Sentence Generation Using the Systemic Workbench

Michael O'Donnell

GMD/Institut für Integrierte Publikations- und Informationssysteme
Dolivostraße 15
64293 Darmstadt, Germany
email: *mick@darmstadt.gmd.de*

1 Introduction

The Systemic Workbench for Analysis and Generation (WAG: O'Donnell 1994a, 1994b) is an application for working with Systemic Grammars. Various processes are supported including:

- **Single Sentence Generation**: The user provides a semantic-specification of a sentence, from which a sentence is produced (see below).
- **Parsing**: The user provides text (running text accepted) and the system provides a grammatical analysis of each sentence in turn (multiple analyses where the sentence is grammatically ambiguous).
- **Lexical Acquisition Tool**: A window-based tool for the acquisition of new lexical items, or for the modification of existing ones.
- **Grapher**: Allows graphing of the various data-structures in the system (system networks, systemic structures, either conceptual or grammatical). See figure 1 below for a sample graph of a system network.
- **HyperText Resource Explorer**: A tool which generates a card for each resource object (system, feature, function, lexeme, unit, etc.), displaying the information associated with that object. Clicking on the information displayed in these cards will produce a card describing these objects.
- **WAG-KRL**: a systemic-based knowledge-representation language (re-implementing many aspects of the Loom Knowledge Representation System (MacGregor & Bates 1987), although from a Systemic perspective), for representing semantic, or grammatical, knowledge.

WAG is available for distribution, at no charge. It requires Macintosh Common Lisp to run (a version for Suns is under development). The distribution package includes manuals, binaries, and a small resource-set for English.

2 Semantic Input

WAG's sentence generator has been developed to replicate the functionality of Penman, although there is no code in common between the two systems. Because of this, WAG's semantic input is similar to Penman's SPL, with some differences. For instance, the following semantic-specification would result in the generation of the sentence: "I'd like information on some panel beaters".

```
(say dialog-5
   :is (:and initiate propose)
   :Speaker (Caller :is human :number 1)
   :Hearer  (Operator :is human :number 1)
   :Proposition
     (P5 :is like-process
         :sensor Caller
         :phenomenon (info :is (:and information
                                     generic-thing)
                           :matter (pb :is panel-beater
                                       :number 2))
         :polarity (pol5 :is positive)
         :modality (mod5 :is (:and volitional
                                   conditional)))
   :theme Caller
   :relevant-entities (P5 info pol5 Caller pb))
```

This form improves on Penman's SPL in the following ways:

- The representation is speech-act-based, not ideation-based. Penman has been designed with monologic text in mind, so the need for varied speech-acts is not well integrated. An SPL is basically an ideational specification of a sentence, with the speech-act added as an additional (and optional) field. A WAG specification includes the ideational specification only as a field of the speech-act specification. This approach improves WAG's integration into a system intended for dialogic interaction, such as a tutoring system.
- In a WAG semantic-specification, the :proposition slot is allowed to be just a pointer into the knowledge-base, rather than a complete ideational specification. The rest of the information in the semantic-specification (speech-act and textual information) tailors the expression of the indicated conceptual structure.

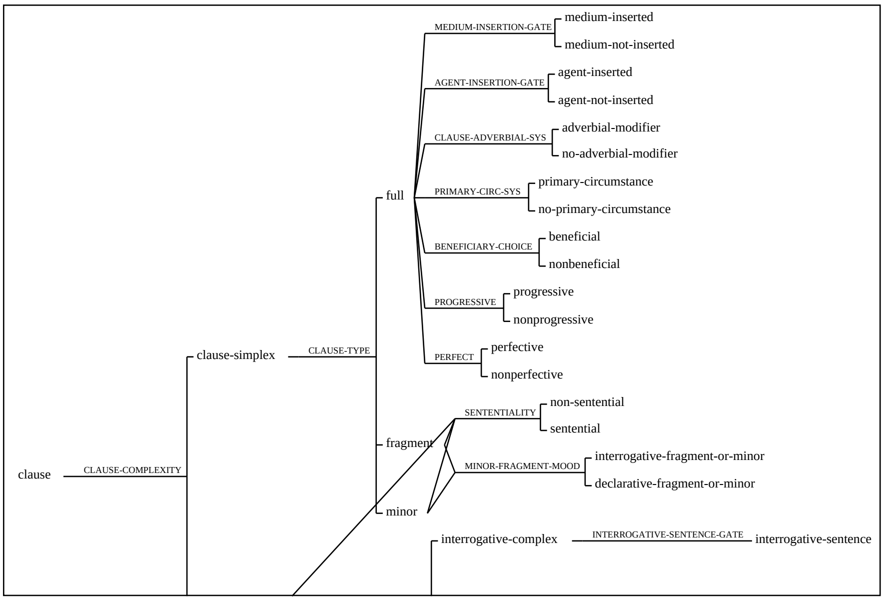

Figure 1: A Partial Graph of the Clause Sub-Network

- Penman allows specification of textual information by direct specification of inquiry responses (preselections into the interstratal mapping process). WAG represents textual specifications in a more abstract and theoretically-based manner. WAG maintains three variables which represent textual fields of *relevance*, *identifiability*, and *recoverability*. Each of these variables contains a set of conceptual entities, which represents, for example, the set of currently relevant entities. The interstratal mapping constraints can query whether a particular ideational entity is in one of these sets, and thus determine appropriate realisation. Semantic representation is thus freed from the particular labels used in the interstratal mapping component, which is not the case with Penman.

3 Sentence Generation

To generate a sentence, one just evaluates a semantic representation. The sentence string, graphologically formatted appears. The user can also use the "Generation Interface" to step through the generation process (see Figure 2).

3.1 Improvements Over Penman

Apart from the resource maintenance tools which make WAG an easier system to develop grammars on, WAG's generation system improves on Penman in two major ways. Firstly, Penman's Semantic-Grammar mapping rules (the Chooser-Inquiry interface) has been replaced with a constraint-based approach, whereby each grammatical feature has an associated semantic constraint which must be met for that feature to be chosen in generation (a formalism originally developed by Bob Kasper, unpublished). This move to a declarative representation allows the Systemic mapping resources to be used for both analysis and generation, which was not possible with the Chooser-Inquiry approach. Also the feature-based approach makes it easier to modify the resources, since these constraints map directly from grammatical features to the semantic-form, while using Penman, one needs to look from feature to system to chooser to inquiry to semantic-form.

Secondly, WAG uses the same formalism for representing ideational (conceptual) networks/structures and also for grammatical networks/structures. The same unificational processes can be used for all levels of representation. Penman uses Loom for ideational potential, and knowledge representation, an internal representation for SPLs, and another form for grammatical networks.

3.2 Viewing Generated Sentence

The default output of sentence generation is just the generated text. However, we can also view the generated sentence structure in terms of a printed functional representation of the sentence. Alternatively, we can view a graph of the generated sentence structure, as shown in figure 3. Another option is to explore the structure in a hypertext manner using the Resource Explorer.

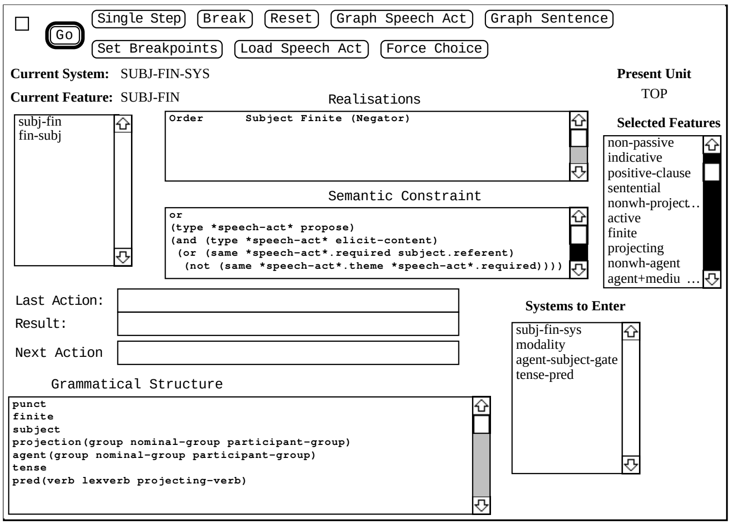

Figure 2: WAG's Generation Interface

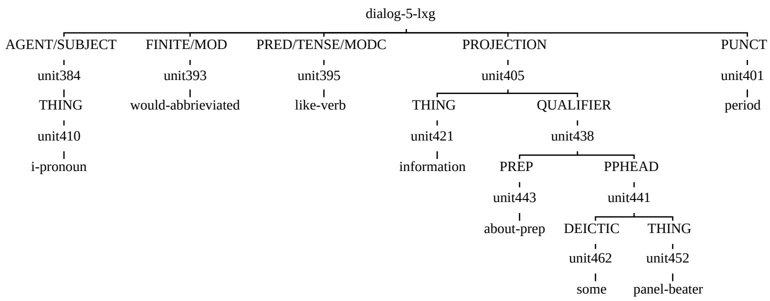

Figure 3: The Generated Sentence Structure

4 Summary

The WAG system offers an easy-to-use Systemic Grammar Development Environment, allowing easy modification and viewing of the resources. WAG can also be used as a sentence generation component in a multi-sentential text generation system, or a dialogue system. At present, because of the limited linguistic coverage of the supplied resource models, it may function better as a computational linguistics teaching tool, although the resource size is quickly approaching usable coverage.

5 Bibliography

MacGregor, Robert & Raymond Bates 1987 The LOOM Knowledge Representation Language, Technical Report ISI/RR-87-188/Information Sciences Institute.

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

O'Donnell, Michael 1994b *Workbench for Analysis and Generation: User Manual*, WAG System Documentation.