Input Specification in the WAG Sentence Generation System

Michael O'Donnell
Department of AI, University of Edinburgh,
80 South Bridge, Edinburgh. EH1 1HN, UK.
email: micko@aisb.ed.ac.uk
Keywords: realisation, grammar theory
Submission Type: Paper

February 4, 1996

Abstract

This paper describes the input specification language of the WAG Sentence Generation system. The input is described in terms of Halliday's (1978) three meaning components, ideational meaning (the propositional content to be expressed), interactional meaning (what the speaker intends the listener to do in making the utterance), and textual meaning (how the content is structured as a message, in terms of theme, reference, etc.).

1 Introduction

This paper describes the input specification language of the WAG Sentence Generation system. The input is described in terms of Halliday's (1978) three meaning components, ideational meaning (the propositional content to be expressed), interactional meaning (what the speaker intends the listener to do in making the utterance), and textual meaning (how the ideational content is structured as a message, in terms of theme, reference, etc.).

One motivation for this paper is the lack of descriptions of input-specifications for sentence generators. I have been asked at various times to fill this gap.

Perhaps a better motivation is the need to argue for a more abstract level of input. Many of the available sentence generators require specification of syntactic information within the input specification. This means that any text-planner which uses this system as its realisation module needs to concern itself with these fiddling details. One of the aims in the WAG system has been to lift the abstractness of sentence specification to a semantic level. This paper discusses this representation.

1.1 The WAG Sentence Generation System

The WAG Sentence Generation System is one component of the Workbench for Analysis and Generation (WAG), a system which offers various tools for developing Systemic resources (grammars, semantics, lexicons, etc.), maintaining these resources (lexical acquisition tools, network graphers, hypertext browsers, etc.), and processing (sentence analysis – O'Donnell 1993, 1994; sentence generation O'Donnell 1995b; knowledge representation – O'Donnell 1994; corpus tagging and exploration – O'Donnell 1995a).

The Sentence Generation component of this system generates single sentences from a semantic input. This semantic input could be supplied by a human user. Alternatively, the semantic input can be generated as the output of a multi-sentential text generation system, allowing such a system to use the WAG system as its realisation component. The sentence generator can thus be treated a black-box unit. Taking this approach, the designer of the multi-sentential generation system can focus on multi-sentential concerns without worrying about sentential issues.

WAG improves on earlier sentence generators in various ways. Firstly, it provides a more abstract level of input than many other systems (Mumble: McDonald 1980; Meteer et al. 1987; FUF: Elhadad 1991), as will be demonstrated throughout this paper. The abstractness improves even over the nearest comparable system, Penman (Mann 1983a; Mann & Matthiessen 1985), in its treatment of textual information (see below). Other sentence generators, while working from abstract semantic specifications, do not represent a generalised realiser, but are somewhat domain specific in implementation, e.g., Proteus (Davey 1974/1978); Slang (Patten 1986, 1988). Other systems do not allow generation independent from user interaction, for instance, Genesys (Fawcett & Tucker 1990) requires the user to make decisions throughout the generation process.

Against WAG, it does not yet have the grammatical and semantic coverage of Penman, FUF or Mumble, although its coverage is reasonable, and growing quickly.

1.2 Semantic Metafunctions

The input to the WAG Sentence generation system is a specification of an utterance on the semantic stratum. We thus need to explore further the nature of Systemic semantic representation. Halliday (1978) divides semantic resources into three areas, called *metafunctions*:

1. **Ideational Metafunction**: concerned with the propositional content of the text, structured in terms of processes (mental, verbal, material, etc.), the participants in the process (Actor, Actee, etc.), and the circumstances surrounding the process (Location, Manner, Cause, etc.).

2. **Interactional Metafunction**: viewing language as interaction, i.e., an activity involving speakers and listeners, speech-acts, etc. Interactional meaning includes the attitudes, social roles, illocutionary goals, etc of interactants.

3. **Textual Metafunction**: how the text is constructed as a message conveying information. This concerns, for instance, the thematic structuring of the ideation presented in the text, its presentation as recoverable or not, the semantic relevance of information, etc.

Although these metafunctions apply to both the semantics of sentence-size and multi-sentential texts, this paper will focus on sentential semantics, since we are dealing with the input to a sentence generator. Below we explore the nature of this semantic specification in more detail.

2 Interactional Specification

Interactional representation views the text as part of the interaction between participants. Sentences themselves serve an important part in interaction, they form the basic units - the *moves* – of which interactions are composed. Moves are also called *speech-acts*.

The input to the WAG generator is basically a speech-act specification, although this speech-act specification includes ideational and textual specification. Figure 1 shows a sample speech-act specification, from which the generator would produce: *I'd like information on some panel beaters*. The distinct contributions of the three meta-functions are separated by the grey boxes. *Say* is the name of the lisp function which evaluates the speech-act specification, calling the generator. *dialog-5* is the name of this particular speech-act – each speech-act is given a unique identifier, its unit-id.

In specifying the speech-act, there are several important things which need to be specified:

- **Speech-Function**: what does the speaker requires the hearer to do in regard to the encoded proposition?[1] This is called in Systemics the *speech-function*. Is the hearer supposed to accept the content as a fact? Or are they supposed to complete the proposition in some way? Or perform some action in response to the utterance?

- **Participants**: who is uttering the speech-act (the *Speaker*), and who is it addressed to (the *Hearer*). These roles are not important in every sentence-specification, but may be in some, for the following reasons:

  – **Pronominalisation**: If the filler of the Speaker or Hearer role happens

[1]For ease of writing, I use the terms 'Speaker' and 'Hearer' to apply to the participants in both spoken and written language.

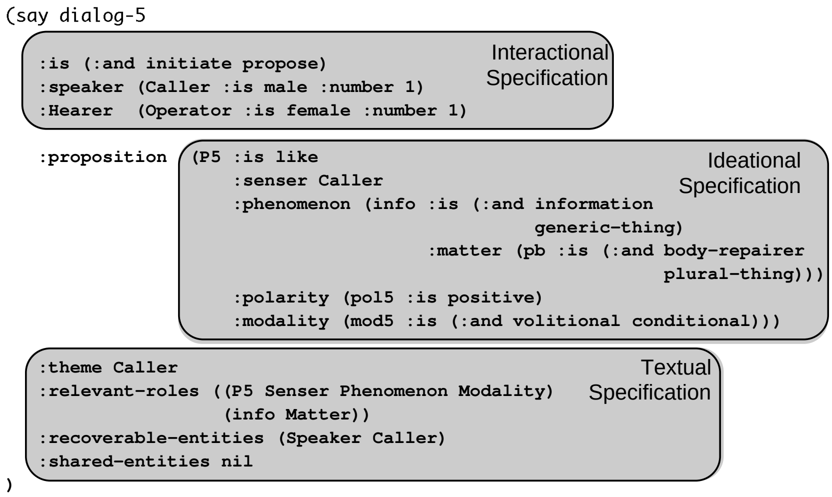

Figure 1: Typical Speech-Act Representation

to play some role in the ideational specification, then an appropriate pronoun will be used in the generated string (e.g., 'I', 'you').

- **Voice Selection**: If the spoken output mode is used, WAG will select a voice of the same gender as the speaker entity.
- **User Modelling**: In theory, the Speaker and Hearer fields are available for user-modelling purposes (cf. Paris 1993): lexico-grammatical choices can be constrained by reference to attributes specified in the Speaker and Hearer roles.[2] This has not, however, been done at present: while the implementation is set up to handle this tailoring, the resources have not yet been appropriately constrained.

- **Content**: what proposition is being negotiated between the speaker and hearer?

This form of semantic input improves over that of Penman in regards to the relationship between the speech-act and the proposition. In Penman, the ideational specification is central - a semantic specification is basically an ideational specification, with the speech-act added as an additional (and optional) field. This approach is taken because Penman was designed with monologic text in mind, so the need for varied speech-acts is not well integrated.

WAG however takes the speech-act as central, the semantic specification is a specification of a speech-act. The ideational specification is provided as a role of the speech-act (the *:proposition* role). WAG thus integrates with more ease into a system intended for dialogic interaction, such as a tutoring system. In particular, it simplifies the representation of speech-acts with no ideational content, such as greetings, thank-yous, etc.

2.1 Types of Speech-Acts

Figure 2 shows the systems of the speech-act network used in WAG (based on O'Donnell 1990). The main systems in this network are as follows:

- **Initiation**: The grammatical form used to realise a particular utterance depends on whether the speaker/writer is initiating a new exchange, or responding to an existing exchange (e.g., an answer to a question). Responding moves reflect a far higher degree of ellipsis than initiating moves. In particular, a move responding

[2] Since the fillers of the Speaker and Hearer roles are ideational units, they can be extensively specified for user-modelling purposes, including the place of origin, social class, social roles, etc of the participant. Relations between the participants can also be specified, for instance, parent/child, or doctor/patient relations. Lexico-grammatical decisions can be made by reference to this information: tailoring the language to the speaker's and hearer's descriptions.

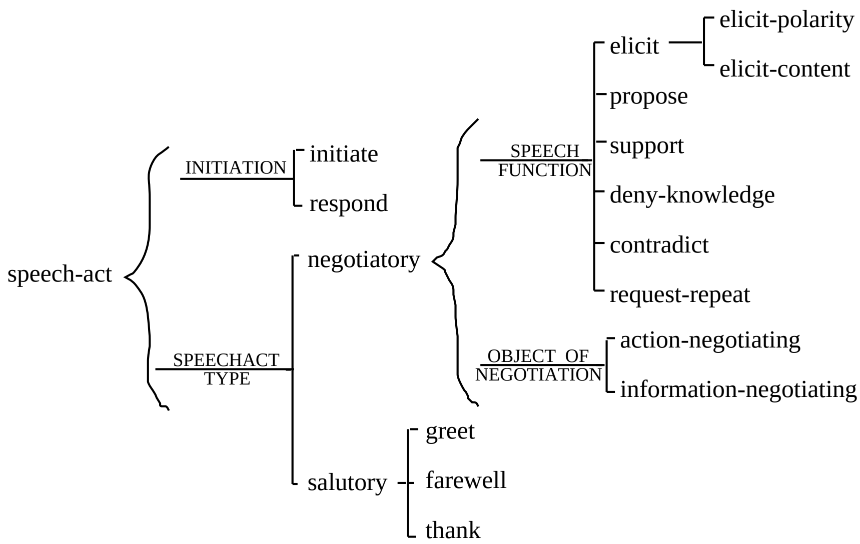

Figure 2: The Speech-Act Network

to a wh- question usually only needs to provide the wh- element in their reply.

- **Negotiatory vs. Salutory**: *negotiatory* speech-acts contribute towards the construction of an ideational proposition. while *salutory* moves do not, rather serving a phatic function, for instance, greetings, farewells, and thank-yous.

- **Speech Function**: The speech-function is the speaker's indication of what they want the hearer to do with the utterance. An *elicit* move indicates that the speaker requires some content-full response, while a *propose* move may require changes of state of belief in the hearer. *support* moves indicate the speaker's acceptance of the prior speaker's proposition. Other speech-functions cater to various alternative responses in dialogue, for instance: *deny-knowledge* - the speaker indicates that they are unable to answer a question due to lack of knowledge; *contradict*: the speaker indicates that they disagree with the prior speaker's proposition; *request-repeat*: the speaker indicates that they did not fully hear the prior speaker's move.

- **Object of Negotiation**: Speech-acts can negotiate *information* (questions, statements, etc.), or *action* (commands, permission, etc.). A move with features *(:and elicit negotiate-action)* would be realised as a request for action (e.g., *Will you go now?*), while a move with features *(:and propose negotiate-action)* would be realised as a command (e.g., *Go now!*).

In writing a speech-act specification, the *:is* field is used to specify the the speech-act type (the same key is used to specify the types of ideational units in the proposition). The speech-act of figure 1 is specified to have features *(:and initiate propose)*. Feature-specifications can be arbitrarily complex, consisting of either a single feature, or a logical combination of features (using any combination of *:and*, *:or* or *:not*). One does not need to specify features which are systemically implied, e.g., specifying *propose* is equivalent to specifying *(:and move speech-act negotiatory propose)*.

3 Ideational Specification

Once we have specified what the speech-act is doing, and who the participants are, we need to specify the ideational content of the speech-act. An ideational specification is a structure of entities (processes, things and qualities), and the relations between these entities.

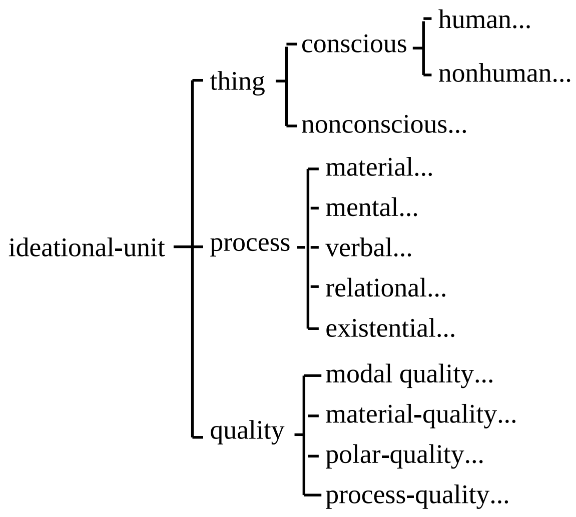

Figure 3: The Upper Model

3.1 Ideational Representation

When talking about ideational specification, we need to separate out ideational *potential* – the specification of what possible ideational structures we can have; and ideational *instantials* – actual ideational structures. The first is sometimes termed terminological knowledge – knowledge about terms and their relations, the second, assertional knowledge – knowledge about actual entities and their relations.

Ideational potential is represented in terms of an ontology of semantic types, a version of Penman's *Upper Model* (UM) (Bateman *et al.* 1990).[3] The root of this ontology is shown in figure 3. Many of the types in this ontology will have associated role constraints, for instance, a *mental-process* requires a *Sensor* role, which must be filled by a *conscious* entity. The UM thus constrains the possible ideational structures which can be produced.

The UM provides a *generalised* classification system of conceptual entities. For representing concepts which are *domain-specific* e.g., *body-repairer*), users provide domain-models, where domain-specific concepts are subsumed to concepts in the UM.

An ideational structure is specified by providing two sets of information for each entity (as in the propositional slot of figure 1):

- **Type Information**: a specification of the semantic types of the entity, derived from the UM, or associated domain-model.
- **Role Information**: a specification of the roles of the entity, and of the entities which fill these roles.

3.2 Generating from a Pointer into the KB

Typically, text generation systems are modularised between multi-sentential component (the text planner), and a component dealing with the realisation of sentences (the sentence generator). The communication between these components usually takes the form of semantic specification of a sentence. The multi-sentential module produces a number of sentence specifications, and the sentence generator constructs a sentence to express each of these.

Since both of these tasks are large, there has been a growing tendency for these two components to be totally separate from each other – text planners are built on top of stand-alone sentence generators. General-purpose sentence generators are thus becoming common, such as Penman, FUF, and Mumble. Each of these has been the platform supporting various text-planners (often experimental).

WAG has been designed to support this separation if need be – WAG can act as a stand-alone sentence generator. However, WAG functions best when integrated with the text-planner, at least to the extent that it has access to the same underlying KB.

One fundamental difference between WAG's input language, and that of Penman, involves the relation between the sentence specification and the knowledge-base (KB). These can be related in two ways:

1. **Input Specification Re-Expresses KB**: the semantic specification includes an ideational specification, which re-states

[3] WAG's Upper Model has been re-represented in terms of system networks, rather than the more loosely defined type-lattice language used in Penman. WAG thus uses the same formalism for representing ideational, interactional and lexico-grammatical information.

the contents of the KB, although the correspondence does not need to be 100%. This re-expression is performed so that the language of the input specification and the language employed in the KB can diverge. This allows the realisation system to work with multi-sentential generators regardless of the form of representation in the KB. The multi-sentential system translates information from the form used in the KB to the form acceptable in the input-specification. This approach also allows sentence-specifications to be written even if no KB is connected. The Penman sentence generator takes this approach to generation.

2. **Input Specification Points into KB**: in the second approach, ideational material is not included within the input specification. Rather, the input specification provides only a pointer into the attached KB. Since the information to be expressed is already present in the KB, why does it need to be re-expressed in the semantic specification? Taking this approach, the role of the semantic specification is to describe *how* the information in the KB is to be expressed, including both interactional and textual shaping.

Since WAG needs to work as a black-box in other multi-sentential generation systems, it is capable of working in the first (Penman-like) mode: an ideational specification can be embedded within the input-specification. This was the case in the example of figure 1.

However, WAG was designed with the second mode of generation in mind: to operate with a high-degree of integration between the knowledge representation system (KRS) and the sentence realiser. This integration allows economies of generation not possible where content used for text-planning and content used for sentence generation are represented distinctly. One benefit involves economy of code – many of the processes which need to be coded to deal with ideation for a text as a whole can also be used to deal with ideation for single sentences. Another involves the possibility of integrating the two processes – since the sentence realiser has access to the same knowledge as the multi-sentential planner, it can make decisions without requiring explicit informing from the planner.

```
; Participants
(tell John :is male :name "John")
(tell Mary :is female :name "Mary")
(tell Party :is spatial)

;Processes
(tell arrival
  :is motion-termination
  :Actor John
  :Destination Party)
(tell leaving
  :is motion-initiation
  :Actor Mary
  :Origin Party)

;Relations
(tell causation
  :is causal-relation
  :head arrival
  :dependent leaving)
```

Figure 4: Building a Knowledge-Base

To demonstrate this integrated approach to sentence generation, we show below the generation of some sentences in two stages – firstly, assertion of knowledge into the KB, and secondly, the evaluation of a series of speech-acts, which selectively express components of this knowledge.

3.2.1 Assertion of Knowledge into KB

Figure 4 shows the forms which assert some knowledge about John and Mary into the KB. The information basically tells that Mary left a party because John arrived at the party. *tell* is a lisp macro form used to assert knowledge into the KB.

3.2.2 Selective Expression of KB

Now we are ready to express this knowledge. The following sentence-specification indicates that the speaker is *proposing information*, and that the *leaving* process is to be the semantic head of the expression. It also indicates which of the roles of each entity are relevant for expression (and are thus expressed if possible), and which entities are identifiable in context (and can thus be referred to by name).

The generation process, using this specification, produces the sentence shown after the form.

```
(say example-1
  :is propose
  :proposition leaving
  :relevant-roles ( (leaving Actor)
                    (causation Head
                               Dependent)
                    (arrival Actor))
  :identifiable-entities (John Mary))

  =>  Mary left because John arrived.
```

As stated above, this approach does not require the sentence-specification to include any ideational-specification, except for a pointer into the KB. The realiser operates directly on the KB, using the information within the sentence-specification to tailor the expression.

Alternative sentence-specifications result in different expressions of the same information, for instance, including more or less detail, changing the speech-act, or changing the textual status of various entities. The expression can also be altered by selecting a different entity as the head of the utterance. For instance, the following sentence- specification is identical to the previous, except the *cause* relation is now taken as the head, producing a substantially different sentence:

```
(say example-2
  :is propose
  :proposition causation
  :relevant-roles ((causation Head
                              Dependent)
                   (leaving Actor)
                   (arrival Actor))
  :identifiable-entities (John Mary))

  =>  John's arrival caused
            Mary to leave.
```

We will now turn to the textual component of the input specification, which is responsible for tailoring the expression of the ideational content.

4 Textual Specification

Textual semantics concerns the role of the text and its components as a *message*. While creating a text (whether a single utterance or a whole book), we have a certain amount of content we wish to encode. But there are various ways to encode this information, to present our message. The textual semantics represents the various strategies for structuring the message.

4.1 Relevant-Roles

One of the main steps in the text generation process involves content selection – the selection of information from the speaker's knowledge base for presentation. Such a process must decide what information is relevant at each point of the unfolding discourse.

In some systems, content selection is driven through the construction of the rhetorical structure of the text (e.g., Hovy et al. 1992). As we build a rhetorical structure tree, the ideation which is necessary for each rhetorical relation is selected. For instance, if we add an *evidence* relation to an existing RST tree, the ideation which functions as evidence is selected for expression. The rhetorical structure thus organises the ideational content to be expressed, selecting out those parts of the ideation-base which are relevant to the achievement of the discourse goals at each point of the text. I use the term *rhetorical relevance* to refer to this sort of relevance.[4]

Rhetorical relevance is dynamic – it changes as the text progresses. It represents a shifting *focus* on the ideation base (Halliday & Matthiessen, 1995, pp373-380). What is relevant changes as the text unfolds, as the rhetorical structure is realised. Relevance forms what Grosz (1977/86) calls a *focus space*.[5] Halliday & Matthiessen (1995) extend Grosz's notion of focus space to include other types of textual spaces: thematic spaces, identifiability spaces, new spaces, etc. (p376). Each of these spaces can be though of as a pattern stated over the ideation base (p373).

According to Grosz, focus is "that part of the knowledge base relevant at a given point of a dialog." (p353). However, Grosz's notion of relevance is based on the needs of a text understanding system – which objects in the

[4]See Pattabhiraman & Cercone (1990) for a good computational treatment of relevance, and its relation to salience.

[5]Various earlier linguists and computational linguists have also used the notion of 'spaces' to represent textual status, see for instance, Reichman (1978); Grimes (1982).

knowledge-base can be used to interpret the utterance. My sense of relevance is derived from relevance in generation – what information has been *selected* as relevant to the speaker's unfolding discourse goals. She is dealing with a set of objects which may *potentially* appear in the text at this point, while I am dealing with the set of objects which most probably *do* appear in the text.

To represent the relevance space in a sentence specification, I initially provided a *:relevant-entities* field, which listed those ideational entities which were relevant for expression. However, problems soon arose with this approach. Take for instance a situation where Mark owns both a dog and a house, and the dog destroyed the house. Now, we might wish to express a sentence to the effect that *A dog destroyed Mark's house*, which ignores Mark's ownership of the dog. In a system where relevance is represented as a list of entities, we could not produce this sentence.

What we need is a representation of the *relevant relations* in the KB. To this end, WAG's input specification allows a field *:relevant-roles*, which records the roles of each entity which are currently relevant for expression, e.g., as was used in the examples of section 3.2.2.[6]

While constructing a sentence, the sentence generator refers to this list at various points, to see if a particular semantic role is relevant, and on the basis of this, chooses one syntactic structure over another.

EXAMPLE

4.2 Theme

The *:theme* field of the speech-act specifies the unit-id of the ideational entity which is thematic in the sentence. If a participant in a process, it will typically be made Subject of the sentence. If the Theme plays a circumstantial role in the proposition, it is usually realised as a sentence initial adjunct. WAG's treatment of Theme needs to be extended to handle the full range of thematic phenomena. Theme specification in WAG is identical to that used in Penman.

[6] If the explicit ideational specification is included in the *say* form (as in figure /reffig:say1), then the relevance space need not be stated, it is assumed that all the entities included within the specification are relevant, and no others.

4.3 Information Status

The participants in an interaction each possess a certain amount of information, some of which is shared, and some which is unshared. I use the term *information status* to refer to the status of information as either shared or unshared.

The information status of ideational entities affects the way in which those items can be referred to. Below we discuss two dimensions of information status:

1. **Shared Entities**: entities which the speaker believes are known to the hearer can be referred to using *identifiable reference*, e.g., *definite deixis*, e.g., *the President*; and *naming*, e.g., *Ronald Reagan*. Entities which are not believed to be shared require some form of indefinite deixis, e.g., *a boy called John; Eggs; Some eggs*, etc. A speaker uses indefinite deixis to indicate that he believes the entity not to be known to the hearer. It is thus a strategy used to introduce unshared entities into the discourse. Once the entity is introduced, some form of definite reference is appropriate.

2. **Recoverable Entities**: Entities which are part of the *immediate discourse context* can be referred to using *pronominalisation*, e.g., *she, them, it, this*, etc. *substitution*, e.g., *I saw one;* or *ellipsis* (the non-mention of an entity), e.g., *Going to the shop?*. The immediate discourse context includes entities introduced earlier in the discourse; and entities within the immediate physical context of the discourse, e.g., the discourse participants (speaker, hearer, or speaker+hearer) and those entities which the participants can point at, for instance, a nearby table, or some person.

Two fields in the semantic specification allow the user to specify the information status of ideational entities – and thus how they are referred to in discourse:[7]

[7] These lists will typically be maintained by the text-planner as part of its model of discourse context.

- The Shared-Entities Field: a list of the ideational entities which the speaker wishes to indicate as known by the hearer, e.g., by using definite reference.

- The Recoverable-entities Field: a list of the ideational entities which are recoverable from context, whether from the prior text, or from the immediate interactional context (e.g., the speaker and hearer).

EXAMPLE

5 Conclusions

The input specification for the WAG sentence generator is a speech-act, which includes an indication of which relations in the KB are relevant for expression at this point. Other information in the input specification helps tailor the expression of the content, such as an indicator of which KB element to use as the head of the generated form, which is theme, which elements are recoverable and identifiable.

In taking this approach, WAG attempts to extend the degree to which surface forms can be constrained by semantic specification. In many sentence generation systems, direct specifications of grammatical choices or forms is often needed, or, in the case of Penman, the user needs to include arcane *inquiry preselections* – interventions in the interstratal mapping component, perhaps more arcane than grammar-level intervention.

By providing a more abstract form of representation, text-planners using WAG do not need to have any knowledge of grammatical forms, and can spend more of their efforts dealing with issues of text-planning.

Although WAG has extended the level at which surface forms can be specified semantically, there are still gaps. To allow for this, WAG allows input specifications to directly constrain the surface generation, either by directly specifying the grammatical feature(s) a given unit must have, or alternatively, specifying grammatical defaults: grammatical features which will be preferred if there is a choice.

The advantages of WAG's input specification language are summarised below:

1. **Interactional Specification**: WAG's input specification allows a wider range of specification of the speech-act type than used in Penman and other sentence-generation systems. Also, the specification of the Elicited element in wh-questions is made more abstract. By placing the proposition as a role of the speech-act, rather than visa-versa,....

2. **Ideational Specification**: WAG allows two modes of expressing the KB – in one mode, each sentence specification is a self-contained specification, containing all the ideational information needed (the 'black-box' mode). In the other, a sentence specification contains only a pointer into the KB, allowing finer integration between text-planner and sentence realiser.

3. **Textual Specification**: WAG introduces a high level means of representing the textual status of information to be expressed. Following Grosz (1977/86), and Halliday & Matthiessen (1995), I use the notion of textual spaces, partitionings of the ideation base, each of which shifts dynamically as the discourse unfolds. I have outlined:

   - a *relevance space*: the information which is rhetorically relevant at the present point of the discourse;
   - a *shared-entity space*: the information which is part of the shared knowledge of the speaker and hearer.
   - a *recoverability space*: the information which has entered the discourse context, including the entities which have been mentioned up to this point in the discourse. Information in the recoverability space can be presumed, or pronominalised.

While the WAG generator has only been under development for a few years, and by a single author, in many aspects it meets, and in some ways surpasses, the functionality and power of the Penman system, as discussed above. It is also easier to use, having been designed to be part of a Linguist's Workbench –

a tool aimed at linguists without programming skills.

The main advantage of the Penman system over the WAG system is the extensive linguistic resources available. Penman comes with a large grammar and semantics of English (and other languages). WAG comes with a medium-sized grammar of English.[8]

6 Bibliography

Bateman, John, Robert Kasper, Johanna Moore & Richard Whitney 1990 "A General Organisation of Knowledge for Natural Language Processing: the Penman Upper Model", USC/Information Sciences Institute Technical Report.

Davey, Anthony 1974/1978 Discourse Production: a computer model of some aspects of a speaker, Edinburgh University Press, Edinburgh, 1978. Published version of Ph.D. dissertation, University of Edinburgh, 1974.

Elhadad, Michael 1991 "FUF: The Universal Unifier User Manual Version 5.0", Technical Report CUCS-038-91, Columbia University, New York, 1991.

Fawcett, Robin P. – Gordon H. Tucker (1990) "Demonstration of GENESYS: a very large semantically based Systemic Functional Grammar". In *Proceedings of the 13th Int. Conf. on Computational Linguistics (COLING '90).*

Halliday, M.A.K. 1978 Language as social semiotic. The social interpretation of language and meaning. London: Edward Arnold.

Halliday, M.A.K. & Christian Matthiessen 1995 *Construing experience through meaning: a language-based approach to cognition.* Pinter: London.

Mann, William C. 1983 "An Overview of the Penman Text Generation System", USC/ISI Technical Report RR-84-127.

Mann, W. C. – C. I. M. Matthiessen (1985) "Demonstration of the Nigel Text Generation Computer Program", In Benson and Greaves, 1985.

Martin, James R. 1992*English Text: system and structure*. Amsterdam: Benjamins.

McDonald, D. 1980 *Language Production as a Process of Decision-making under Constraints*, MIT Ph.D. Dissertation, 1980. MIT Report.

Meteer, M., D. McDonald, S. Anderson, D. Forster, L. Gay, A. Huettner, & P. Sibun. 1987 "Mumble-86: Design and Implementation", COINS Technical Report 87-87, University of Massachusetts at Amherst, Computer and Information Science.

O'Donnell, Michael 1990 "A Dynamic Model of Exchange" in Word, vol. 41, no. 3 Dec. 1990

O'Donnell, Michael 1993 "Reducing Complexity in a Systemic Parser ", in *Proceedings of the Third International Workshop on Parsing Technologies*, Tilburg, the Netherlands, August 10-13, 1993.

O'Donnell, Michael 1994 *Sentence Analysis and Generation - A Systemic Perspective.* Ph.D., Department of Linguistics, University of Sydney.

O'Donnell, Michael 1995a "From Corpus to Codings: Semi-Automating the Acquisition of Linguistic Features", in Proceedings of the AAAI Spring Symposium on Empirical Methods in Discourse Interpretation and Generation, Stanford University, California, March 27 - 29.

O'Donnell, Michael 1995b "Sentence Generation Using the Systemic WorkBench", in Proceedings of the Fifth European Workshop on Natural Language Generation, 20-22 May, Leiden, The Netherlands, pp 235-238.

Pattabhiraman, T. 1993 *Aspects of Salience in Natural Language Generation.* Ph.D. Dept. of Computing Science, Simon Fraser University.

Patten, Terry (1986) *Interpreting Systemic Grammar as A Computational Representation: A Problem Solving Approach to Text Generation*, Ph. D. dissertation, University of Edinburgh.

[8] While the WAG system can work with the grammar and lexicons of the Nigel resources, the resources which map grammar and semantics in Nigel are in a form incompatible with WAG).