and P. Mc Kevitt. An emergent computation approach to natural language processing. In *Proceedings of the Fourth Irish Conference on Artificial Intelligence and Cognitive Science*, University College Cork, Ireland, September 1991.

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

[37] 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.

[38] Y. Wilks and A. Ballim. Multiple agents and the heuristic ascription of belief. In *Proceedings of*

the 10th International Joint Conference on Artificial Intelligence (IJCAI-87), pages 119–124, August 1987.