Chapter 1

Introduction:
Integrating Diverse Descriptions

1. Problems of Integration in Modeling

The goal of this thesis has not been to solve a specific problem, but rather to integrate all the different parts of a linguistic system, in a single self-consistent framework. This is important to ensure the compatibility of the solutions found for specific problems. In contrast, many theses highlight a single problem and explore all aspects of the issue. Such works are necessary, as they take us up the path of knowledge one hard step at a time. If all theses were of this type, though, we would have a problem in that in each specific area we tackle, we need to make various assumptions about the model as a whole. However, when we move to a different area, a different problem, we tend to make another set of model-theoretic assumptions which are more convenient to solving the problems in that area. It is only when we try to integrate the two areas of work that we see that the two solution-spaces are incompatible. While working on a grammatical problem, we can say of an exception "that is handled in the semantics!", but when working on a semantic problem, we say "that is handled in the grammar!".

This is not to say the thesis does not include original material, but rather, where problems are solved, the solutions are grounded in the wider framework. Problems might not be explored to the depth which is often desired, but that is not the goal of this thesis.

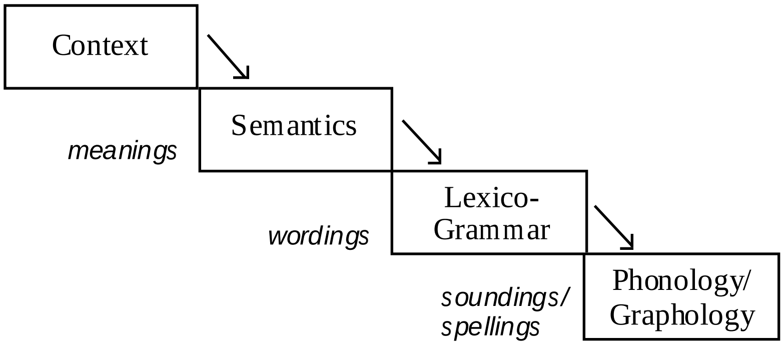
The breadth of this thesis does however need to be limited. To this end, I limit myself to the linguistics of the single sentence. I will not look at linguistic structuring which crosses sentence boundaries. I will be looking at the sentence as a graphological, lexico-grammatical and semantic object.

Systemic-Functional Linguistics (SFL - Halliday 1961; Hudson 1971; Berry 1975/77; Fawcett 1980; Butler 1985; Martin 1992) will be used, since it offers a theory well suited to integration, given its emphasis on modularisation (stratification and meta-functional layering). SFL has also proven very useful in both parsing and analysis, as shown by the large body of computational work using the formalism (e.g., Winograd 1972; Patten 1986, 1988; Davey 1974/78; Mann & Matthiessen 1985; Kasper 1988a; Fawcett & Tucker 1990; Cross 1991).

I will be exploring integration across several dimensions:

- **Theory and Implementation**: It is one thing to put forward one's thoughts on paper, and another to implement those thoughts as a computer program. Linguists often call implementations linguistically trivial, even those programs at the forefront of technology. There is a real gap between where linguists can go in their minds and where implementation can follow. Implementation lags behind the continually expanding theory.

  On the other hand, implementers often look at non-computational linguistics and find it "fuzzy", and lacking in detail. Implementation requires a completeness and explicitness, which a theoretician would find tedious. Much of theoretical



.c.'Figure 1.1: The Strata of a Systemic Model'

linguistics has not yet reached a level of understanding at which it becomes implementable.

One goal of this thesis has been to close the gap between theory and implementation, relating a theoretical model of Systemic-Functional linguistics to a particular implementation which closely follows the model. The implementation is called the WAG system -- the *Workbench for Analysis and Generation*.

- **Process and Resource:** The main dimension of the linguistic system to be explored involves the distinction between linguistic *resources* (which describe well-formed utterances) and linguistic *processes* (which describe *how* the resources can be used). In regards to resources, I am concerned with graphological, lexico-grammatical, and semantic representations of sentences, and the integration of these resources. While descriptions of Systemic resource models are common (cf. Halliday 1985; Berry 1975/77; Hudson 1971), integrated overviews of Systemic processing are lacking. This thesis is intended to fill that lack, describing both analysis and generation of sentences.
- **Resource Modularity**: As one means to handle the complexity of language, Systemic Linguistics makes strong use of modularisation -- division of the resource model into several self-contained but inter-related layers of description. Several kinds of modularity are used:

  a) Stratal Modularity: Systemics views language as a stratified system, treating language on number of strata: Context, Semantics, Lexico-grammar and Phonology/Graphology. Stratification allows language to be structured along a number of different dimensions -- as doings, meanings, wordings and soundings (see Halliday 1973). These distinct layers of representation are not independent from each other but display regular co-occurrence -- particular doings are encoded in particular meanings, particular meanings are encoded in particular wordings, and particular wordings through particular soundings. A complete description of language thus needs also to state these inter-stratal correspondences. [1]

  b) Meta-Functional Modularity: Halliday further modularises each stratum along functional grounds: Context in terms of Field, Tenor and Mode; Semantics in terms of Ideational, Interactional and Textual metafunctions; and Lexico-grammar into Transitivity, Mood and Theme. I will only explore the semantic modularity. These metafunctional layers are:

  - Ideational Meaning: the text as representation of experience, the propositional content of the text.

[1] This thesis does not deal with either Context or Phonology. For more details on the Context stratum, see for instance, Halliday (1978), Halliday & Hasan (1985) or Martin (1992). For more detail on Systemic Phonology, refer to Halliday (1970) or Mock (1985).

  - Interactional Meaning: the text as interaction, between speaker/writer and the audience. This includes the speaker's intrusion on the content, his/her attitudes, desires, etc. The term 'interpersonal' is often used here also.
  - Textual Meaning: the text as a message, concerning how the text is structured to communicate the ideational and interactional meanings.

- **Process Integration:** While there have been several Systemic systems for sentence *generation* (Davey 1974/78; Mann & Matthiessen 1985; Patten 1986; Fawcett & Tucker 1990; Cross 1991), and a few for *analysis* (Winograd 1972; McCord 1977; Cummings & Regina 1985; Kasper 1988a, 1988b, 1989; O'Donoghue 1991a, 1991b; Weerasinghe & Fawcett 1993), there has been few attempts at a Systemic system performing both *analysis* and *generation*. in the same program[2]. It is here that the problem of model over-specialisation is most apparent -- the resources developed for generation have proven difficult to use for analysis without substantial modification[3], and after modification, these resources are no longer suitable for generation. My goal in this thesis has been to develop a resource model usable for both generation and analysis -- a truly bi-directional system.

In summary, this thesis is intended to offer a model of sentences which is both theoretical and implementable, for both processing and representation of language. It is equally applicable to generation or analysis, grammar and semantics.

Each of these dimensions of integration represents a modularisation of the problem space. This in itself offers one of the best means for handling the problems of dealing with large systems -- dividing the system up into smaller, well-defined sub-components. Each sub-component, being smaller, is easier to manage than the whole. Each module can make use of a different sub-set of methods or assumptions. Some part of the integration problem remains, in that we need to relate the inter-dependencies of these modules to each other.

1.1 Terminology

I will first define some terms which are perhaps problematic:

- **Speaker & Listener**: I use the terms *speaker* to designate the producer of an utterance, regardless of whether the text is realised as speech or writing. Similarly, *listener* is used to indicate the intended addressee of a text, whether written or spoken.
- **Graphology**: To describe the structuring of text in terms of sentences, words, characters, etc., I use the term *graphology* rather than the more common *orthography*. This term is more aligned with the alternative realisation plane: *phonology*.
- **Representational Levels**: The Systemic resource model includes several types of linguistic levels: *strata* (levels of representation), *rank* (levels of constituency within a strata), and *layer* (simultaneous functional descriptions within each stratum -- see chapter 2). I use the term *level* as a generalisation over these three terms.

[2] While the generator of Mann and Matthiessen (1985) and the parser of Kasper (1988a) share the same lexico-grammatical resource (the Nigel grammar), the two programs use a completely distinct set of processes. Bateman *et al.* (1992) however reports some experimental work in which a fragment of a Systemic grammar is re-represented in a Typed Feature Structure (TFS) formalism (cf. Emele & Zajac (1990)). The TFS system can do limited bi-directional processing. Note however, that this system does not 'parse' sentences, but needs to be provided with a partial systemic lexico-grammatical analysis, which the system then augments using the Systemic resources. The grammar fragment is also very small.

[3] Kasper (1988a) forms an exception here -- once he has translated the grammar into the form used for parsing (using the Functional Unification Grammar (FUG) formalism), the grammar can still be used for generation (although no processes have been provided for this). The parsing grammar used by Weerasinghe & Fawcett (1993) was not automatically compiled from the GENESYS generation grammar, but is hand-translated.

2. Single Sentence Representation and Processing

2.1 Defining Sentence

Webster’s Dictionary defines a *sentence* as:

> “A word or group of words which states, commands or exclaims something. It usually includes a subject and a predicate, and is conventionally written with a capital letter at the beginning, and ends with a punctuation mark (period, question mark, etc.).” (The New Lexicon Webster’s Dictionary of the English language, Lexicon Publications: New York 1987, pp908).

This definition, while not necessarily totally accurate, reflects one important fact about the sentence -- it is a unit whose definition draws upon:

semantics: “states, commands or exclaims something”;

lexico-grammar: “It usually includes a subject and a predicate”, and

graphology: “A word or group of words ... conventionally written with a capital letter at the beginning, and ends with a punctuation mark (period, question mark, etc.)”.

For the purposes of this thesis, a sentence is primarily a graphological unit, defined in terms of words and punctuation marks on the page (or words and intonation in speech). The lexico-grammar and semantics constrain the valid sequences of words and punctuators in the sentence, but a sentence is not an object itself on the lexico-grammatical and semantic strata.

There are however *typical correspondences* between the sentence and units on these strata -- the move (or speech-act) in the semantics, and the clause (sometimes a clause-complex) in the lexico-grammar. A sentence typically expresses a single speech-act (called a move in Systemics), such as a question, statement or command. Exceptions include two speech-acts in a single sentence: “Today is Monday and what do we do today?” (a teacher to her class).[4] Similarly, the correspondence between clause and sentence is only typical. We have frequent examples in dialogue of fragmentary sentences, e.g.,

Mary:	*Where are you going to?*
John:	*London.*

John replies with a single word, a nominal group at the lexico-grammatical level. One approach, which I follow, sees this not as an isolated nominal group, but rather as a clause with most of the structure elided since it is recoverable. In that case, the sentence is also a fragmentary clause, and the clause-sentence correspondence holds in this case. However, see Sefton (1992) for a convincing example of non-correspondence.

I occasionally use the term *utterance*, rather than sentence. An utterance is the phonological equivalent of a sentence, and I use the terms interchangeably.

2.2 Sentence Processing

Sentence processing involves either the analysis, or the generation, of sentences, and sometimes both together. In the analysis direction, we start with a sentence and derive some more abstract representation, and in the generation direction, we start with an abstract representation, and produce a sentence expressing it. Sentence processing can thus be seen as a means of mapping between various representations of the sentence.

[4] Example brought to my attention by Imagen Hunt (University of Sydney).

A system for sentence processing is useful for several reasons:

1) **As a test-bed for linguistic theory & description**: Sentence Processing systems offer the linguist a means to develop their theories through rigorous testing. Computational systems require an explicit statement of the grammar, forcing the linguist to examine all aspects of the resource. The analyses produced by the parser, and the sentences produced by the generator, will point out problems in the resource model. The WAG system has been used to develop and test various parts of a Systemic formalism, including semantics, lexico-grammar, lexicon designs and inter-stratal mapping descriptions.

2) **As a module in larger systems**: Sentence generation and analysis systems are integral components of larger scale systems. For instance,

   - Multi-Sentential Generation Systems: Systems which generate paragraphs of text, for instance, generating a set of sentences describing the contents of a database (e.g., McKeown 1985; Paris 1987).
   - Text Summarisation Systems: Systems which analyse each sentence, extract out in some sense the important content, and generate sentences to express this. Note that many systems use only basic graphological and/or lexical analysis to discover the important parts of the text (e.g., first sentence of each paragraph, or highlighted text) (cf. Tait 1982; Fum *et al.* 1985).
   - Natural Language Interfaces: Systems which allow a human to interact with a machine using natural language, for instance, to access information from a database. These systems require both sentence analysis (to interpret the human's input), and sentence generation (to produce the machine's output) (cf. Jacobs 1985; Perrault & Grosz 1986).
   - Machine Translation: Systems which take language input in one system and produce natural language text or speech in another language. These systems potentially perform sentence analysis on the input[5], and use the analysis to produce sentences in the second language (cf. Nirenburg 1986).

3. Process & Resource

A linguistic theory consists of two sorts of information -- *resource* information (the static knowledge of language, e.g., the lexicon, the well-formedness conditions of grammar and semantics, and the mapping relations between them), and *process* information (concerning *how* the resource knowledge is used, e.g., how the static knowledge is used to parse or generate text)[6]. Figure 1.2 illustrates how process and resource are two integral parts of our linguistic knowledge.

[5] Some systems may perform translation with only the minimum analysis of the input, e.g., word for word translators.

[6] The terms *process* and *resource* have entered the computational community during the eighties (cf. Bateman 1991; Paris & Maier 1991), but it is difficult to pin down their source. Winograd (1983) talks of linguistic resources. Kay (1985) pushed the concept of process/resource separation, without using these terms. There has long been discussion of declarative vs. procedural representation of grammars (cf. Winograd (1975)).

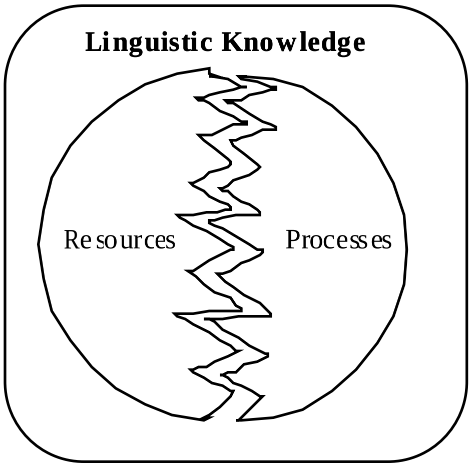


Figure 1.2: Process and Resource

With rare exception, linguists have dealt almost entirely with linguistic resource. The grammars that are set to paper are entirely a statement of the static well-formedness conditions of language. Linguistic process has been an invisible entity[7]. While each new generation of linguists have been taught how to use their grammars, rarely have these descriptions of linguistic process been committed to paper. It has been taught mainly through demonstration -- teacher demonstrates technique to student until student can replicate technique. Linguistic theory has thus been taught through two channels -- the explicit, written presentation of language descriptions (resources) and the implicit, personal communication of process knowledge.

The advent of computational linguistic since the fifties, however, has forced the explicit representation of linguistic process. Computers cannot be taught by demonstration, so they must be told explicitly how to use the grammar. This explicit statement of linguistic process is given to the computer in the form of a computer program. The program directs the computer in its use of the grammar to achieve the desired process[8].

To address this imbalance, this thesis will attempt to set out a model of the processes involved in using a Systemic theory and formalism. Note however that I am not attempting to model the way Systemicists use the formalism. Rather, I will describe the way in which the Systemic formalism can be used computationally, particularly in regards to two processes -- analysis and generation.

The theory/implementation division introduced in section 1 can be applied to both resource and process. Resource theory concerns the nature of the resources, the formalism, the nature, content and relation between strata. Resource implementations are actual descriptions of language. These descriptions draw on the resource theory in terms of the formalism used, the criteria for positing linguistic units and relations, etc.

Process theory concerns higher level issues about how processing takes place (e.g., intermixed *vs.* pipeline processing), how to evaluate different approaches (e.g., measuring complexity), various basic algorithms (e.g., system network traversal algorithms), etc. A process implementation is a computer program to process a resource description, drawing on the process theory.

The term ‘model’ can be seen as problematic, since it is ambiguous between a theoretical model (formalism, or stratal model, etc.) and a descriptive model (a resource implementation). I will avoid the use of the term in this thesis, unless prefixed with the modifiers ‘theoretical’ or ‘descriptive’.

[7] I acknowledge the seeming process nature of transformation rules. Still, *how* these rules are used is usually left implicit.

[8] Note however that the process/resource distinction was not made clearly until the seventies.

4. Declarativisation: Process-Resource Separation

A computational linguistic system involves the implementation of both linguistic resources and processes. In many systems, the designers do not clearly distinguish the two types of knowledge -- the program represents resources and processes in the program itself -- the resources are *procedurally* implemented. For instance, a sentence parser might consist of the following instructions:

1) Recognise an NP,

2) Recognise a VP,

3) Return the NP^VP structure as a sentence.

Procedural implementations have their problems -- the grammar can only be used for one process (e.g., analysis). They are also difficult to modify except by programmers. It is hard to enforce formal restrictions on the grammar, since the writer of the grammar has available the full power of the programming language in which to encode the grammar.

One approach which has been gaining increasing popularity over the last decade involves the *declarativisation* of the resource component of the model (cf. Kay 1985). The linguistic resources are provided to the program separately. The program (or programs since multiple processes can share the resource) loads these resources before processing begins.

The main advantage of the declarative approach is re-usability: a resource can be used for a number of different processes, e.g., for generation and for analysis. The resource can even be exported to a new system, using a different programming language. This is important to our goals of integrating analysis and generation in a single system. Also, a declaratively defined resource is not written in a programming language, so can be extended/modified by a linguist without programming experience.

One of the main advantages of the WAG system is that it has taken a strict declarative approach -- resource descriptions are declarative, and the processes themselves do not make any assumptions about the resource description (except that they conform to the supported Systemic formalism). The program can thus work with several different resource descriptions, e.g., a mini-grammar for experimental purposes, and a more comprehensive grammar (e.g., the Nigel grammar) for practical usage. Resource descriptions for different languages can also be handled, to allow generation or analysis in multiple languages.

5. The WAG System

The bulk of the research behind this thesis has involved the design and implementation of a computer program to analyse and generate single-sentences using the Systemic formalism. This system is called the WAG system, for *Workbench for Analysis and Generation*. This thesis is largely a description of the modeling issues which arose during this work, and of the computational algorithms which were developed to process Systemic descriptions. Reference will be made to this system, and to the grammar it contains.

Reference will also be made occasionally to the Penman system, a single-sentence generation system developed at the Institute of Information Sciences (USC/ISI), Los Angeles (Mann & Matthiessen 1985). The linguistic model of WAG is based on the linguistic model inherent in Penman, although it differs in substantial ways. My work has been inspired by the limitations of Penman, and the desire for something better. WAG has also borrowed some aspects of Penman’s generation *process*, in particular, the way Penman lets the grammar, rather than the semantics, control generation (this will be discussed later).

I originally started working with the grammar from Penman -- the Nigel grammar (see Matthiessen 1985). This is a large computational implementation based heavily on Halliday (1985). The grammar is the result of contributions from several linguists, principally Christian Matthiessen during his ten year stay at ISI. I eventually found this grammar too large and complex for parsing development (the larger the grammar, the slower the processing). While it *could* be used, the system was too slow for practical use. Because of this, I have developed my own, smaller, grammar for use in the system. This grammar, apart from having a smaller coverage, is designed to be less complex for processing by removing some of the redundancy. This will be discussed in more detail in chapter 2.

The WAG system is more than just a sentence analyser/generator. It has been designed as a grammar development platform. One of the main problems when working with the Penman system is finding how any change in one part of the grammar will affect other parts of the grammar. To simplify the task of grammar development, several tools have been implemented:

1) **A Systemic Grapher**: presents the various data-structures (system networks, structural representations, etc.) used in the system in a graphical form, for easy viewing.

2) **Resource Explorer**: allows access to the resource model through a hyper-card like interface. On call, the system produces a description of each object in the resource model (system, feature, function, lexical-item, etc.). Clicking on any field of the description will produce a description of that object.

3) **Generation and Parsing Debugging Interfaces:** WAG provides step-through interfaces for both sentence generation and analysis, allowing the user to discover exactly where the process is going astray, typically due to mistakes in the resource model.

6. Contributions of this Work

Probably the main contribution of this thesis is that it fills a documentation gap -- this is the first description of a computational system for Systemic generation *and* analysis. While there have been various descriptions of Systemic generation systems (Davey 1974/78; Mann & Matthiessen 1985; Patten 1986; Fawcett & Tucker 1990; Cross 1991), and of Systemic analysis (Winograd 1972; McCord 1977; Cummings & Regina 1985; Kasper 1988a, 1988b, 1989; O'Donoghue 1991a, 1991b; Weerasinghe & Fawcett 1993), there has not to date been a system which performs both analysis and generation[9]. This thesis fills the void by providing an integrated view of Systemic sentence processing in general. Even outside of Systemics, detailed reports of bi-directional systems are rare, and this thesis should prove valuable.

In terms of original work, this thesis reports original research in several areas, including ground-breaking work in Systemic analysis. Each of the analysers mentioned above has been limited in some way, either resorting to a simplified formalism (Winograd; Cummings & Regina; McCord; Weerasinghe & Fawcett), or augmenting the Systemic analysis by initial segmentation of the text using another grammar formalism (Kasper: Phrase Structure Grammar; Bateman *et al.*: Head-driven Phrase Structure Grammar; O'Donoghue: his own ‘Vertical Strip Grammar' (VSG)). Except for the work reported in this thesis, there has not been a parser which uses the full Systemic formalism, without help from another formalism.

There has long been discussion of inter-stratal mapping in Systemics, with little formalisation of the discussion. This thesis offers a detailed account of inter-stratal mapping between semantics and grammar. While credit for the inter-stratal formalism

[9] See however, footnote 2 above.

belongs to Robert Kasper (unpublished), the WAG system represents its first implementation (apart from a very prototypical version in the Penman system). The formalism has developed throughout its implementation.

In regards to the representation of ideational and interactional information for computational use, this thesis has improved on the existing Systemic work by making the relations between these meta-functional components more explicit. Systemics does not address the issue of how the ideational content of an utterance is related to the speech-function. The WAG system also improves on the computational implementations of Systemics in this regard. WAG also makes advances in the computational representation of textual meaning, representing themacity, relevancy, recoverability and identifiability of information.

The major result of the research behind this thesis has perhaps been the WAG program itself. This is a tool available to the Systemic community to facilitate their resource development, and for use in teaching environments, allowing students to experience SFL interactively.

7. Thesis Structure

This thesis presents an integrated overview of a linguistic system, implemented computationally. The overview is limited to sentence modeling to reduce the scope of the discussion, although this still represents a large enough area to allow problems of integration to become apparent.

This introduction has discussed the basic issues behind the thesis, starting with the problem of integrating single-issue models when one attempts to build a general-purpose system. Some modeling tools to aid in the integration have been discussed, particularly process/resource separation, declarativisation, and modularisation.

The body of the thesis is structured around the process/resource distinction, and is thus divided into two parts: part A discusses the resource model, while part B discusses the process model.

Each of these parts is structured to reflect the internal modularity. Part A consists of five chapters, discussing, in turn:

- An overview of the sentence resource model, describing the graphological, lexico-grammatical and semantic strata (chapter 2);
- Three chapters describing the three layers of the micro-semantic stratum -- ideational resources, interactional resources, and textual resources (chapters 3, 4 and 5);
- A description of the inter-stratal mapping formalism, which maps semantic and lexico-grammatical resources (chapter 6).

Part B takes up the description of sentence processing, using five chapters:

- A discussion of some global issues in sentence processing (chapter 7),
- A description of the Systemic feature logic system which underlies the WAG system (chapter 8);
- Two chapters exploring sentence analysis, chapter 9 on parsing strategies, and chapter 10 describing the WAG analyser;
- A description of the generation component of the WAG system (chapter 11);

Chapter 12 then concludes the thesis with a summary of the work, drawing conclusions, and pointing towards future work.