Sentence generation and systemic grammar: an introduction

John A. Bateman

Written for: *Iwanami Lecture Series: Language Sciences*, Volume 8. Tokyo: Iwanami Shoten Publishers. (Appears in Japanese).

Abstract. This paper provides a first introduction to some basic issues in sentence generation. It describes how systemic-functional grammars deal with the problems of sentence generation and gives a brief overview of the use of systemic-functional grammars in sentence generation research.

Contents:

1 What is sentence generation?

Sentence generation is the computational process of automatically producing sentences in some human language on the basis of a specification of communicative intention. A sentence generation component receives as input a specification of what it is supposed to communicate, and produces as output a corresponding natural language expression. In most current approaches to sentence generation, such input is either a *logical form* or a set of facts in some *knowledge representation language* . Thus, a generation component might accept as input a logical form such as:

$$\text{on}'(x, y) \wedge \text{book}'(x) \wedge \text{red}'(x) \wedge \text{table}'(y)$$

and convert this into something like the natural language expression "The red book is on the table".

The end result of such a generation process is relatively clear: a sentence must be produced which is grammatically correct and which has made appropriate choices of words (or *lexemes*). The basic components of a sentence generator are, therefore, as in analysis or syntactic parsing , a *grammar* and a *lexicon*. The earliest sentence generators to be built (cf. Yngve 1962, Friedman 1969) used only these minimal components. This restricted them to the random generation of grammatically correct sentences since specifications of intended communicative purpose were not present. This is of rather limited use—for example, for testing grammars for correctness. Subsequent generation programs therefore attempted to 'cover', in some sense, an intended meaning given as input. The precise form of this input is still, however, an area of considerable debate. This is quite the reverse of the situation in syntactic parsing where it is the input to the analysis process—a string of characters or sequence of sounds—that is clear.

The input for generation presents a problem precisely because a 'bare logical form' of the type used as an illustration above seriously underconstrains the kind of sentence that must be produced. For example, it might be required in some contexts that the sentence generator produce not the example sentence above but instead "It is the red book that is on the table", "The red book on the table" or "The book on the table is red". Moreover, it might not even be clear just how much of the input is to be packed into single sentences; another possible variation is "There is a book on the table. It is red." Aspects of this problem have been usefully discussed as the 'logical-form problem' by, for example, Shieber (1993, p188) and McDonald (1993). It is not in general possible to provide a sufficiently unique input specification in terms of logical forms. The issues involved here make contact with questions of pragmatics and functional approaches to language to which we return below.

The first treatments of semantic specifications within the context of generation were based on semantic networks (Simmons & Slocum 1972, Shapiro 1979). The suggestive relation sometimes drawn between semantic networks and cognitive models of knowledge made it natural to add to the input specification information concerning 'topic' or 'focus' (e.g., where in the network current attention is to be located) and 'areas of interest' (partitions): these additions could be put to good use for specifying the communicative intention more precisely. It was already clear that specifications of focus and a partitioning into given and new information play a crucial role for the generation process.

Sentence generation has since come to be pursued within two broad orientations. The first was a direct outgrowth of the early semantic network experiments. Here the semantic network expresses the knowledge of an information system and the generation task uses that knowledge as one basis for forming its utterances whenever communicative goals are to be fulfilled. Sentence generation of this kind is the principal component of *text generation*—now also commonly called *natural language generation* (NLG). The central question studied in NLG is how one can program computers to produce high-quality natural language texts from some computer-internal representation of information. Motivation for this study ranges from the entirely theoretical (linguistic, psycholinguistic) to the entirely practical (to produce information presentation components for computer programs). The focus of linguistic attention is predominantly on the *text* rather than on the sentence—most crucial is to understand how sentences are constructed in order to fit together to create a text rather than as the expressions of isolated semantic specifications. We illustrate this below.

The second orientation has developed more recently and may be considered as an *analysis-oriented view*. Here the generation process is seen as essentially the reverse process of syntactic parsing—the kind of input to the generation process is assumed to be equivalent to the kinds of output produced by analysis programs. This concentrates attention on issues familiar from syntactic parsing: syntactic structure, the computational properties of the process (e.g., does it always terminate with all inputs), and the guarantee that the sentence produced corresponds *only* to the semantic input (i.e., it neither entails less nor more semantic content than the input). The most well known and established algorithm for this style of generation is the 'semantic head-driven generation' algorithm of Shieber, van Noord, Pereira & Moore (1990). Sentence generation within this orientation remains an endeavor for establishing more sophisticated mappings between a semantics (representation of propositional content) and sets of possible surface strings (sentences, phrases, etc.). Linguistic attention is focused primarily on the grammatical unit *sentence*. An overview of some of the key issues for this kind of generation work is given by van Noord & Neumann (1996); however, as much of the distinctive nature of sentence generation research that we will see below is not found in this kind of approach, we will in this article focus mostly on the first, original orientation.

Both the research field of NLG and the techniques for information presentation that it provides are finding ever more possibilities of application. Established examples include the generation of weather reports from meteorological data (Kittredge, Polguère & Goldberg 1986) and the generation of letters responding to customer queries (Springer, Buta & Wolf 1991, Coch, David & Magnoler 1995). Systems that can be applied in areas such as the automatic production of technical documentation (cf. Reiter, Mellish & Levine 1995, Rösner & Stede 1994), of instructional texts (cf. Not & Stock 1994, Paris, Linden, Fischer, Hartley, Pemberton, Power & Scott 1995), of patent claims (cf. Sheremetyeva, Nirenburg & Nirenburg 1996), of computer-supported cooperative work (cf. Levine & Mellish 1994), of patient health information and education (cf. Cawsey, Binsted & Jones 1995, DiMarco, Hirst, Wanner & Wilkinson 1995), of medical reports (Li, Evens & Hier 1986) and in natural language interfaces to databases (cf. Androutsopoulos, Ritchie & Thanisch forthcoming) and information systems (cf. Bateman & Teich 1995) are now real possibilities. The relevance of NLG for work on machine translation has also recently become clearer in approaches that place more of the responsibility of appropriate target language sentences on generation components rather than on accurate syntax-based transfer rules (cf. Whitelock 1992, Copestake, Flickinger, Malouf, Riehemann & Sag 1995). Moreover, a trend that will undoubtedly increase dramatically in the near fu-

ture is the use of NLG to produce World-Wide Web pages on demand on the basis of user queries and available information: prototype systems supporting such 'synthetic documents' are already available (cf. Gruber, Vemuri & Rice 1995, Milosavljevic & Dale 1996).

The requirement that NLG systems are able to create appropriate texts of the distinct kinds and varieties just suggested raises a different set of concerns to those of natural language understanding. Indeed, the view that generation is "basically the reverse process of understanding" (e.g., Engelien & McBryde 1991, p32) is quite mistaken. It is partially as a consequence of this misunderstanding that NLG has become a disjoint subfield of computational linguistics with its own central techniques, problems and issues. Moreover, the particular demands of the generation task have a crucial influence on the desirable organization and properties of sentence generation components. It is therefore necessary to see something of the aims of NLG in order to understand the criteria that are to be employed when deciding between possible sentence generation procedures.

These considerations have determined the basic organization of this article. First, we will see in more detail some of the issues that are raised by the need to generate *texts* rather than sentences. Second, we will see how this can be handled effectively by organizing grammars around the notion of *choice*. This leads directly to a discussion of the other main topic of the article: *systemic grammar*; this form of grammar is particularly suited to the generation task and so has been more widely adopted in NLG systems than any other linguistic approach. We can then be more concrete about the input specifications necessary for generation systems, showing just what information must be provided in addition to the simple propositional content contained in traditional logical forms. A brief overview of generation systems employing systemic grammar is then given, followed by a consideration of formalisations of systemic grammar and its relations to *constraint-based* approaches to grammar—one of the most widespread approaches to language processing found outside of generation . The article ends with a sketch of future directions for both sentence generation and systemic grammar combined.

2 Putting sentences together into texts

The requirements of sentence generation, as the basic component of any NLG system, are naturally shaped by the main concerns of NLG; these concerns may be illustrated by the following examples of generated texts taken from core text generation research.

The first example consists of a pair of texts output by the generation system PAULINE (Hovy 1988). The motivations for these and some other texts, which all concern the outcome of a hypothetical U.S. primary election, are described in more detail by Hovy (1987).

(1) a. Carter and Kennedy were the candidates in a primary in Michigan on 20 February. Carter lost to Kennedy by 335 votes. At present, Kennedy has a better chance of getting the nomination than before. Carter is also closer to getting the nomination than before. Both Carter and Kennedy want to get the nomination.

b. I am pleased to inform you that Carter has improved his chances of winning the nomination. At the present time, Carter has many more delegates than he had in the past; also, Carter has many more than Kennedy does.

Both texts were generated fully automatically with respect to a single, common body of information represented in a knowledge representation language. The differences in the texts were controlled by selecting parameters defining the *context* of the texts to be generated. Text (1a) is the result of parameter settings for an objective attitude to the text's subject matter, as described to an acquaintance, with neither interlocutor having strong opinions about the election's outcome. In contrast, text (1b) has contextual parameter settings that reflect a more positive attitude of the speaker towards Carter and a more formal setting, as in giving a public lecture. The social distince between the speaker and hearer(s) is therefore assumed here to be greater. Significant is that the texts are substantially different in their chosen syntactic forms, their selected lexical items, and even in the information they present. Any simple approach to re-expressing a logical form more or less one-to-one as a corrresponding sentence would fail here completely.

The second example consists of three contrasting sentences generated by the Penman text generation system (Mann & Matthiessen 1985). The generation system is being used here as the natural language output component for a research expert system for digital circuit design (Bateman & Paris 1989). The Penman system was the first extensively developed general purpose generation system using systemic grammar; it will be described in more detail below.

(2)

a. The system is faulty, if there exists a O in the set of the output terminals of the system such that the expected value of the signal part of O does not equal the actual value of the signal part of O and for all I in the set of the input terminals of the system, the expected value of the signal part of I equals the actual value of the signal part of I.

b. The system is faulty, if all of the expected values of its input terminals equal their actual values and the expected value of one of its output terminals does not equal its actual value.

c. The system is faulty, if the inputs are fine and the output is wrong.

Again, all three texts were generated with respect to a common set of facts in a knowledge representation language—in this case a predicate calculus-like internal representation used for inferencing by the expert system (most closely re-expressed in text (2a)). The generation system was responsible for expressing these facts differently depending on the specifications of the intended contexts that were given. Text (2a) is illustrative of texts for expert system *developers* who are concerned with the exact internal representation employed; text (2b) is for reasonably sophisticated *users* of the expert system who are concerned with the digital circuit behaviour not the internal detail of the expert system; and text (2c) is for naive users of the expert system who are not interested in technical details of any kind.

Even within texts of a particular text type, aimed at a particular audience group, it is still essential that the sentences produced 'fit together'. Violation of the rules of textual development quickly results in texts that are, at best, unnatural sounding and, at worst, unintelligible. The constructed text shown in (3), used as an example by Halliday (1978, p134), illustrates what can happen if things go wrong: which they might well do as long as generation remains random rather than controlled. Larger sentence grammars as used in NLG cover the range of syntactic variation deployed in this text: it is therefore essential that the sentence grammar is able to select appropriately from the possibilities on offer.

(3) Now comes the President here. It's the window he's stepping through to wave to the crowd. On his victory his opponent congratulates him. What they are shaking now is hands. A speech is going to be made by him. 'Gentleman and ladies. That you are confident in me honours me. I shall, hereby pledge I, turn this country into a place, in which what people do safely will be live, and the ones who grow up happily will be able to be their children.

In this text every sentence involves at least one textually motivated error. The result is that the text is seriously deficient: any native speaker of English would reject it utterly. Note, however, that the kinds of errors are not such that they introduce *grammatical* problems: each sentence is, in the correct textual context, an appropriate sentence and so needs to be covered by a generation sentence grammar. One type of error in the text is the ordering of elements: as in many languages, English uses ordering to express textual importance of various kinds (thus "Now comes the President here" wrongly places the textually more prominent temporal information first in the clause rather than last in the clause as would be natural in this context; similarly with the initial subject matter "On his victory"). Another type of error is the wrong use of textual focusing forms such as active/passive (e.g., there is no motivation in the text for passivization of "A speech is going to be made...), extrapositions of various kinds (e.g., "What they are shaking now...", "in which what people do safely", "That you are confident in me..."). Any error of these kinds threatens to reduce a text to unintelligibility.

The examples shown in (1)–(3) make it clear that one of the main goals of the traditional NLG enterprise is to explain *what motivates the selection of different forms of expression.* NLG proper is not so much concerned with the generation of *a* text, but more with the generation of *the* text that is most appropriate for its context. Two principal requirements are then:

- the options taken up in consecutive sentences must be *textually consistent* with respect to their lexical selections, granularity, style of expression—it would be unacceptable, for example, if a text suddenly switched from using general terms to using very specific, possibly technical terms, or switched from using rather polite forms of expression to very familiar expressions;
- the syntactic forms selected must serve to build a developing text rather than a set of unrelated sentences—each new contribution to the text changes the state of knowledge of the reader/hearer, and subsequent sentences in the text both indicate how they relate to the current knowledge state (via pronouns, ellipsis, focusing constructions, etc.) and how they extend that knowledge (via connectives such as *Moreover*, *But*, *In contrast*, etc.).

This goes beyond mapping a given logical form to a corresponding surface string—which would be the natural reverse of the main goal of sentence parsing in natural language understanding. An NLG component must be able both to *select* appropriate logical forms and to provide additional information that constrains the generation process to pick just the natural language expression of these logical forms required to produce natural texts. A sentence generation component must therefore be able to interpret the additional *textual* information provided. Languages typically offer a rich range of possibilities for creating text, and these resources must be utilized appropriately by a successful NLG system. This is different to the aims of natural language understanding where, most often, there is an attempt to *neutralize* different forms of surface expression that a language may provide for the same logical information. A shown most clearly in (3), this would be disastrous for generation.

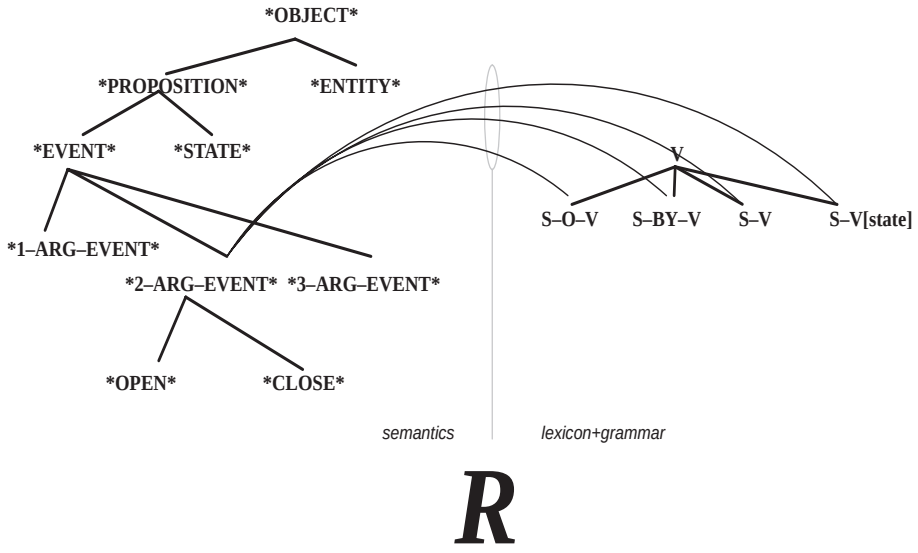

Figure 1: Mapping between meaning and form
(adapted from: Emele, Heid, Momma & Zajac 1992)

3 Generation as 'functionally motivated choice'

We have noted that the fact that NLG represents a different set of concerns to the more commonly met areas of natural language understanding and 'parsing' also has consequences for the kind of approaches to sentence generation that are most typically adopted. We now make the task and problem of generation more concrete. Figure 1 shows one possible graphical representation of the task. On the left-hand side we have a hierarchy of semantic concepts of a kind typically found as the higher organizing nodes in current knowledge ontologies defined in Artificial Intelligence and Knowledge-based systems (cf. Patil, Fikes, Patel-Schneider, McKay, Finin, Gruber & Neches 1992, Bateman 1992). On the right-hand side we have a range of possible linguistic realizations for those concepts; these linguistic realizations are defined in terms of classes similar to 'subcategorization' types found in, e.g., Head-driven Phrase Structure Grammar (HPSG: Pollard & Sag 1987). The generation task is then to determine and computationally instantiate the relation R between meaning and form.

The first and simplest approach to the problem might consider all the possible natural language sentences that express a given semantic category. The mapping between semantics and surface string could consist of a simple disjunction of relations. This might be adequate for a refinement of the 'random generation' task: here we would specify a semantic specification and could produce nondeterministically possible corresponding sentences. For example, a semantic specification shown for illustration as (4a) could be linked with the disjunction of possible forms shown in (4b).

(4) a.

$$\texttt{sem:} \begin{bmatrix} \texttt{process:} & *\text{OPEN}* \\ \texttt{actor:} & *\text{SPEAKER}* \\ \texttt{affected:} & *\text{WINDOW}* \end{bmatrix}$$

b. [S-O-V "I opened the window]
∨ [S-BY-V "The window was opened by me"]
∨ [S-V "The window was opened"]
∨ [S-V [state] "The window was open"]

Although this correspondence can, as indicated in the figure, be generalized to hold between the semantic supertype *2-ARG-EVENT* and the general subcategorization types, it is still insufficient for sentence generation proper. We do not yet have the means of picking *the* sentence structure that is appropriate for a particular text and particular place in that text that it is to appear.

It is necessary to extend this view of the generation process by imposing additional organization on the simple disjunction of possible mappings reflected in (4). In particular, we can attempt to organize the mapping around the *communicative functions* that the structural realizations achieve. In other words, we try to add information concerning when it is appropriate to use one of the possible ways of expressing a concept rather than another. A straightforward first approximation is shown in (5).

(5)
- •asserting / questioning / ordering:
  e.g., *the window is open*, *is the window open?*, *open the window!*
- •positive / negative:
  e.g., *the window is open*, *the window is not open*
- •not evaluated / evaluated (e.g., modalized):
  e.g., *the window is open*, *the window might be open*
- •express 1 place relation / express 2 place relation:
  e.g., *the window was open*, *I opened the window*
- •express 2 place relation with agency / ... without agency:
  e.g., *I opened the window*, *the window was opened*
- •foreground the agent / background the agent:
  e.g, *I opened the window*, *the window was opened by me*
- •foreground the affected / do not foreground the affected:
  e.g., *The window I opened*, *I opened the window.*

This provides a more 'natural' upward connection from the grammatical possibilities to the semantics: It is far more likely that an information system knows whether it wants to assert something or to question something than it is to know what grammatical structure it wants to use. Similarly, in terms of text planning—the step of NLG that precedes sentence generation—it is usual that text plans concern themselves with these kinds of communicative purposes rather than with issues of grammatical structure. Such a classification then enables the generation process to separate out the disjunctive mapping in Figure 1 between the concept *2-ARG-CONCEPT* and the four possible realizations illustrated in (4) by using supporting constructs such as the foregrounding and backgrounding of various semantic elements.

While this replaces a larger set of equally ranked disjunctions with a smaller set of functionally labelled disjunctions, or 'choices', a number of important questions remain. For example: is it possible to make the categories that appear—e.g., 'agency', 'foregrounding', 'backgrounding', 'evaluation', etc.—more precise? This is particularly important for a computational account. And: is it possible to provide further organization? A list of labelled choices is in itself not that

much of an improvement over the original disjunction. It is in the pursuit of these questions that the main differences between sentence generation and natural language understanding in their approaches to grammar and lexical selection can be found. It is also here, therefore, that we can locate some of the most distinctive shaping influences on the process and resources of sentence generation.

First, we can readily observe (for any given language) that there *are* dependencies between the kinds of functional choices listed in (5). It is not possible to consider all of the choices equally: for example, neither an expression of evaluation (*the window might be open*), nor the selection of an expression of a state of affairs or of a backgrounded agent (*the window is open*, *the window was opened*), is readily combinable with giving an order (*open the window!*). If a generation system knows that it needs to give an order, therefore, some of the options for foregrounding/backgrounding and other modulations are not available. This indicates one possible benefit of the organization in terms of functional choices—the search space of available forms of expression can be quickly restricted. This approach is taken to its logical conclusion with systemic grammar, which we introduce in the following section.

Second, when the kinds of approaches to grammar adopted in natural language understanding work concern themselves with 'alternations' of the kind illustrated in (5), they necessarily do so primarily from a *structural* perspective. They focus on the differences in *syntactic structure* that are exhibited. As an illustration, Figure 2 shows examples of descriptions of the 'active/passive' alternation for English in various structural approaches. All of these treatments of passive can be seen to focus primarily on the syntactic structural distinctions between active and passive sentences. Only the HPSG account explicitly mentions the semantics of the expressions being related and only to state that it is unchanged across the passive alternation: again, neutralizing the distinction semantically. The crucial question for NLG—i.e., when might one use an active sentence and when a passive sentence, what *differences in communicative effects* they might have—is not then actively addressed.

Because of this difference in focus, it is common for approaches to NLG and sentence generation to draw more strongly on the results and methodologies of a different linguistic tradition to that found in analysis-oriented work. Rather than adopting the structural/generative paradigm made dominant by American linguistics, NLG looks equally, if not more, to the *functional* linguistic paradigm including text linguistics, functional linguistics, and pragmatics in its broadest sense. Approaches to linguistics from this perspective consider the relation of language to social and psychological factors as determinative of its organization. It is usual within this paradigm to investigate the conditions of use of particular expressions found in natural languages and this is precisely what is necessary for a sentence generation program that is to generate sentences appropriate to their contexts of use. The functional linguistic paradigm includes a very broad and active range of work—often not formal, and usually with no thoughts of computational application. For computational sentence generation, however, it is necessary not only to find functional descriptions but also to find such descriptions that may be made sufficiently precise as to be embodied in computer programs. This restricts the kinds of approach that are most usefully adopted.

Transformational Grammar: Passive transformation

SD:	NP,	[+V, +AUX],		[+V, -AUX],	NP	
	1	2		3	4	
SC:	4	2	BE+EN	3	BY 1	

GPSG: (Gazdar, Klein, Pullum & Sag 1985, p59) Passive metarule:

$$\begin{array}{c} \text{VP} \rightarrow W, \text{NP} \\ \Downarrow \\ \text{VP [PAS]} \rightarrow W, (\text{PP } [by]) \end{array}$$

LFG: (Bresnan 1982, p20) *Passivization*

Operation on lexical form:	(SUBJ) $\mapsto \phi$ / (BY OBJ) (OBJ) $\mapsto$ (SUBJ)
Morphological change:	$V \mapsto V_{[\text{Part}]}$

HPSG: (Pollard & Sag 1987, p215) PASSIVE:

$$_{base \wedge trans}\left[\begin{array}{ll} \text{PHON} & \boxed{1} \\ \text{PAST-PART} & \boxed{2} \\ \text{SYN|LOC|SUBCAT} & \langle \ldots, []_{\boxed{3}}, []_{\boxed{4}} \rangle \\ \text{SEM|CONT} & \boxed{5} \end{array}\right] \mapsto {}_{passive}\left[\begin{array}{ll} \text{PHON} & f_{PSP}(\boxed{1}, \boxed{2}) \\ \text{SYN|LOC|SUBCAT} & \langle (\text{PP[BY]}_{\boxed{4}}), \ldots, []_{\boxed{3}} \rangle \\ \text{SEM|CONT} & \boxed{5} \end{array}\right]$$

Figure 2: Example structural accounts of the paradigmatic relationship between active and passive sentences

4 Representing choice: Systemic grammar

The types of functional account that have found most application within sentence generators are those based on *systemic-functional linguistics* (SFL). The theory of systemic grammar (more properly systemic-functional grammar: SFG) pursued within SFL combines a focus on inter-relationships between alternative forms of expression analogous to those illustrated in the previous section and a good match with the requirements of computational formalization. This organization of SFG around linguistic function is due to its origins within, on the one hand, anthropological (e.g., Malinowski 1923) and sociological (e.g., Firth 1957/1935) approaches to language and, on the other hand, European functional linguistics—particularly the Prague School and its forerunners (e.g., Bühler 1934). SFG itself grew out of work by Michael A.K. Halliday in England throughout the 1960s.

SFG builds on the traditional linguistic distinction between the *paradigmatic* and *syntagmatic* dimensions of linguistic organization (cf. de Saussure 1959/1915, Halliday 1963). Paradigmatic relationships are relationships between possible linguistic alternatives; syntagmatic relationships are relationships within individual linguistic structures. The alternations shown in (5) and Figure 2 are all examples of paradigmatic relationships. In a structural approach, the syntagmatic aspect of the alternation is the most important: hence the descriptions in Figure 2 all focus on the description of a structural difference. In contrast, in a functional approach such as SFG, it is the paradigmatic description that is central; here it is the *functional* aspects of the classifications exemplified in (5) that receive the most attention. This organization has been found to help significantly in the construction of NLG systems and in the formulation of the mapping between meaning and form. It also provides a metaphor for linguistic organization that contrasts usefully with structural accounts: rather than describing *constraints* on what kinds of structures are possible, the SFL view describes language in terms of what a speaker *can do with the structures of a language*. That is, language is interpreted as a *resource* for achieving various aims. This metaphor extends naturally to include the case where the 'speaker' is, in fact, an NLG system.

The most important feature of SFG is then its organization around the concept of 'choice'. Within SFG all grammatical variation is captured by abstract choices between minimal grammatical alternatives of the kind illustrated in (5). Moreover, all strata of the linguistic system—i.e., phonology, grammar, semantics—are represented within SFL as paradigmatically organized resources. The central representation adopted for this, which is found in all systemic approaches, is called the *system network*. It is with system networks that we therefore begin our detailed introduction.

The system network

A system network is a directed graph with labelled arcs whose nodes are choice points—the 'systems' from which systemic theory takes its name. An example of a small section of a system network for the grammar of English is shown in Figure 3. This subnetwork consists of eight systems, whose names are capitalized.

The outward directed labelled arcs of each system denote the output features, or *terms*, of that system. Each system has two or more terms, which at the stratum of grammar represent grammatical alternations: A grammatical system is then a point of 'minimal functional

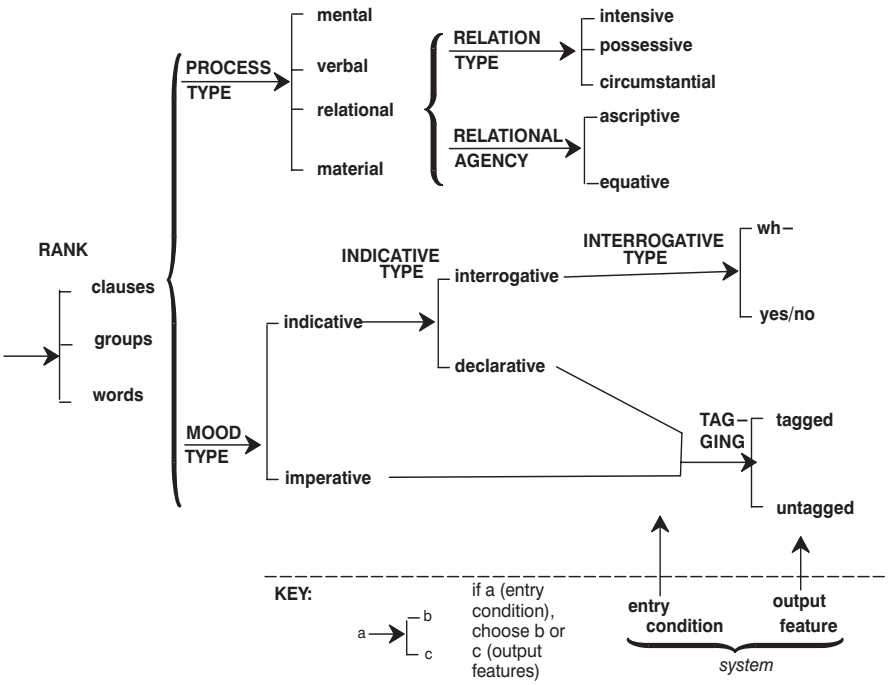

Figure 3: An example system network fragment

alternation'—i.e., a grammatical alternation that is attributable to a single functional difference; each of the alternatives shown in (5) would correspond in a systemic treatment to a grammatical system. In Figure 3, the terms of each system are shown in lower case; for example, the grammatical system PROCESS TYPE presents a choice of four possible grammatical features: 'material', 'mental', 'verbal', and 'relational'. This claims both that these four classes are functionally distinct and that they govern different patterns of syntactic behaviour; an extensive justification of this analysis of English is given by Matthiessen (1995*b*).

The inward directed arcs for each system denote an *entry condition* which determines the paradigmatic context in which the alternation represented by the system is relevant. Entry conditions can consist of disjunctions and conjunctions of output features from other systems; negation is not typically employed and the graph is generally required to be acyclic in computational approaches. A system network therefore partially orders its grammatical systems in terms of their interdependencies and consists of a collection of 'paradigms' of the form 'a linguistic unit of type A is either an A of functional subtype X, or an A of functional subtype Y, ..., or an A of functional subtype Z'. This kind of organization emphasizes statements of possible alternatives in related forms of expression and explicitly represents the choices available in the 'environment' of those choices that have already been made. Thus, 'clauses' in Figure 3 is the entry condition for the PROCESS TYPE system and this states that the system represents a functional alternation that is a necessary finer discrimination of the possible functions of clauses. Only when a linguistic unit bearing the grammatical feature 'clauses' is being described/generated, is it also necessary to make the selection of feature between 'material', 'mental', etc.

A disjunctive entry condition is illustrated in the TAGGING system. Here the alternation

between 'tagged' and 'untagged' is relevant when *either* 'declarative' *or* 'imperative' are selected, but not, for example, when 'interrogative' has been chosen. Possible English clauses compatible with combinations of these grammatical features as licensed by the network are shown in examples (6a)–(6d). An ungrammatical example corresponding to an inconsistent combination of features (i.e., a selection of features not possible given the interdependencies of the network) is shown in (6e).

(6)	a.	You are going.	{declarative, untagged}
	b.	You are going, aren't you?	{declarative, tagged}
	c.	Go away!	{imperative, untagged}
	d.	Go away, will you?	{imperative, tagged}
	e.	* Are you going, aren't you?	{interrogative}, {tagged}

Systems can also share entry conditions, indicating that they are all relevant in the paradigmatic context described by the shared entry condition. Such systems are called *simultaneous*. This occurs in Figure 3 with the PROCESS TYPE and MOOD TYPE systems: both have the entry condition 'clauses' and so both are relevant in the same paradigmatic context. A linguistic unit bearing the feature 'clauses' must make further selections from *both* MOOD TYPE and from PROCESS TYPE. This captures the fact that functional discrimination may proceed along several independent dimensions in parallel: it is not enough to decide whether a clause is asserting, ordering or questioning, it is also necessary to decide what kind of semantic event is being generated. Following any one path leaves the linguistic unit being described underconstrained with respect to the options presented by other simultaneous paths. A further example of simultaneity in the network is that of RELATION TYPE and RELATIONAL AGENCY—the distinctions offered by these systems both occur in the context of the grammatical feature 'relational'. The connectivity of the network therefore defines a partial ordering of systems from least *delicate* (most general) to most delicate (most specific).

One final kind of organization inherent in the system network is useful both linguistically and computationally for resource construction and maintenance. This involves the central systemic notion of *metafunction* (Halliday 1978). Metafunctions divide a grammar into three semantically motivated areas: one concerned with the expression of propositional content (called *ideational* in SFL), one concerned with the interaction between speaker and hearer (in terms of asking questions, giving orders, being polite or familiar, expressing opinions, etc.), and one concerned with the textual construction of a stretch of natural language (in terms of anaphors, focusing devices, ellipses, etc.). These all receive similar treatments within a system network. Thus, the choice between a pronoun and a full nominal group is seen as a functionally motivated alternation just as is the choice between a passive and active sentence, between a tagged or untagged sentence, or between a question or an assertion.

The three metafunctional areas of a grammar are interlinked in the normal way—i.e., simply by simultaneity of entry conditions. Systems of the network concerned with functions derived from the ideational metafunction will generally be simultaneous with systems concerned with the interpersonal and textual metafunctions. For example, the MOOD TYPE system and its more delicate dependent systems (e.g., INDICATIVE TYPE and TAGGING) form part of the description of the interpersonal component of the grammar of English—they all refer to aspects of discourse interaction. The PROCESS TYPE system and its dependent systems are, however, part of the ideational component—they describe the classification of experience in

terms of types of processes and of objects that may participate in those processes. This is why the MOOD TYPE and PROCESS TYPE systems are simultaneous; the functional discriminations from both metafunctional directions are pursued in parallel.

This simultaneity of metafunctions represents the theoretical claim that selections from *all three metafunctions* are necessary for the complete specification of any linguistic entity. As Halliday writes:

> "Whatever we are using language for, we need to make some reference to the categories of our experience; we need to take on some role in the interpersonal situation; and we need to embody these in the form of text."(Halliday 1974, p49)

Much information that would in other accounts be found, if at all, in a post- or pre-grammatical process of interpretation/generation are therefore in an SFG already represented as (functional) grammatical phenomena. This is very useful for sentence generation. The difference between sentences (7a) and (7b), for example, is represented in SFG as a distinctive selection of a *textual* grammatical feature—i.e., whether or not the time that is expressed also functions textually as a 'theme' or not. Representationally, such textual distinctions do not then differ from the ideational distinctions concerned with propositional content that are more traditionally found in grammar descriptions and so are readily available for control during the generation process.

(7)	a.	He went to London in June.	{unmarked-subject-theme}
	b.	In June he went to London.	{marked-theme, temporal-theme}

The simultaneity of portions of the grammar network according to metafunction also captures the fact that grammatical features motivated by distinct metafunctions can be varied largely independently of one another. Sentence (7b), which itself has the features 'declarative' along the interpersonal dimension defined by the INDICATIVE TYPE system and 'material' along the ideational dimension defined by the PROCESS TYPE system, can be additionally mutated along both these dimensions independently; this is shown in (8).

(8)	a.	In June did he go to London?	{marked-theme, temporal-theme, interrogative}
	b.	In June he was in London.	{marked-theme, temporal-theme, relational}
	c.	In June was he in London?	{marked-theme, temporal-theme, interrogative, relational}

The grammar is therefore responsible for taking independent descriptions from the different metafunctions and combining these into coherent paradigmatic and syntagmatic linguistic units.

The metafunctions also serve to define distinct functional regions within a grammar. Such regions possess strong intra-region dependencies and weak inter-region dependencies. This creates a modularity that is beneficial for grammar design, maintenance and development. This is also one analogous development to the use of 'modularity' in structural linguistics where different principles apply different sets of constraints . In SFG, the description of some phenomenon is also often distributed across several contributing metafunctions. We will see several further consequences of the metafunctional organization for sentence generation below.

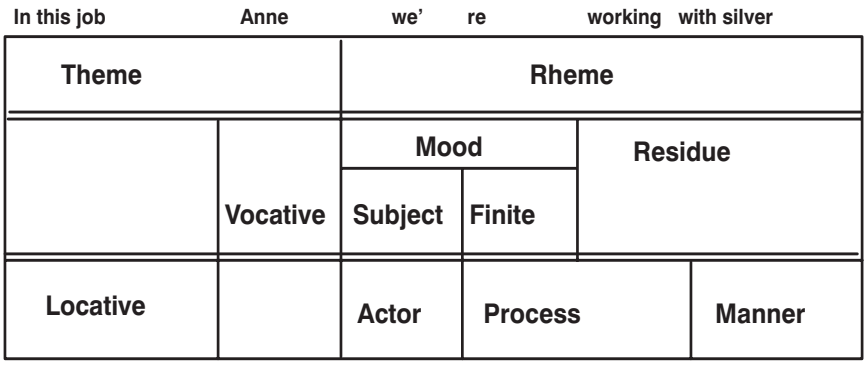

Figure 4: An example of functional grammatical structure
(simplified from an example in: Halliday 1985, p348)

Specifications of linguistic structure

In SFG grammatical features provide an abstract specification not of linguistic structure directly, but of the *functions* that a linguistic structure may realize. The 'bare' system network represents a purely paradigmatic grammar which determines the permissible combinations of grammatical features for each type of linguistic unit handled. Thus, the choices shown in Figure 3 represent the minimal functional alternations of types of process employed in a clause; they do not commit to a particular structural realization. In order to create the structural expression of these functional classifications, syntactic structure needs to be created. In SFG this is built up in terms of 'syntagmatic specifications' which are derived by means of *realization statements* associated with features in the system network. Syntagmatic realizations are thus always given in particular paradigmatic contexts. This separation of function and its structural expression, or *realization*, has been shown to be particularly useful for *multilingual* grammars; we return to this briefly below.

Syntagmatic organization is represented in terms of ordered sequences of constituents composed of grammatical *micro-functions*, such as Subject, Finite, Location, Goal, etc.; in this respect, SFG accounts resemble the functional approach to structure proposed in Lexical-Functional Grammar (LFG:). The number and range of micro-functions is, however, more finely developed within SFG since structure is specified completely and only in functional terms. These functions help perform two main tasks: the correct specification of ordering at each level of structure and an appropriate restriction of possible lower level realizations of constituents. An example of the 'functional' description of the clause: "In this job, Anne, we're working with silver" is shown in Figure 4. This gives a representative, but simplified, example of the kind of information that is specified in an SFG syntagmatic unit. Since much of the information necessary is encoded in terms of distinctly labelled microfunctions, such structures are usually 'flat' as shown here. This is generally the case for syntactic treatments that rely on a richer labelling system than pure constituency-based approaches.

Syntagmatic units in SFG are *layered* according to metafunction so that functions 'co-describe' constituents from the differing perspectives the metafunctions offer. The first layer shown in

Figure 4 is a textual layer, consisting of the functions Theme and Rheme. The Theme section of the clause serves to textually situate the interpretation of the Rheme: in Halliday's terms, it marks the "point of departure" of the clause. The second layer is an interpersonal layer: constituents that realize the functions of the speaker's "intrusion into the speech event" are located here. One of the constituents described interpersonally, the Vocative that explicitly realizes the addressee of the utterance, is also described textually by its co-labelling by Theme. The other constituent, labelled as the Mood element and consisting of Subject and Finite, realizes the communicative function of the speaker as making a positive statement about something that is currently the case and names the protagonist of the statement as 'we'. Finally, the third layer represents propositional (ideational) content—in particular the process (Process), participants (Actor), and circumstances (Locative and Manner) of the event being realized. Here also, one of the constituents described ideationally, the Location, is simultaneously serving a textual function as Theme. Another of the ideational functions, the Actor, co-describes the constituent labelled as Subject. In general, each column in such a syntagmatic description represents a set of grammatical functions which have been *conflated* (or unified) to define a single clause constituent. This provides a means of recombining the distinct contributions to structure made by the different metafunctional components of the system network. The *grammatical function* that each constituent is serving for the clause as a whole across all of the metafunctions is thereby made clear.

Syntagmatic level products, such as that of Figure 4, are built up by realization statements. Realization statements are associated with individual features from the system network. This defines a 'reflex in form' for the particular functional alternations that the features represent. For example, the reflex in form for English of the 'yes/no' alternation in the INTERROGATIVE TYPE system of Figure 3 is to order the constituent of the clause that functions as the 'Finite' element to the left of the constituent that functions as the 'Subject'—as in (8a) above where the Subject is 'he' and the Finite is 'did'. For Japanese, the reflex in form of an equivalent feature would be the clause-final addition of the particle *ka*. Example realization statements for English also appear in the network fragment repeated in Figure 5 below.

Just as the grammatical features of the system network specify minimal points of grammatical contrast, realization statements specify 'minimal' aspects of syntagmatic organization; this is similar to the current division betweeen Immediate Dominance ('ID') and Linear Precedence ('LP') rules found in structural approaches to syntax. Three principal types of realization operators are generally employed in SFG:

- The first type consists of those operators that introduce and build fragments of syntactic structure; these are: 'insert' (for specifying the presence of a functionally labelled constituent), 'conflate' (for specifying that two functionally labelled constituents co-describe a single constituent), and 'expand' (for constructing internal substructure). Examples of constraints consistent with the structure shown in Figure 4 are therefore: [insert Subject], [conflate Subject Actor] and [expand Mood Subject].

- The second type consists of those operators that constrain ordering; the most important of which is simply 'order' (for specifying the relative ordering of functions). A constraint again consistent with the example structure of Figure 4 is: [order Subject Finite].

- The third type consists of realization operators that further contrain the properties of fragments of substructure; the most important of these is 'preselect' (for specifying

grammatical features for immediate constituents). This constrains the realizations of subconstituents to be linguistic units of particular types; although not shown in the example, relevant constraints here include: [preselect Actor nominal-group] and [preselect Manner prepositional-phrase].

Because of the strict placing of such realization statements within the system network, they can only apply given appropriate paradigmatic feature selections. Thus, considering again the structure of Figure 4, only when it becomes known that a Manner must be expressed does it become appropriate to insert the necessary functionally labelled constituent; and then, subsequently, only when it is known that that Manner has a particular semantic type, does it become necessary to preselect it as having the feature 'prepositional-phrase', not before. Similarly, only when it is known that, for example, a Location is going to be used as the primary textual foregrounding device for the rest of the clause, does it become relevant to conflate Locative with Theme and so to place it at the beginning of the sentence. In short, *structural constraints are only applied in order to realize determinate choices of communicative function that need to be expressed grammatically.* A theoretical description of the possibilities for realization statements within an SFG is provided by Matthiessen (1985).

Chooser and Inquiry Semantics

The system network provides a representational resource for the functional potential of a language—i.e., for the types of functions that a language provides for a speaker—and the realization statements provide the means of constructing corresponding syntactic structures that fulfil those communicative functions. In order to make use of this in a computational context, it is necessary to further motivate the actual choices of grammatical features that can be made. That is, it is not enough for a sentence generator to specify that the language offers a minimal functional alternative between, for example, indicative (simple statements) and imperative clauses (simple orders), which in turn bring specific constraints to bear on structure. It is also necessary to specify the precise conditions under which indicative must be chosen over imperative, or vice versa.

This task has been handled in computational implementations of SFGs in one of four ways:

- by means of the *chooser and inquiry* semantics developed within the Penman text generation system (Mann 1985),
- by the *register-driven* semantics developed within the SLANG generation system (Patten 1988),
- by *probabilistic weightings* of feature selections (Fawcett & Tucker 1990),
- and by means of *constraint relations* as explored within several more experimental approaches to generation (cf. Bateman, Emele & Momma 1992, O'Donnell 1994).

We will focus in this section on the approach of the Penman system, since this is the one that has been used most widely in NLG systems. We mention register-driven semantics and probabalistic methods in our overview of systemic generators below, while constraint relations

can be found in those formalizations of SFL that include accounts of semantics in addition to grammar.

The chooser and inquiry framework for SFG arose out of the need to make a text generation system that was modular and re-usable across different contexts and across different computational systems, different knowledge representation languages, and different text planning components. To achieve such modularity, it is necessary that semantic control of a grammar component be provided without insisting that a user, or other computational system, be aware of the grammatical organization employed within the grammar. Since systemic grammars tend to be quite large, it would be a serious handicap for a generation system or user if they had to master the internal details of the grammar before being able to generate. This handicap is often still present in approaches based on other frameworks however.

The chooser and inquiry framework achieves this goal by associating a *chooser* with each grammatical system in the system network. A chooser is a small 'choice expert' that knows how to make an appropriate choice among the grammatical features of its associated system. It makes the choice by asking one or more questions, called *inquiries*. Inquiries take parameters that, typically, refer to aspects of the meaning, concepts, etc. that need to be expressed. It is the responsibility of these inquiries to obtain the information relevant for the grammatical decision. As far as the grammar and choosers are concerned, therefore, the inquiries represent oracles which can be relied on to motivate grammatical alternations appropriately for the current communicative goals being pursued. A simplified view of part of our example system network fragment with choosers (and realization statements) attached is shown in Figure 5.

This is a simpler task than directly requiring a selection of grammatical features since the choosers and inquiries decompose a single selection among minimal grammatical distinctions into a number of selections among minimal *semantic* distinctions. While the grammatical alternations may not be directly relevant to a component external to the SFG, the semantic distinctions are relevant: this semantic level supplies a situation-independent semantic classification in terms of which a computational system can usefully organize its information for expression in natural language.

The chooser and inquiry approach has proved itself useful not only for modularizing the generation process, but also for providing a bridge between more informal, functional-descriptive approaches to linguistic phenomena of the kind often found in text linguistics and more formal approaches that are the goal of computational instantiation. This is important both for opening up a dialogue between computational and non-computational (functional) linguistics and for allowing sentence generation to proceed even in areas where our understanding of the mechanisms involved are still too incomplete to admit of detailed formalizations. We show these processes at work by presenting two more detailed examples of choosers and inquiries. The first concerns the selection of either singular or plural nominal groups in English, showing how this is more complex than might be thought prior to examining how these forms actually function in texts. The second returns to our active-passive alternation example, again taking English as illustration language.

The grammar of English (and of course several other languages) shows a functional distinction that characterizes of nominal groups in terms of 'plurality': i.e., the basic distinction between 'lion' and 'lions'. A semantic motivation in terms of the cardinality of the set of objects referred to is not sufficient for full control of the grammatical options available during sentence generation. If we examine texts where plural and singular distinctions occur, we see a range

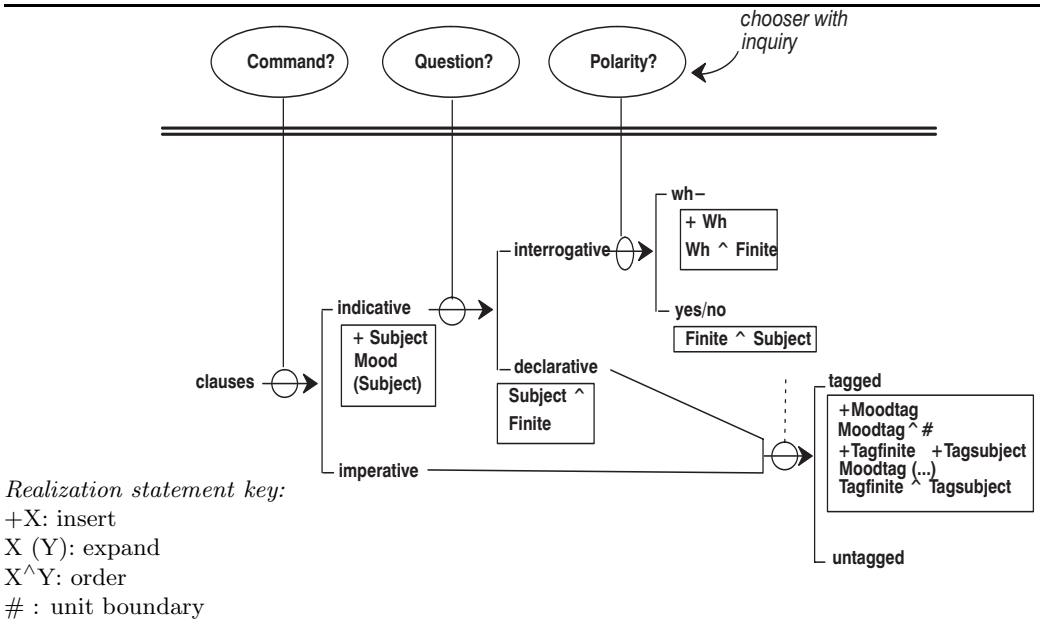

Figure 5: Choosers and system network

of types both of plural and of singular. For example, the sentences (9a) and (9b) do not differ so much in meaning despite that fact that one selects a plural nominal group where the other selects a singular one.

(9) a. The lion is almost extinct.
b. Lions are almost extinct.

Within plurality alone there are a number of possible meanings; each of which licenses a different set of inferences. These are illustrated in the sentences of (10). In the first sentence, there are actual individuals involved—this comes perhaps closest to the 'pure' use of plural as an indication of set cardinality. In the second, a general statement about the entire set of lions is made: this statement is not being made of any particular instances of the species, it is the set as a whole that is under discussion. And in the third, a statement is made that applies to any particular instance taken from the set as a whole. A sentence generator has to know that it can use the plural form to express the meanings indicated, and that for cases (10b) and (10c) singular forms, although differing in emphasis, are also possible.

(10)	a.	Three lions chased me	[existing individuals]
	b.	Lions are almost extinct	[species consisting of individuals]
	c.	Lions have sharp teeth	[generic statement about individuals]

The distinctions involved can be captured by associating an appropriate chooser with the PLURALITY system of the English grammar —i.e., the grammatical system in the system network that chooses between 'plural' and 'nonplural' nominal groups. Such a chooser is shown in Figure 6; this chooser is drawn from the grammar of English developed within the Penman project, the 'Nigel' grammar, which is based closely on Halliday (1985). From the

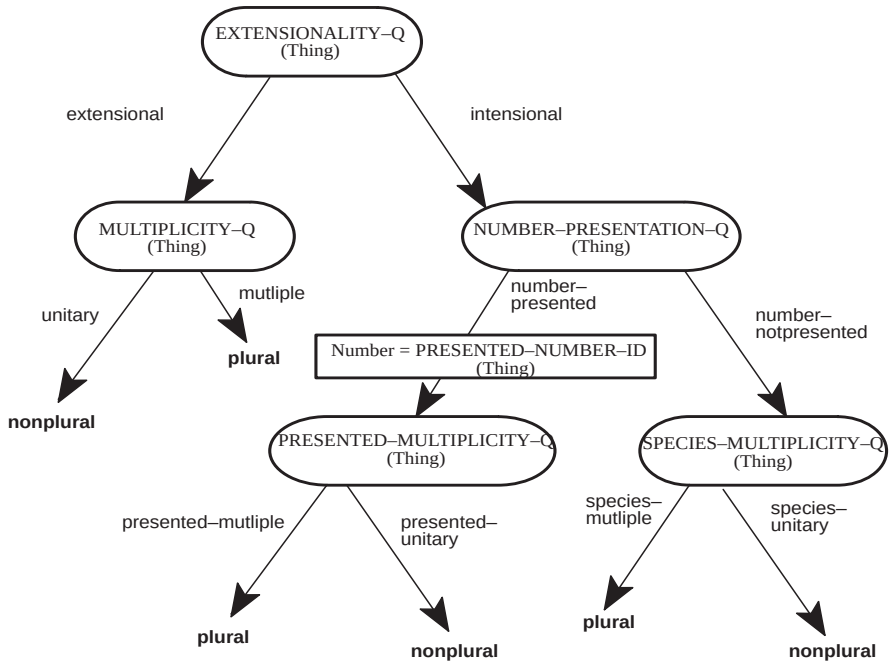

Figure 6: An example chooser for plurality in English

figure we can see that a chooser is straightforwardly represented as a 'decision tree' with a particular kind of inquiry, called a *branching* inquiry, forming the decision points. Branching inquiries are generally written suffixed by "-Q" to indicate their 'querying' status. The role of a branching inquiry is to make a decision among its prespecified set of possible responses on the basis of the current semantics being expressed, thereby navigating through the options provided by its choosers. In Figure 6, for example, the inquiry EXTENSIONALITY-Q has two possible responses: EXTENSIONAL and INTENSIONAL. Which of these is selected in a particular context determines the particular lines of reasoning followed up subsequently in the chooser. The chooser thus decomposes the grammatical choice between 'plural' and 'nonplural' into a number of basic semantic discriminations, each represented as an inquiry.

The meaning of inquiries is defined in two ways:

- first, an *informal natural language description* of the semantic discrimination at issue for the inquiry can be given,
- second, an actual computational process may be implemented which interrogates a knowledge base, text plan, etc. in order to establish the response appropriate for the particular communicative goal being achieved.

The first form of definition is normally based on a linguistic analysis and distills the results of that analysis into a succint statement. Ideally, such statements could explicitly refer to the empirical work on which they are based. The description then provides not only a convenient repository for the results of, possibly informal, linguistic analysis but also an initial specification of what needs to be implemented in any particular computational system that concerns itself with the distinction being described.

The inquiries in the current example ask, for example, first whether the concept being expressed is extensional or intensional: if it is extensional and the set cardinality is greater than one, then 'plural' must be selected; if it is intensional, then it can still be the case that that intensional object denotes some set and the question becomes whether the plurality of this set is to be focused on or not. Note that this is usually a 'textual decision'—i.e., it can only be made on the basis of the perspective being taken in a text; it is not a propositional issue concerning the truth-conditions of the asserted statements. Even if the plurality of the intensional object is not being focused on, it may still be the case that individuals drawn from that intensional object are being focused on as exemplars of their class—they are not then unique and so can similarly receive a 'plural' selection. The following sets out the three paths through the chooser that correspond to the three distinct kinds of plural illustrated in (10) respectively.

either				
		EXTENSIONALITY-Q	:	extensional
	and	MULTIPLICITY-Q	:	multiple
or				
		EXTENSIONALITY-Q	:	intensional
	and	NUMBER-PRESENTATION-Q	:	numberpresented
	and	PRESENTED-MULTIPLICITY-Q	:	presentedmultiple
or				
		EXTENSIONALITY-Q	:	intensional
	and	NUMBER-PRESENTATION-Q	:	numbernotpresented
	and	SPECIES-MULTIPLICITY-Q	:	speciesmultiple

The second form of inquiry definition, the implementation, is usually a computational approximation to the theoretical (but empirically motivated) specification given in the first form. For areas of meaning where formal theories are not well developed, the implementation provided represents a current 'best guess' for a working system. This can be refined as formalization improves without requiring changes elsewhere in the generation component.

Inquiry implementations must access knowledge base entities and other information, such as text plans and communicative intentions, in order to make their decisions. They achieve this via their specified *parameters*. In the example of Figure 6 the parameter for all of the inquiries is the same, i.e., `Thing`. This parameter is a grammatical micro-function in the sense defined in the previous section. Each grammatical function may be associated with a concept (represented in whatever knowledge representation language is adopted by the system interacting with the sentence generator) which may then be used by the inquiries as a pointer to information external to the grammar. The fact that choosers are defined entirely in terms of the grammar-internal grammatical functions ensures that they remain application-independent and preserves the modularity of the grammar–semantics component.

Associations between grammatical functions and external concepts are established by means of a second type of inquiry. These are called *identifying* inquiries and are typically suffixed by "-ID". In contrast to branching inquiries, identifying inquiries return pointers to unique concepts. These pointers are then associated with the grammatical functions named in the chooser. An example of this in Figure 6 is the identification of the grammatical micro-function `Number`—corresponding, for example, to the "three" in example (10a)—by means of the inquiry PRESENTED-NUMBER-ID being put with respect to the grammatical function `Thing`. An appropriate implementation of this inquiry would examine the knowledge base

entity identified by the association of `Thing` and return a pointer to the information concerning the 'numberhood' of that concept.

Our second chooser example concerns a generation treatment of the grammatical alternation illustrated in a clause paradigm such as (11). The system network offers a grammatical system covering this three-way alternation with output features: 'benefactive-operative' (11a), 'benereceptive' (11b), and 'medioreceptive' (11c); the terminology is defined and motivated in, for example, Halliday (1985).

(11) a. John gave the book to Mary
b. Mary was given the book (by John)
c. The book was given to Mary (by John)

As always for sentence generation, it is necessary to find a discrimination pattern that allows the generator to decide on the basis of its input which of these alternatives is most appropriate for a particular text. A first approximation to the relevant motivations here can be found in questions of 'salience': It might be expected that the element that is most salient might be made the grammatical Subject. It remains to consider in more detail what salience might mean in this context. The favored methodology here is again, as with the plural examples, to turn to empirical text analysis. The particular account of the active-passive distinctions illustrated in (11) that is adopted in the grammar of the Penman sentence generator is based on functional work by Thompson (1987); further proposals for functional work on communicative salience, such as that in the Prague School tradition (cf. Sgall, Hajičová & Panevová 1986), would provide similarly appropriate starting points for a sentence generator.

The main empirical result of Thompson's (1987) study is that the selection of grammatical Subject in English is often based on a notion of 'conceptual distance' between an object in question and a discourse notion of *paragraph theme*. In general, the participant that is 'nearest' the paragraph theme will be the strongest candidate for becoming Subject. We will not concern ourselves here with the details of this empirical study, but will rather use it as a further example of how empirical results can be used for setting up corresponding choosers. A chooser for the grammatical system involved in (11) can be set up straightforwardly as follows.

First, the position of the grammatical system in the grammar network as a whole means that the three participants denoted by the grammatical micro-functions `Agent` ('John' in our examples), `Medium` ('the book') and `Beneficiary` ('Mary') are to be expressed in the sentence. These will therefore have already been 'identified' by appropriate -ID inquiries. Second, a pointer to the discourse 'paragraph theme' must be provided. This may also be achieved straightforwardly by identifying a function `Paratheme` by application of a further -ID inquiry PARAGRAPH-THEME-ID. This has the natural language description:

"What is the paragraph theme of the paragraph containing EVENT?"

It is important that no commitments are made to just how a particular theory of discourse will implement this. Regardless of implementation, the account provided in the grammar will continue to function for sentence generation. Third, the relevant distances between the paragraph theme and the sentence participants must be provided. This is modelled in

terms of 'conceptual paths' (e.g., following links in a semantic network); these can also be ascertained by an identifying inquiry. An inquiry called READER-KNOWLEDGE-PATH-ID, with the following natural language description, is sufficient:

> "What symbol represents the most salient chain of relationships in the reader's attention and knowledge between X and Y?"

The paths needed are then accessed by applying this inquiry for each participant:

AGENT-PATH = READER-KNOWLEDGE-PATH-ID (AGENT PARATHEME)
MEDIUM-PATH = READER-KNOWLEDGE-PATH-ID (MEDIUM PARATHEME)
BENEFICIARY-PATH = READER-KNOWLEDGE-PATH-ID (BENEFICIARY PARATHEME)

The empirical analysis suggests that the grammatical decision rests principally on whether the beneficiary is more central than the agent or not. When this condition is met then the grammatical selection must be 'benereceptive' (11b). The path configurations for this state of affairs are summarized in (12).

(12)
paratheme — *medium*
& *paratheme* — *beneficiary* — *agent*
paratheme — *beneficiary* — *agent* — *medium*
paratheme — *beneficiary* — *medium* — *agent*

Only if (i) the medium lies explicitly between the beneficiary and the paragraph theme and (ii) the agent is not nearer to the paragraph theme, does the medium become a candidate for subject ('medioreceptive': (11c)). This is then required only when either of the path configurations summarized in (13) hold.

(13)
paratheme — *medium* — *beneficiary* — *agent*
paratheme — *medium* — *agent* — *beneficiary*

If a further branching inquiry, PATH-INCLUSION-Q, is defined for comparing such conceptual paths as follows:

> "Does the chain of relationships ARG2 contain the chain of relationships ARG1 as a proper subpart?"

then a chooser decision tree covering the possibilities can be set out straightforwardly as shown in Figure 7. This chooser simply classifies the actual textual state of affairs against the possible kinds of paths. The chooser therefore provides the required choice expert for the grammatical paradigm illustrated in (11). The path configurations compatible with the selection of 'benereceptive' correspond to the following sets of semantic conditions, again expressed in terms of inquiries and their responses (writing here '$\subset$' for the inquiry PATH-INCLUSION-Q):

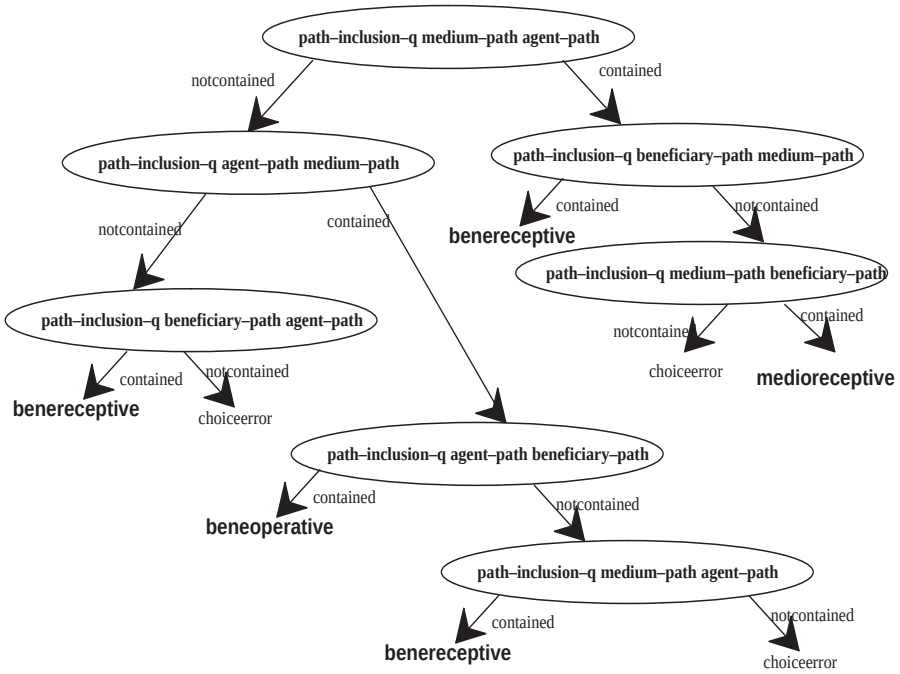

Figure 7: An example chooser for active/passive selection

either				
		BENEFICIARY-PATH ⊂ AGENT-PATH	:	contained
	and	AGENT-PATH ⊂ MEDIUM-PATH	:	notcontained
	and	MEDIUM-PATH ⊂ AGENT-PATH	:	notcontained
or				
		BENEFICIARY-PATH ⊂ AGENT-PATH	:	contained
	and	AGENT-PATH ⊂ BENEFICIARY-PATH	:	notcontained
	and	AGENT-PATH ⊂ MEDIUM-PATH	:	contained
	and	MEDIUM-PATH ⊂ AGENT-PATH	:	notcontained
or				
		BENEFICIARY-PATH ⊂ MEDIUM-PATH	:	contained
	and	MEDIUM-PATH ⊂ AGENT-PATH	:	contained

The three possible paths through the chooser that lead to a selection of the grammatical feature 'benereceptive' indeed correspond to the distinct degrees of relative salience among the three participants involved. But they do so in a way that makes the particular discourse nature of the notion of salience adopted more precise in the ways motivated by the empirical analysis.

It is probably easy to imagine how these informal specifications could be made more formal with respect to any particular model of discourse structure that might be adopted by the NLG system as a whole. Any such formalization would not require any alterations in the grammar component, thus allowing further development of the grammar and consideration of possible interactions of this phenomenon with others independently of the discourse account. It is also the case that such informal chooser specifications can provide a good basis for preparing language teaching materials: a chooser should make the conditions of use of a grammatical alternation more generally intelligible. Several further illustrative examples of choosers and

inquiries for capturing functionally motivated distinctions are given in Matthiessen & Bateman (1991), including an account of tense in English and of the *wa*/*ga* distinction and some issues of expressions of politeness in Japanese.

Generation algorithms

The system network as described in the preceding sections—i.e., including both realization statements and choosers—provides an abstract specification of 'linguistic potential'. As is usually the case in computational and formal linguistic work, the grammar describes only what structures are abstractly possible. It does not automatically include a statement of how such structures are produced. We need in addition to specify a *generation algorithm* which determines how a systemic grammar is interpreted to build linguistic units. The generation algorithm of the Penman NLG system is one of the simplest that can be employed and so we will describe this here. Some more complex possibilities are touched on in the overview of systemic generation systems given below.

The generation algorithm consists of three principal components:

- a *traversal* algorithm for exploring the system network,
- a *structure building* algorithm for executing realization statements,
- a *chooser interpreter* for following the decision trees defined by choosers.

Each grammatical unit that is generated is created by a single complete traversal of the system network. Traversal must follow the decisions made by the choosers and conform to the network's connectivity. The realization statements associated with all the features selected on this traversal are collected. Executing these statements defines the grammatical unit generated. The first step in generation is therefore to provide a starting point for traversal of the system network, to note which grammatical unit is being constructed, and to provide access to the semantic entity that that grammatical unit is to express. For example, if generating a sentence for the logical expression given at the outset of this article, therefore, the generation algorithm would select the feature representing the root of the system network (i.e., the 'left-most feature'), note the grammatical unit being generated to be the full 'sentence' or 'utterance', and provide a pointer to the entire logical expression as an association of a special grammatical function (`Onus`) used specifically for this purpose. The inquiries of the first choosers have this grammatical function as parameter and this enables them to access further semantic information are required.

Traversal proceeds by examining all the systems of the network to see if any of these have their entry conditions satisfied. Only those systems that have the root feature as their entry condition will be satisfied when traversal begins. Such systems are added to a 'waiting list'. Systems are selected from the waiting list randomly and their respective choosers are consulted in order to select further features. For each further feature selected, the systems of the network are again examined to see if any new systems have had their entry conditions satisfied. This procedure repeats until the waiting list is empty. This corresponds to a maximal traversal of the system network. This traversal algorithm is shown as a flowchart in Figure 8.

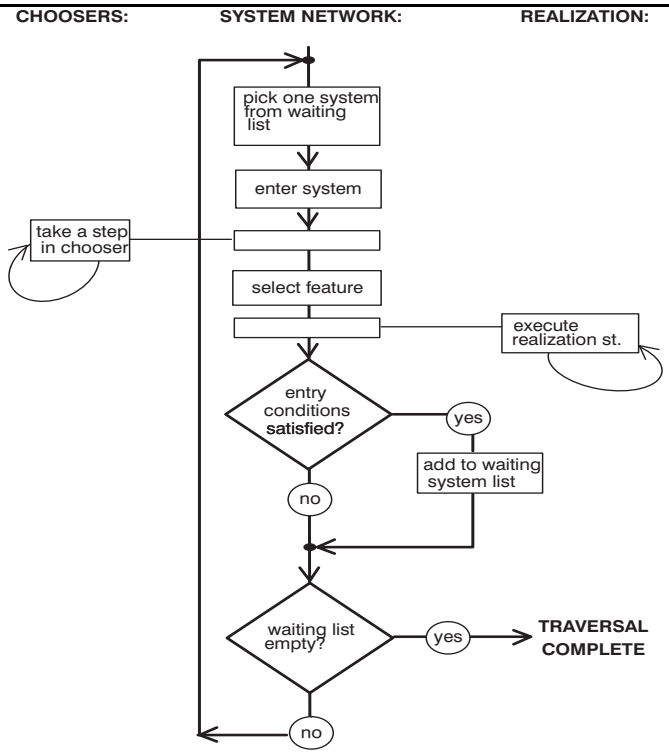

Figure 8: A flowchart for generation
(adapted from: Matthiessen & Bateman 1991, p106)

Combining the realization statements collected on a traversal of the network defines the syntactic structure of the grammatical unit generated *at a single depth of structure*. For example, the sentence generator may know that a clause has been generated and that that clause must have a Subject as immediate constituent, but will not yet know what type of Subject that might be. Such embedded constituents then need to receive their own grammatical descriptions. This is performed in further cycles through the grammar: For each constituent the system network is re-entered at the least level of delicacy with `Onus` set to the component of the semantics corresponding to that constituent. Choices appropriate for the description of the constituent are then made in the same way as they are for the clause. It is also usual that a dominating grammatical unit will set additional constraints on the network traversals for its immediate constituents; for example, that some constituent should receive a particular case marking, etc.

Figure 9 illustrates this process graphically for the generation of a sentence corresponding to our initial example logical form. The suffixes for `Onus` indicate successive cycles through the grammar. Each cycle is therefore concerned with some extract from the logical form as a whole. Since the choosers address different extracts in subsequent cycles, they make different decisions. These decisions determine the various paths through the grammar network and, hence, the component syntactic structures that are built up. For example, during generation of the main clause, a constituent is inserted (`Attribute`) that is associated with the component of the semantics corresponding to:

$$\text{on}'(x, y) \wedge \text{table}'(y)$$

Choosers interrogating the semantics associated with the `Attribute` constituent guide the traversal algorithm to select features that include a preselection realization [preselect Attribute prepositional-phrase]. Some subsequent cycle of grammar network traversal is therefore made responsible for constructing the internal structure of this consitituent. Before commencing this cycle, the semantic extract associated with `Attribute` is made the associated value of the function `Onus` (`Onus-3` in the figure). All choosers that are activated during this grammar traversal therefore ask inquiries relative to just this portion of the semantic input. The preselection constraint guides traversal directly to the area of the grammar for prepositional phrases. This is then sufficient for the generation algorithm to build an appropriate internal structure for the prepositional phrase—consisting in this case of two micro-functions: `MinorProcess` and `Minirange`. The other constituents are generated similarly.

In summary, then, generation proceeds in cycles of *actualisation*: throughout actualisation the grammar is entered at the least level of delicacy and choices are made until the maximally delicate distinctions offered by the network have been drawn. Each complete cycle through the grammar produces a complete paradigmatic description of a linguistic unit of a given 'size', or *rank*. The rank scale is typically given as: clause, group/phrase, word and morpheme; this can vary across languages however. The set of grammatical features accumulated on a cycle through the grammar network is called the *selection expression*. Cycling through the grammar is repeated until paradigmatic descriptions of the linguistic unit at all structural levels (ranks) have been produced. Units that have no immediate constituents form the leaves of the generated result. Such units are typically formed by direct lexicalization. Here a lexical unit is selected on the basis of some portion of the input semantics.

The system network underspecifies the generation algorithm in several respects. For example,

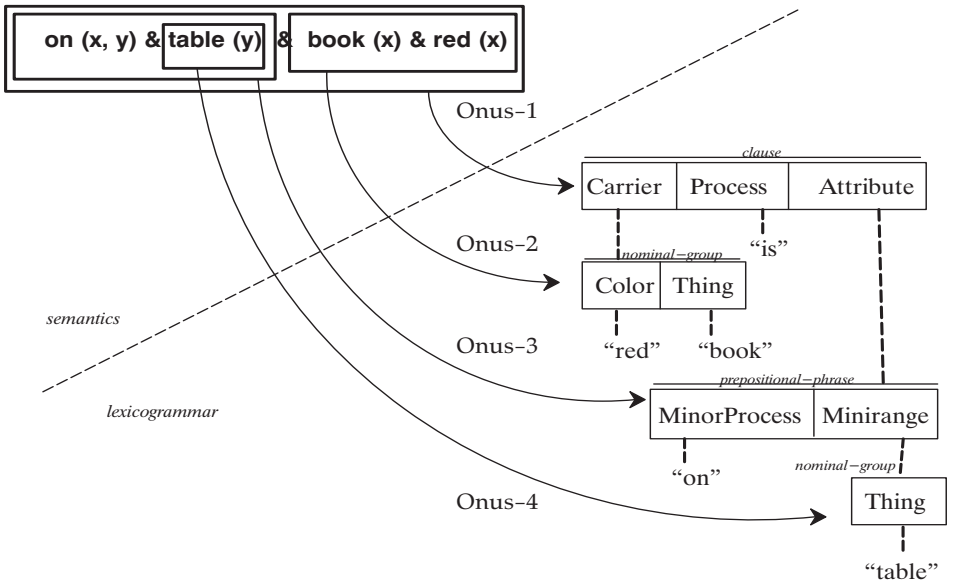

Figure 9: Example of generation cycles linking semantics with syntactic structure

it is not specified in which order 'simultaneous' systems are to be entered, nor is it specified when realization statements are to be evaluated. Some of these underspecifications provide appropriate places for parallel implementations. Others require more theoretical consideration of the consequences of any decisions made. The generation algorithm described here supports efficient implementations: there is no backtracking or non-determinism to incur performance overheads. The traversal of the network directs the generation process precisely as required by the input specifications. The price for this efficiency is the reduced range of interactions possible across different structural ranks and the requirement that choosers always produce a determinate response when triggered.

5 The input to a systemic sentence generator

Various types of input specification are possible for a sentence generation component based on systemic grammars as described here. The most common, and most straightforward for general use, is to provide an input specification that provides sufficient information for inquiries to make appropriate decisions. When this is achieved, a user or external computational component need know nothing of the internal organization of the grammatical component. The chooser and inquiry interface is fully responsible for mediating between such an input and the final generated string.

An adequate input specification could consist solely of a list of semantic labels and the desired inquiry responses. For example, in order to generate the nominal phrase 'lions' as used in (10a), we would need (i) to provide a label representing a pointer to a semantic element for lions as the association for the gramamtical function `Thing` (cf. Figure 6), and (ii) to provide responses to the inquiries described in our first chooser example. Corresponding expressions would be:

`Thing` = thing-id (`Onus`)
extensionality-q (`Thing`) = extensional
multiplicity-q (`Thing`) = multiple

For a complete specification, however, we would also need to provide equivalent inquiry responses for all of the other inquiries that are asked along the path making up that traversal of the systemic network. This would involve far too many inquiries to be of practical use.

The input specification required is simplified greatly when inquiries are implemented. Here inquiries can ascertain their own responses and so this information does not need to be given explicitly. One challenge with this method, however, is again to achieve modularity such that inquiry implementations are themselves maximally re-usable across different applications and knowledge representations. Rewriting inquiry implementations whenever the application changes presents a considerable, and hence unacceptable, overhead. A solution to this problem is to consider the abstract kinds of information that inquiries (and hence inquiry implementations) presuppose. The distinctions represented within inquiries actually induce organizations of knowledge that a computational system needs to support in order to generate language. Regardless of how a computational system represents this information, it must be present in some form if that system is to be able to fully use the flexibility offered by the generation component. This establishes strong constraints on the form of an input specification. We have seen that a systemic grammar includes information drawn from the three metafunctions: that is, the selection expression accumulated on any cycle through the grammar network contains features drawn from all three metafunctional components. Many of the inquiries that are asked during traversal will therefore concern non-propositional aspects of the communicative intention. Since an input specification has to provide this information, extensions beyond a bare logical form are required.

The currently most completely investigated knowledge organisation induced by the inquiries considers all inquiries from the ideational metafunction. This information, corresponding to propositional content and logical form, is still the best understood area and so it is natural that this area be the first to receive a greater degree of formalization. The majority of ideational inquiries ask questions that request a classification of an input semantic category in terms of abstract semantic categories. These categories can be organized into a semantic hierarchy, called the *Upper Model*; this is motivated in detail in Matthiessen (1987) and the latest computational version of such a hierarchy is documented in Bateman, Henschel & Rinaldi (1995). The existence of an Upper Model renders the implementation of many inquiries straightforward: they simply ask whether their parameter is a subtype or not of a category drawn from the Upper Model. In order to reach the part of the grammar that is responsible for generation of a nominal phrase such as "lions", therefore, a semantic concept given as input must simply be placed beneath the appropriate Upper Model concept (e.g., `Object`). This is called 'subordinating a *domain concept* to the Upper Model'. Inquiry implementations then use this information to provide choosers with sufficient information to, for example, reach the nominal group part of the grammar network rather than that part responsible for clauses, or adverbial phrases, etc.

The first task in preparing input for a Penman-style systemic grammar is then to take a semantic representation of the *domain*—the subject area about which texts are to be generated—and to subordinate the concepts of this domain to those of the Upper Model. Since most domains are already represented in terms of a hierarchically organized knowledge system, subordinat-

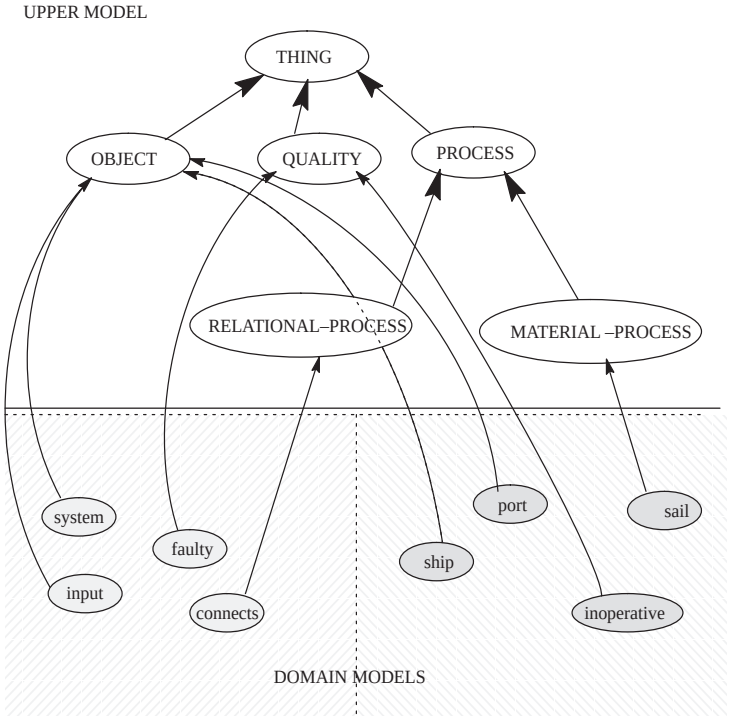

Figure 10: Example re-use of the Upper Model across two domains

ing such concepts to the Upper Model is simplified. The Upper Model is motivated by the grammar rather than the application and so remains constant across different domains. This means that as long as an application system provides some representation of the Upper Model inheritance hierarchy, the inquiry implementations can remain constant across domains. This re-usability across domains is depicted graphically in Figure 10. This strategy absorbs much of the complexity of input specifications and must be performed only once for each domain.

A further simplification of each input specification given is achieved by defining a *default response* for each inquiry. This response is used when no further information is present in the input. This avoids the need for 'negative' information (for example, that there is no particular number information to be presented—i.e., "lions" rather than "three lions"—or that there is no temporal information, etc.) in the input expression. This allows the expression given in (14) to serve as an input specification sufficient to produce the nominal phrase "lions" even when given to the complete grammar of English (which requires responses to be given for approximately 60 inquiries for this phrase alone).

(14)
```
(o / lion :extensionality-q extensional
          :multiplicity-q   multiple)
```

This generator input is expressed in the Penman *Sentence Plan Language* (SPL: Kasper 1989*a*). The specification consists of a set of semantic variables (e.g., `o`), each of which receives a semantic type (`lion`) which must have been subordinated to an Upper Model concept. This covers the logical form aspect of the input. In addition, each variable may have a set of

inquiry and inquiry responses that are to be used when generating the natural language form corresponding to that variable. This permits very fine control of the grammatical possibilities offered by a grammar since information additional to the bare logical form is cleanly integrated in terms of the given inquiry responses. In general, any information that is not fully represented in the knowlege representation system adopted can be given explicity in terms of inquiry responses. The full coverage of the systemic grammar is therefore always available.

An example of a full SPL for sentence (8c) ("In June was he in London?") is shown in Figure 11. This includes information from the interpersonal metafunction concerning the speech function (positive assertion) and from the textual metafunction concerning the thematic status of the temporal location (variable: `t0` for "in June") and the anaphoric status of the person referred to (variable: `x0`). Ideational information is given in teletype script, interpersonal in italics, and textual in bold. The semantic specification includes inquiry responses for temporal questions necessary for calculating the correct tense reference: this is also a mixture of interpersonal (the speaking time), textual (the 'reference' time), and ideational (the event time) information. It is, however, possible to simplify this input specification using inquiry defaults and macros; thus the SPL specification given in (15) produces the same string but hides most of the semantic content. These kinds of specification are typical of inputs for sentence generators in general.

(15)
```
(e0 / spatial-locating
    :tense past
    :theme t0
    :domain (x0 / person :pronoun he)
    :range (x1 / three-d-location :name London)
    :temporal-locating (x2 / three-d-time :name June))
```

The inquiries associated with choosers of systems from the different metafunctions in the grammar require different sources for their responses. Only the ideational inquiries typically examine the knowledge base or domain model of a computational system. Current research and development is therefore considering formalizations for the non-ideational semantic components. Interpersonal inquiries need to examine a user model and textual inquiries examine the text plan and text history; the latter is highly consistent with recent research directions involving Discourse Representation Theory (DRT: cf. Kamp & Reyle 1993). Matthiessen (1987) describes the relation between the metafunctions and different kinds of 'support knowledge' that are required in some detail. This metafunctionally differentiated view of 'semantics' is shown graphically in Figure 12. All such attempts to add higher level organizations of information ease the task of text planning. It is the responsibility of the text planner of an NLG system to produce input specifications for the sentence generator. The more this component can operate at a purely semantic level the greater modularity of components can be achieved.

6 Systemic generation systems: an overview

Broadly systemic-functional approaches have been adopted in a wide range of generation systems—probably more than has any other linguistic framework. This position is due to the close match of the linguistic description task seen from the perspective of SFL and the needs of linguistic resources (such as grammars, semantics, etc.) that are usable for natural language

```
(e0 / spatial-locating
      :reference-time-id et
      :speech-act-id
       (sa-0 / assertion
             :polarity positive
             :speaking-time-id
                (st / time
                            :time-in-relation-to-speaking-time-id et
                   :precede-q (st et) notprecedes)
             :event-time
               (et / time
                   :precede-q (et st) precedes))
    :theme t0
   :domain (x0 / person
                :empty-number-q empty
                :empty-gender-multiplicity-q empty
                :multiplicity-q unitary
                :singularity-q singular
               :gender-q male
                :identifiability-q identifiable )
   :range (x1 / three-d-location
                :name London)
   :temporal-locating (t0 / three-d-time
                           :name June )))
```

Figure 11: Example of full SPL specification for the sentence: "In June was he in London?"

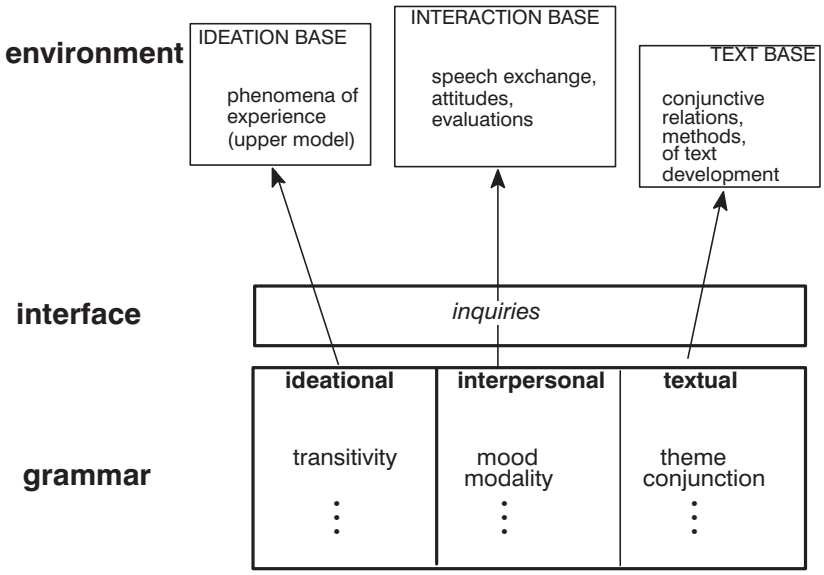

Figure 12: Semantic environment diversified according to metafunction

generation. In this section, we provide a brief overview of generation systems that have been based on systemic grammar, as well as some interesting work where systemic generation has been combined with other theoretical approaches.

Probably the first attempt to use a systemic grammar for generation was that of Henrici (1965), who described a computer program that could generate simple clauses on the basis of an early systemic grammar. This involved one of the first attempts to formalize systemic grammars for computational use—a topic we return to below. A similar, if more extensive, attempt was that of McCord (1977). However, the first system to push the use of systemic generation for the purposes of full text generation was Davey's (1978) PROTEUS system. This was a genuine text generation system in that it planned and produced texts rather than focusing only on the generation of isolated sentences.

The PROTEUS system produced short transcripts of games of noughts and crosses (tac-tac-toe). The input to the program was a list of moves in a possible game as they would have occured. Moves were grouped into sentences in order to reflect 'tactical coherence' relying on the semantics of the game itself to provide textual coherence. The texts produced were quite sophisticated; the quality of the output was in large part due to the sophistication of its grammatical component: a systemic grammar developed by Davey. Davey's system is historically significant in that it made clear how much the output of a generation system could be improved when it had a reasonably sophisticated linguistic model of possible forms of expression—i.e., a grammar. An example of a transcript produced by PROTEUS is given in (16).

(16) The game started with my taking a corner, and you took an adjacent one. I threatened you by taking the middle of the edge opposite that and adjacent to the one which I had just taken but you blocked it and threatened me. I blocked your diagonal and forked you. If you had blocked mine, you would have forked me, but you took the middle of the edge opposite the corner which I took first and the one which you had just taken and so I won by completing my diagonal.

The value of an extensive grammatical component for high quality generation led to a defining characteristic of the original design of the Penman text generation system (Mann 1985). Penman was built to provide a general, application and domain independent text generation system. It also provided an environment within which all areas of research and development relevant to text generation could be addressed. Partly in order to overcome the limitations of an earlier experimental generation prototype, Mann, the originator of the Penman system, decided that the system should be based around an extensive grammatical coverage of English and that the grammatical component should be systemic-functionally based. The Penman system, and its English grammar Nigel (Matthiessen 1981), became one of the largest and most established generation systems and forms the basic reference model for systemic generation. Most of our description of systemic grammar above draws directly on the Penman model.

Although very extensive, Penman by no means exhausts the possible approaches to generation within a systemic framework. A different focus on a systemic model of generation was taken by Patten's (1988) SLANG system. This was the first generation system that attempted to employ systemic *register theory* to explicitly model sub-language variation that is dependent on context. Although more limited in scope than the generation of various registers in Hovy's PAULINE (cf. (1) above), SLANG is interesting in that it applies a single linguistic theory to phenomena ranging from syntax to context. The main computational feature that differ-

entiates SLANG from Penman is that generation is no longer 'delicacy-increasing'—i.e., the generation process does not follow the 'left-to-right' traversal of a system network described above. Instead, a description of the communicative context first preselects arbitrary grammatical features drawn from around the system network representing the grammar. This is a concrete way of stating the basic premise of register theory: i.e., that the language that is used in a particular context will inherit particular linguistic features. The generation algorithm of SLANG then pursues complete grammatical descriptions compatible with the set of preselected grammatical features. It is, however, clear that both the semantic-driven (e.g., Penman's chooser and inquiry semantics) and the context-driven approach of Patten have a role to play in a complete view of the generation process. More recent work has sought to combine these aspects. For example, both Cross's (1992) systemic-functionally based Horace system and Bateman, Maier, Teich & Wanner's (1991) KOMET system include richer descriptions of context, text structure, and lexical choice. Vander Linden, Cumming & Martin's (1992) IMAGENE system also includes an attempt to use system networks for linguistic information other than grammar—in this case, for text planning.

Another approach differing from the premises of the Penman system is that taken in the Communal system of Fawcett & Tucker (1990). The principles underlying the construction of the system network used in Communal require that that network represent more directly the kind of semantic distinctions that are in Penman located in the inquiries and 'environment' (cf. Figure 12). This complicates the construction of grammatical structure somewhat, but frees the network, in theory, from the need to capture syntactic structural differences. Two further respects in which Communal differs from Penman are in the use of probabilities for selecting features from the system network and in the development of lexical resources as an integrated part of the system network. Both these techniques have strong foundations within systemic-functional linguistic theory. A very large description of English 'lexicogrammar' has been developed in this framework as well as detailed work on dialogue (cf. Fawcett & Davies 1992).

As in most areas of computational linguistics, earlier work on text and sentence generation focused predominantly on English. This has now changed significantly and one central theme in NLG is *multilingual generation*, where NLG systems capable of generating in more than one language are considered. Systemic sentence generation has also moved away from providing accounts of English. The first such move was the systemic sentence generator for Japanese developed at Kyoto University by Bateman, Kikui & Tabuchi (1987), with its further experimental extension to Chinese (Bateman & Li 1988). This demonstrated that the systemic model of generation was equally applicable to languages other than English. Subsequently, generation facilities for German were built up within the KOMET system starting from a systemically-inspired treatment of German grammar developed by Steiner, Eckert, Weck & Winter (1988) in the context of the large EUROTRA machine translation project. The KOMET system has since come to include quite extensive generation grammars for both German and Dutch (cf. Teich, Degand & Bateman 1996). Joint work involving Matthiessen's Multilingual Generation Group in Sydney and the KOMET project has now generalized on these approaches to produce a systemic account of multilingual linguistic descriptions (Matthiessen, Nanri & Zeng 1991, Bateman, Matthiessen & Zeng 1996). Generation fragments for a number of languages have been constructed within this framework, including work on Chinese (Zeng 1993), Japanese (Hauser 1995) and French (Paris, Linden, Fischer, Hartley, Pemberton, Power & Scott 1995). This work is continuing with active contact with both noncomputational multi-

lingual systemic descriptions and theories of translation.

With the development of ever larger sets of linguistic resources for sentence generation, the availability of suitable software environments for managing these resources becomes crucial. This problem was first addressed within the Penman project where a window-oriented grammar writers tool was developed drawing in part on user-interface ideas developed within the Kyoto systemic generation project. A number of systemic generation systems have since paid attention to the needs of the grammar writer. O'Donnell's (1994) WAG system ('Workbench for Analysis and Generation') includes graphically-based debugging and editing facilities for systemic grammars alongside the generation/analysis capabilities of the system. Similarly, Kumano, Tokunaga, Inui & Tanaka's (1994) GENESYS system includes a sophisticated graphical presentation of the linguistic resources under development. The former system adopts a knowledge representation-like language as its implementational foundation, the latter adopts a Prolog-encoding of Functional Unification Grammar. Finally, the KPML development environment (Bateman 1996) attempts to provide the definitive user's development environment within a Penman-style architecture augmented by multilinguality; this system is freely available for non-commercial use.

Although the Penman-style generation architecture provides a stable and well understood platform for generation work, it also has theoretical disadvantages and newer solutions are being investigated. The first such investigations considered encoding of system networks in terms of other formalisms; we return to this below. More recently, Zeng's (1995) MULTEX system attempts a thorough reconsideration of a generation architecture employing systemic grammar. Zeng's approach is to allow the generation process to consult and explicitly reason about the linguistic descriptions that are available in order to plan how best to generate a sentence or text. This moves away from the simple traversal found in previous systemic generation systems in favor of a dynamic, planned use of linguistic resources as motivated by the particular communicative intentions. This offers a very flexible, if highly experimental, generation architecture.

To conclude this brief overview, we should note that there are a further range of significant systems that have adopted systemic sentence generation as their basic technology for tactical generation. These include: TechDoc (a system for producing technical documentation: cf. Rösner & Stede 1994), GIST (a system for producing instructions for completing administrative forms: Not & Stock 1994), Drafter (a technical writer's support tool for semi-automating software documentation: Paris, Linden, Fischer, Hartley, Pemberton, Power & Scott 1995) and HealthDoc (a system for producing health information leaflets: DiMarco, Hirst, Wanner & Wilkinson 1995). Further projects are in progress. There are also interesting combinations of systemic generation with other theoretical accounts; for example, Gagnon & Lapalme's (1996) combination of Discourse Representation Theory (DRT) for semantics and systemic grammar for French generation and the combination of ideas from Head-driven Phrase Structure Grammar (HPSG) and systemic grammar pursued in Elhadad & Robin's (1996) SURGE sentence generator for English.

7 Formalisations of systemic grammar

Considering systemic grammar's status as a *functional* approach to language—i.e., one that is not primarily formally oriented—there has nevertheless been a surprisingly long history of

interest in providing formalizations. This is largely due to the even longer interaction between systemic theory and computation: one of the first attempted computational applications of systemic theory was Halliday's (1956) discussion of possibilities for machine translation.

Styles of formalization have changed over this time. Earlier formalizations of systemic grammars focused on providing more explicit specifications of the generation process (cf. Henrici 1965, Patten & Ritchie 1987). Patten and Ritchie provide a particularly detailed logical model of systemic grammars employing familiar formal objects such as trees for the representation of syntactic structure and production rules for the derivation of collections of grammatical features; realization statements are modelled as logical constraints on structures. This formalization is in part motivated by the requirements of Patten's generation system SLANG described above. Since this system no longer assumes that traversal of a network begins with a single set of 'left-most', least delicate grammatical features as in Penman, but instead from an arbitrary set of grammatical features selected around the network at large, the basic traversal flowchart illustrated in Figure 8 above is no longer sufficient. Patten and Ritchie's formalization describes instead how a complete selection expression can be constructed by a set of production rules triggered by the preselected features. The Penman algorithm is simply a special case of the more general generation problem.

More recently, it has become usual to draw comparisons between systemic grammars and 'feature grammars'. Probably the first such comparison was, however, that of Winograd (1983). Modern structural approaches to syntax generally adopt descriptions formulated in terms of constraints on allowable combinations of grammatical features (cf. Gazdar, Klein, Pullum & Sag 1985, Chomsky 1981, Pollard & Sag 1987, Emele, Heid, Momma & Zajac 1990). Seen very abstractly, this also offers an accurate description of systemic grammar. This similarity supports formalizations in terms of other formal systems now more commonly used within computational linguistics for describing feature grammars.

The first such account was Kasper's (1988) attempt to overcome the non-declarative, generation-biased implementation of systemic grammars found in the Penman system. Kasper considered an implementation of SFG for analysis and parsing using an extension of the formal feature structure formalism adopted within Kay's (1979) Functional Unification Grammar (FUG). Here, the grammar network was translated into one large feature structure. Analysis was then attempted by unifying the input (a string) against the feature structure representing the grammar. Although theoretically revealing, the FUG implementation was not able to fully capture the necessary partial ordering of systems within the SFG: this is an essential means for restricting the distinctions that are considered during unification to just those that are licensed by the current paradigmatic or syntagmatic context. In addition, the implementation required the addition of a context-free phrase structure component in order for initial bundles of grammatical features to be gathered for unification to proceed. The FUG implementation was accordingly both slow and theoretically inadequate.

Since then, the FUG-style approach has itself evolved to more closely resemble certain crucial aspects of the systemic approach. Particularly, the use of inheritance hierarchies and typed feature structures within computational linguistics (cf. Pollard & Sag 1987) provides formal mechanisms more suited to formalizing the connectivity and interdependencies of a system network. Accordingly, Bateman, Emele & Momma (1992) outline a re-representation of systemic grammar based on typed-feature structures where the features of the system network are recoded directly as types in a type lattice. The dependency between systems in a system

network is then directly modelled as subsumption relations in a type lattice.

Such formalizations in terms of feature structures have improved the declarative status of systemic accounts and are important preconditions for achieving bi-directional computational linguistic systemic descriptions. There are still, however, substantial problems. Any straightforward formalization of a full systemic grammar runs into serious problems of scale and complexity. The extensive use of multiple inheritance within systemic grammars multiplies the descriptions provided explosively. The most recent formalization of system networks is developed by Henschel (1997). Henschel reviews the previous possibilities for the formal modelling of systemic grammars and provides both a formalization and a general construction algorithm for converting arbitrary system networks into various kinds of formal type lattice. Coding schemes for systemic grammars in several state of the art implementations of typed feature structures are also given. These are then combined with the construction algorithm in order to automatically generate typed feature definitions of substantial grammar fragments (including the full Nigel grammar of English). Unfortunately, Henschel finds that few of the available implementations can support type lattices of the required complexity and that individual systems differ in the precise functionalities covered. Similar problems face approaches that apply techniques from knowledge representation to the formal modelling of systemic grammars; Kasper (1989*b*), for example, explores the use of *classification* from knowledge representation for analysis using systemic grammars and O'Donnell's (1994) WAG system attempts both generation and analysis within this style of formalisation. But whether declarative representations of full-sized systemic grammars will be achieved in the near future, or whether principled simplifications such as those described in Brew (1991) will be necessary, remains an open question.

Regardless of implementation, such formalizations of systemic grammar serve to show clearly that, in many respects, SFG belongs to the same family of grammatical frameworks as do GPSG and HPSG: SFG is also based on statements of the co-occurrence possibilities of grammatical features . The main difference is in the *motivations* employed for those features. As has been suggested above, whereas the motivation for features within most more recently developed grammar frameworks remains primarily syntagmatic, within SFG the principal motivation for grammatical features is to provide an account of the communicative functions that particular linguistic units may serve. It is then these functions, rather than the structural regularities of syntax, that determine the organization of the grammar.

It is now clear both that the paradigmatic orientation of systemic grammar can be usefully complemented by a stronger degree of syntagmatic description and that the structural approaches can benefit from a stronger paradigmatic orientation. There have been several attempts to achieve such a unification starting from systemic grammar (cf. Yang, McCoy & Vijay-Shanker 1991, Elhadad & Robin 1996, Teich 1996). Until the paradigmatic aspects of linguistic representation are properly covered, the status of constructs such as lexical rules in approaches like HPSG will remain less clear and difficult to apply in NLG.

Finally, we should note that despite the current centrality of typed-feature representations in computational linguistics, other approaches to modelling are possible and that these may also bring useful results. One example is the completely different line of 'formalization' that has been pursued recently by Matthiessen (1995*a*) in terms of fuzzy systems. Parallel, connectionist, and statistical approaches may each prove themselves to be similarly relevant.

8 Future directions: multilinguality, multimodality, multimedia

The current explosion in the availability of more or less formally represented information—and the provision of tools for creating such information—creates a corresponding need for sophisticated presentation methods. Internally represented information is useless unless it is possible to present that information in terms that its potential users can understand. The forms of presentation supported even *determine* to a large extent who the users can be. If the presented form remains closely bound to the computer-internal representation then the potential users are restricted to just those familiar with that form of representation. NLG offers a very significant and flexible solution to this problem. It opens up the style of presentation to include natural language—with which all users are familiar. Moreover, NLG offers techniques for enforcing consistency during text production (e.g., consistent selection of terminology, desired grammatical forms, 'controlled languages', etc.), customizing texts for particular audiences or expertise levels, and maintaining up-to-date texts in the face of changing situations and requirements—all major problems for real document production. Computational approaches that provide for efficient and flexible NLG, such as the marriage of systemic grammar and sentence generation described here, are then crucial foundations for supporting the required rapid development in information presentation.

This foundational role of the systemic-functional approach will also serve well for the extensions to traditional sentence generation that are now necessary. It is no longer appropriate that generation systems concern themselves solely with 'texts' as strings of words and characters. First, the role of spoken language systems will grow considerably: the need to generate spoken language as flexibly as written language is already with us. Second, there is considerable demand that information system users are able to interact with systems in their own language and that they should have access to information that may not originate in their own language. Therefore, as mentioned above, the role of 'multilingual NLG' is also important. Finally, future applications require that 'text' is just one aspect of the information presented: such text has to be used effectively together with visual, graphical, and acoustic information as needed. Current work in NLG generally increasingly seeks to combine presentation media (cf., e.g., Maybury 1993): documents consisting only of text will not be acceptable.

It is then significant that there is both computational and theoretical work within the systemic-functional approach on all of these areas of extension. Systemic-functional linguistics considers itself as one aspect of a *socio-functional semiotics* and is, as such, not limited narrowly to 'language' as traditionally conceived. Spoken language has long been a central concern of systemic linguistics, but there are also systemically inspired approaches to, for example, both music (Steiner 1988) and visual images and graphics (Kress & van Leeuwen 1990). This work is beginning to influence computational accounts. Work on the combination of presentation media for computational systems is reported by Matthiessen, Kobayashi, Zeng & Cross (1995) and a description of the interrelationship of text generation and automatic page layout/design is given by Reichenberger, Rondhuis, Kleinz & Bateman (1996). It is therefore likely that the systemic approach to generation will continue to play a significant role in future generation systems—even when those systems are concerned with a far broader set of capabilities than currently associated with 'sentence generation'.

Suggestions for further reading

Basic introductory texts to the processes of natural language generation include McKeown (1985), Hovy (1988), and Paris (1993). A longer discussion of the particular interaction between systemic linguistics and natural language generation is given by Matthiessen & Bateman (1991). Brief up to date overviews of the field of NLG from different perspectives are given by Bateman (to appear) and de Smedt, Horacek & Zock (1996); useful previous reviews include: McKeown & Swartout (1987), and Bateman & Hovy (1992).

The state of formalization of systemic grammar in terms of modern computational linguistic formalisms is given in Henschel (1997). Uses of systemic grammar for spoken language generation are given by Fawcett (1990) and Teich, Hagen, Grote & Bateman (1997).

A reasonably extensive view of the theoretical assumptions of SFL as a whole and the role of grammar is given in Halliday (1978). A good introduction to a systemic grammar of English can be found in Halliday (1985); this is developed further in considerable detail in Matthiessen (1995*b*).

References

Androutsopoulos, I., Ritchie, G. & Thanisch, P. (forthcoming), 'Natural language interfaces to databases–an introduction', *Journal of Language Engineering* . (Also available as Research Paper no. 709, Department of Artificial Intelligence, University of Edinburgh, 1994).

Bateman, J. A. (1992), The theoretical status of ontologies in natural language processing, *in* S. Preuß & B. Schmitz, eds, 'Text Representation and Domain Modelling – ideas from linguistics and AI', KIT-Report 97, Technische Universität Berlin, pp. 50 – 99. (Papers from KIT-FAST Workshop, Technical University Berlin, October 9th - 11th 1991).
*http://www.darmstadt.gmd.de/publish/komet/papers/tu-paper.ps

Bateman, J. A. (1996), *KPML Development Environment: multilingual linguistic resource development and sentence generation*, German National Center for Information Technology (GMD), Institute for integrated publication and information systems (IPSI), Darmstadt, Germany. (Release 1.0).
*http://www.darmstadt.gmd.de/publish/komet/kpml.html

Bateman, J. A. (to appear), Automatic discourse generation, *in* A. Kent, ed., 'Encyclopedia of Library and Information Science', Marcel Dekker, Inc., New York.

Bateman, J. A., Emele, M. & Momma, S. (1992), The nondirectional representation of Systemic Functional Grammars and Semantics as Typed Feature Structures, *in* 'Proceedings of COLING-92', Nantes, France.

Bateman, J. A., Henschel, R. & Rinaldi, F. (1995), Generalized upper model 2.0: documentation, Technical report, GMD/Institut für Integrierte Publikations- und Informationssysteme, Darmstadt, Germany.
*http://www.darmstadt.gmd.de/publish/komet/gen-um/newUM.html

Bateman, J. A. & Hovy, E. H. (1992), Computers and text generation: principles and uses, *in* C. S. Butler, ed., 'Computers and written texts', Basil Blackwell, Oxford, England and Cambridge, Massachusetts, pp. 53 – 74. (Applied Language Studies).

Bateman, J. A., Kikui, G.-i. & Tabuchi, A. (1987), Designing a computational systemic grammar of Japanese for text generation: a progress report, Technical report, Kyoto University, Dept. of Electrical Engineering, Kyoto, Japan.

Bateman, J. A. & Li, H. (1988), The application of systemic-functional grammar to Japanese and Chinese for use in text generation, *in* 'Proceedings of the 1988 International Conference on Computer Processing of Chinese and Oriental Languages', Toronto, Canada, pp. 443–447.

Bateman, J. A., Maier, E. A., Teich, E. & Wanner, L. (1991), Towards an architecture for situated text generation, *in* 'International Conference on Current Issues in Computational Linguistics', Penang, Malaysia. Also available as technical report of GMD/Institut für Integrierte Publikations- und Informationssysteme, Darmstadt, Germany.

Bateman, J. A., Matthiessen, C. M. & Zeng, L. (1996), A general architecture of multilingual resources for natural language processing, Technical report, Institute for Integrated Publication and Information Systems (German National Research Centre for Information Technology, Darmstadt) and Macquarie University, Sydney.

Bateman, J. A. & Paris, C. L. (1989), Phrasing a text in terms the user can understand, *in* 'Proceedings of the Eleventh International Joint Conference on Artificial Intelligence', IJCAI-89, Detroit, Michigan.

Bateman, J. A. & Teich, E. (1995), 'Selective information presentation in an integrated publication system: an application of genre-driven text generation', *Information Processing and Management* **31**(5), 753–768. (Special Issue on Summarizing Text).

Bühler, K. (1934), *Sprachtheorie: die Darstellungsfunktion der Sprache*, Fischer, Jena.

Bresnan, J. (1982), The passive in lexical theory, *in* J. Bresnan, ed., 'The Mental Representation of Grammatical Relations', M.I.T. Press, Cambridge, Massachusetts, pp. 3–86.

Brew, C. (1991), 'Systemic Classification and its Efficiency', *Computational Linguistics* **17**(4), 375 – 408.

Cawsey, A., Binsted, K. & Jones, R. (1995), Personalised explanations for patient education, *in* 'Proceedings of the Fifth European Workshop on Natural Language Generation,Leiden, the Netherlands, 20-22 May 1995', pp. 59–74.

Chomsky, N. (1981), *Lectures on Government and Binding*, Foris, Dordrecht.

Coch, J., David, R. & Magnoler, J. (1995), Quality test for a mail generation system, *in* 'Proceedings of Linguistic Engineering '95', Montpellier, France.

Copestake, A., Flickinger, D., Malouf, R., Riehemann, S. & Sag, I. (1995), Translation using Minimal Recursion Semantics, *in* 'Proceedings of the 6th. International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-95)', Leuven, Belgium.

Cross, M. (1992), Choice in text: a systemic approach to computer modelling of variant text production, PhD thesis, School of English and Linguistics, Macquarie University, Sydney, Australia.

Davey, A. (1978), *Discourse Production: A Computer Model of Some Aspects of a Speaker*, Edinburgh University Press, Edinburgh. Published version of Ph.D. dissertation, University of Edinburgh, 1974.

de Saussure, F. (1959/1915), *Course in General Linguistics*, Peter Owen Ltd., London. (translated by W.Baskin).

de Smedt, K., Horacek, H. & Zock, M. (1996), Architectures for natural language generation: problems and perspectives, *in* G. Adorni & M. Zock, eds, 'Trends in natural language generation: an artificial intelligence perspective', number 1036 *in* 'Lecture Notes in Artificial Intelligence', Springer-Verlag, pp. 17–46.

DiMarco, C., Hirst, G., Wanner, L. & Wilkinson, J. (1995), Healthdoc: Customizing patient information and health education by medical condition and personal characteristics, *in* A. Cawsey, ed., 'Proceedings of the Workshop on Patient Education', University of Glasgow, Glasgow.

Elhadad, M. & Robin, J. (1996), A reusable comprehensive syntactic realization component, *in* 'Demonstrations and Posters of the 1996 International Workshop on Natural Language Generation (INLG '96)', Herstmonceux, England, pp. 1–4.

Emele, M. C., Heid, U., Momma, S. & Zajac, R. (1990), Organizing linguistic knowledge for multilingual generation, *in* '13th. International Conference on Computational Linguistics (COLING-90)', Helsinki, Finland. Also available as: Project Polygloss Paper, Institut für Maschinelle Sprachverarbeitung, University of Stuttgart, Germany.

Emele, M., Heid, U., Momma, S. & Zajac, R. (1992), 'Interactions between linguistic constraints: Procedural vs. declarative approaches', *Machine Translation* **6**(1). (Special edition on the role of text generation in MT).

Engelien, B. & McBryde, R. (1991), *Ovum: Natural Language Markets: Commercial Strategies*, Ovum Ltd., London, England.

Fawcett, R. P. (1990), The computer generation of speech with discoursally and semantically motivated intonation, *in* '5th. International Workshop on Natural Language Generation, 3-6 June 1990', Pittsburgh, PA.

Fawcett, R. P. & Davies, B. L. (1992), Monologue as a turn in dialogue: towards an integration of exchange structure theory and rhetorical structure theory, *in* R. Dale, E. Hovy, D. Rösner & O. Stock, eds, 'Aspects of automated natural language generation', Springer-Verlag, pp. 151 – 166. (Proceedings of the 6th International Workshop on Natural Language Generation, Trento, Italy, April 1992).

Fawcett, R. P. & Tucker, G. H. (1990), Demonstration of GENESYS: a very large, semantically based systemic functional grammar, *in* '13th. International Conference on Computational Linguistics (COLING-90)', Vol. I, Helsinki, Finland, pp. 47 – 49.

Firth, J. (1957/1935), The technique of semantics, *in* 'Papers in linguistics 1934-1951', Oxford University Press, London, pp. 7 – 33.

Friedman, J. (1969), 'Directed random generation of sentences', *Communications of the Association for Computing Machinery* **12**(6).

Gagnon, M. & Lapalme, G. (1996), Prétexte: a generator for the expression of temporal information, *in* G. Adorni & M. Zock, eds, 'Trends in natural language generation: an artificial intelligence perspective', number 1036 *in* 'Lecture Notes in Artificial Intelligence', Springer-Verlag, pp. 238–259.

Gazdar, G., Klein, E., Pullum, G. & Sag, I. A. (1985), *Generalized Phrase Structure Grammar*, Blackwell Publishing and Harvard University Press, Oxford, England and Cambridge, Massachusetts.

Gruber, T., Vemuri, S. & Rice, J. (1995), Virtual documents that explain how things work: Dynamically generated question-answering documents, Technical report, Knowledge Systems Laboratory, Stanford.
*http://www-ksl.stanford.edu/people/gruber/virtual-documents-htw/

Halliday, M. (1974), *Language and Social Man*, Longman, London. Schools Council Programme in Linguistics and English, Teaching Papers 11, 3.

Halliday, M. A. (1956), 'The linguistic basis of a mechanical thesaurus, and its application to English preposition classification', *Mechanical Translation* **3**.

Halliday, M. A. (1963), 'Class in relation to the axes of chain and choice in language', *Linguistics* **2**, 5 – 15. Reprinted in abbreviated form in Gunther R. Kress (ed.)(1976) *Halliday: system and function in language*, London: Oxford University Press, pp84-87.

Halliday, M. A. (1978), *Language as social semiotic*, Edward Arnold, London.

Halliday, M. A. (1985), *An Introduction to Functional Grammar*, Edward Arnold, London.

Hauser, B. (1995), 'Multilinguale Textgenerierung am Beispiel des Japanischen'. (Diplomarbeit).

Henrici, A. (1965), Notes on the systemic generation of a paradigm of the English clause, Technical report, O.S.T.I. Programme in the Linguistic Properties of Scientific English. (Reprinted in M.A.K. Halliday and J.R. Martin (Eds., 1981) *Readings in Systemic Linguistics*, London: Batsford.).

Henschel, R. (1997), Compiling systemic grammar into feature logic systems, *in* S. Manandhar, ed., 'Proceedings of CLNLP', Springer.

Hovy, E. H. (1987), Some pragmatic decision criteria in generation, *in* G. Kempen, ed., 'Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics', Kluwer Academic Publishers, Boston/Dordrecht, pp. 3–17.

Hovy, E. H. (1988), *Generating natural language under pragmatic constraints*, Lawrence Erlbaum, Hillsdale, New Jersey.

Kamp, H. & Reyle, U. (1993), *From discourse to logic: introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*, Kluwer Academic Publishers, London, Boston, Dordrecht. Studies in Linguistics and Philosophy, Volume 42.

Kasper, R. T. (1988), An Experimental Parser for Systemic Grammars, *in* 'Proceedings of the 12th International Conference on Computational Linguistics, August 1988', Association for Computational Linguistics, Budapest, Hungary. Also available as Information Sciences Institute Technical Report No. ISI/RS-88-212, Marina del Rey, CA.

Kasper, R. T. (1989*a*), A flexible interface for linking applications to PENMAN's sentence generator, *in* 'Proceedings of the DARPA Workshop on Speech and Natural Language'. Available from USC/Information Sciences Institute, Marina del Rey, CA.

Kasper, R. T. (1989*b*), Unification and classification: an experiment in information-based parsing, *in* 'Proceedings of the International Workshop on Parsing Technologies', pp. 1–7. 28-31 August, 1989, Carnegie-Mellon University, Pittsburgh, Pennsylvania.

Kay, M. (1979), Functional grammar, *in* 'Proceedings of the 5th meeting of the Berkeley Linguistics Society', Berkeley Linguistics Society, pp. 142 – 158.

Kittredge, R., Polguère, A. & Goldberg, E. (1986), Synthesizing weather reports from formatted data, *in* 'Proceedings of the 11th. International Conference on Computational Linguistics', International Committee on Computational Linguistics, Bonn, Germany, pp. 563 – 565.

Kress, G. & van Leeuwen, T. (1990), *Reading Images*, Sociocultural aspects of language and education, Deakin University Press, Geelong, Victoria, Australia.

Kumano, T., Tokunaga, T., Inui, K. & Tanaka, H. (1994), Genesys: an integrated environment for developing systemic functional grammars, *in* 'Proceedings of the Workshop on Sharable Natural Language Resources', Nara, Japan. Also available as: Department of Computer Science, Tokyo Institute of Technology, Technical Report: 94TR-0028.

Levine, J. & Mellish, C. (1994), CORECT: combining CSCW with natural language generation for collaborative requirements capture, *in* 'Proceedings of the 7th. International Workshop on Natural Language generation (INLGW '94)', Kennebunkport, Maine, pp. 236–239.

Li, P., Evens, M. & Hier, D. (1986), Generating medical case reports with the linguistic string parser, *in* 'Proceedings of 5th. National Conference on Artificial Intelligence (AAAI-86)', Philadelphia, PA, pp. 1069–1073.

Malinowski, B. (1923), *Supplement I*, Harcourt, Brace, and Co., Inc. Supplement to C.K. Ogden and I.A. Richards *The Meaning of Meaning*.

Mann, W. C. (1985), An introduction to the Nigel text generation grammar, *in* J. D. Benson & W. S. Greaves, eds, 'Systemic Perspectives on Discourse: Selected Theoretical Papers from the 9th. International Systemic Workshop', Ablex Pub. Corp., Norwood, N.J., pp. 84–95.

Mann, W. C. & Matthiessen, C. M. (1985), Demonstration of the Nigel text generation computer program, *in* J. D. Benson & W. S. Greaves, eds, 'Systemic Perspectives on Discourse, Volume 1', Ablex, Norwood, New Jersey.

Matthiessen, C. (1995*a*), Fuzziness construed in language: a linguistic perspective, *in* 'Proceedings of the International Joint Conference of the 4th. IEEE International Conference on Fuzzy Systems and the 2nd. International Fuzzy Engineering Symposium', pp. 1871–1878.

Matthiessen, C. M. (1981), A grammar and a lexicon for a text production system, *in* 'Proceedings of the 19th Annual Meeting of the Association for Computational Linguistics'.

Matthiessen, C. M. (1985), The systemic framework in text generation: Nigel, *in* J. D. Benson & W. S. Greaves, eds, 'Systemic Perspectives on Discourse, Volume 1', Ablex, Norwood, New Jersey, pp. 96–118.

Matthiessen, C. M. (1987), Notes on the organization of the environment of a text generation grammar, *in* G. Kempen, ed., 'Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics', Kluwer Academic Publishers, Boston/Dordrecht. Paper presented at the Third International Workshop on Natural Language Generation, August 1986, Nijmegen, The Netherlands.

Matthiessen, C. M. (1995*b*), *Lexicogrammatical cartography: English systems*, International Language Science Publishers, Tokyo, Taipei and Dallas.

Matthiessen, C. M. & Bateman, J. A. (1991), *Text generation and systemic-functional linguistics: experiences from English and Japanese*, Frances Pinter Publishers and St. Martin's Press, London and New York.

Matthiessen, C. M., Kobayashi, I., Zeng, L. & Cross, M. (1995), Generating multimodal presentations: resources and processes, *in* 'Proceedings of the Australian Conference on Artificial Intelligence', Canberra.

Matthiessen, C. M., Nanri, K. & Zeng, L. (1991), Multilingual resources in text generation: ideational focus, *in* 'Proceedings of the 2nd Japan-Australia Joint Symposium on Natural Language Processing', Kyushu Institute of Technology, Kyushu, Japan.

Maybury, M. T. (1993), *Intelligent Multimedia Interfaces*, AAAI Press and MIT Press, Cambridge, Massachusetts.

McCord, M. C. (1977), 'Procedural systemic grammars', *International Journal of Man-Machine Studies* **9**, 255–286.

McDonald, D. D. (1993), 'Issues in the choice of a source for natural language generation', *Computational Linguistics* **19**(1), 191 – 197.

McKeown, K. R. (1985), *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*, Cambridge University Press, Cambridge, England.

McKeown, K. R. & Swartout, W. R. (1987), Language generation and explanation, *in* 'Annual Reviews in Computer Science'.

Milosavljevic, M. & Dale, R. (1996), Text generation and user modelling on the web, *in* 'Proceedings of *User Modelling for Information Filtering on the World Wide Web* workshop at the *User Modelling '96* conference'.
*http://www-comp.mpce.mq.edu.au/msi/people/mariam/nlg-um-www.html

Not, E. & Stock, O. (1994), Automatic generation of instructions for citizens in a multilingual community, *in* 'Proceedings of the European Language Engineering Convention', Paris, France.

O'Donnell, M. (1994), Sentence analysis and generation: a systemic perspective, PhD thesis, University of Sydney, Department of Linguistics, Sydney, Australia.

Paris, C. L. (1993), *User modelling in text generation*, Pinter Publishers, London.

Paris, C., Linden, K. V., Fischer, M., Hartley, A., Pemberton, L., Power, R. & Scott, D. (1995), A support tool for writing multilingual instructions, *in* 'Proceedings of International Joint Conference on Artificial Intelligence', Montréal, Canada, pp. 1398–1404.

Patil, R. S., Fikes, R. E., Patel-Schneider, P. F., McKay, D., Finin, T., Gruber, T. R. & Neches, R. (1992), The DARPA knowledge sharing effort: progress report, *in* C. Rich, B. Nebel & W. R. Swartout, eds, 'Principles of knowledge representation and reasoning: proceedings of the third international conference', Morgan Kaufmann, Cambridge, MA.

Patten, T. (1988), *Systemic Text Generation as Problem Solving*, Cambridge University Press, Cambridge, England.

Patten, T. & Ritchie, G. (1987), A formal model of systemic grammar, *in* G. Kempen, ed., 'Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics', Kluwer Academic Publishers, Boston/Dordrecht. Paper presented at the Third International Workshop on Natural Language Generation, August 1986, Nijmegen, The Netherlands.

Pollard, C. & Sag, I. A. (1987), *Information-based syntax and semantics: volume 1*, Chicago University Press, Chicago. Center for the Study of Language and Information; Lecture Notes Number 13.

Vander Linden, K., Cumming, S. & Martin, J. (1992), Using system networks to build rhetorical structures, *in* R. Dale, E. Hovy, D. Rösner & O. Stock, eds, 'Aspects of automated natural language generation', Springer-Verlag, pp. 183 – 198. (Proceedings of the 6th International Workshop on Natural Language Generation, Trento, Italy, April 1992).

Reichenberger, K., Rondhuis, K., Kleinz, J. & Bateman, J. A. (1996), Effective presentation of information through page layout: a linguistically-based approach., Technical Report Arbeitspapiere der GMD 970, Institut für Integrierte Publikations- und Informationssysteme (IPSI), GMD, Darmstadt. (Paper presented at the workshop: 'Effective Abstractions in Multimedia, Layout and Interaction', held in conjunction with ACM Multimedia '95, November 1995, San Francisco, California.).

Reiter, E., Mellish, C. & Levine, J. (1995), 'Automatic generation of technical documentation', *Applied Artificial Intelligence* **9**.

Rösner, D. & Stede, M. (1994), Generating multilingual documents from a knowledge base: the TECHDOC project, *in* 'Proceedings of the 15th. International Conference on Computational Linguistics (COLING 94)', Vol. I, Kyoto, Japan, pp. 339 – 346.

Sgall, P., Hajičová, E. & Panevová, J. (1986), *The Meaning of the Sentence in Its Semantic and Pragmatic Aspects*, Reidel Publishing Company, Dordrecht.

Shapiro, S. C. (1979), Generalized augmented transition network grammars for generation from semantic networks, *in* 'Proceedings of the Seventeenth Meeting of the Association for Computational Linguistics', pp. 25–29.

Sheremetyeva, S., Nirenburg, S. & Nirenburg, I. (1996), Generating patent claims from interactive input, *in* 'Proceedings of the 8th. International Workshop on Natural Language Generation (INLG'96)', Herstmonceux, England, pp. 61–70.

Shieber, S. M. (1993), 'The problem of logical-form equivalence', *Computational Linguistics* **19**(1), 179 – 190.

Shieber, S., van Noord, G., Pereira, F. & Moore, R. (1990), 'Semantic head-driven generation', *Computational Linguistics* **16**(1), 30–42.

Simmons, R. & Slocum, J. (1972), 'Generating English discourse from semantic nets', *Communications of the Association for Computing Machinery* **15**(10), 891–905.

Springer, S., Buta, P. & Wolf, T. (1991), Automatic letter composition for customer service, *in* R. Smith & C. Scott, eds, 'Innovative applications of artificial intelligence 3', AAAI Press. (Proceedings of CAIA-1991).

Steiner, E. (1988), The interaction of language and music as semiotic systems: the example of a folk ballad, *in* J. D. Benson, M. J. Cummings & W. S. Greaves, eds, 'Linguistics in a Systemic Perspective', Benjamins, Amsterdam, pp. 393–441.

Steiner, E. H., Eckert, U., Weck, B. & Winter, J. (1988), The development of the EUROTRA-D system of semantic relations, *in* E. H. Steiner, P. Schmidt & C. Zelinksy-Wibbelt, eds, 'From Syntax to Semantics: insights from Machine Translation', Frances Pinter, London.

Teich, E. (1996), A proposal for dependency in Systemic Functional Grammar: metasemiosis in Computational Systemic Functional Linguistics, PhD thesis, University of the Saarland, Saarbrücken, Germany.

Teich, E., Degand, L. & Bateman, J. A. (1996), Multilingual textuality: Experiences from multilingual text generation, *in* G. Adorni & M. Zock, eds, 'Trends in Natural Language Generation: an artificial intelligence perspective', number 1036 *in* 'Lecture Notes in Artificial Intelligence', Springer-Verlag, Berlin, New York, pp. 331–349. (Selected Papers from the 4th. European Workshop on Natural Language Generation, Pisa, Italy, 28-30 April 1993).

Teich, E., Hagen, E., Grote, B. & Bateman, J. (1997), 'From communicative context to speech: integrating dialogue processing, speech production and natural language generation', *Speech Communication* **21**(1–2).

Thompson, S. A. (1987), The passive in English: A discourse perspective, *in* R. Channon & L. Shockey, eds, 'In honor of Ilse Lehiste', Foris, Dordrecht.

van Noord, G. & Neumann, G. (1996), Syntactic generation, *in* R. A. Cole, J. Mariani, H. Uszkoreit, A. Zaenen & V. Zue, eds, 'Survey of State of the Art in Human Language Technology', Kluwer Academic Press, chapter 4.2. (Contribution to Chapter on 'Language Generation').

Whitelock, P. (1992), Shake-and-bake translation, *in* 'Proceedings of COLING-92', Vol. II, pp. 784 – 791.

Winograd, T. (1983), *Language as a Cognitive Process, Volume 1: Syntax*, Addison Wesley, New York, chapter 6: Feature and function grammars, pp. 272–310.

Yang, G., McCoy, K. F. & Vijay-Shanker, K. (1991), 'From functional specification to syntactic structures: systemic grammar and tree adjoining grammar', *Computational Intelligence* **7**(4), 207 –219.

Yngve, V. H. A. (1962), Random generation of English sentences, *in* 'The 1961 Conference on Machine Translation of Languages and Applied Language Analysis', Her Majesty's Stationery Office, London.

Zeng, L. (1993), Coordinating ideational and textual resources in the generation of multisentential texts in Chinese, *in* 'Proceedings of the Meeting of the Pacific Association of Computational Linguistics', Vancouver.

Zeng, L. (1995), Reasoning with systemic resources in text planning, *in* 'The 2nd. Conference of the Pacific Association for Computational Linguistics (PacLing-II)', Brisbane, Australia, pp. 317–328.