Chapter 4

Language Generation

4.1 Overview

Eduard Hovy

University of Southern California, Marina del Rey, California, USA

The area of study called natural language generation (NLG) investigates how computer programs can be made to produce high-quality natural language text from computer-internal representations of information. Motivations for this study range from entirely theoretical (linguistic, psycholinguistic) to entirely practical (for the production of output systems for computer programs). Useful overviews of the research are Dale, Hovy, et al. (1992); Paris, Swartout, et al. (1990); Kempen (1987); Bateman and Hovy (1992); McKeown and Swartout (1987); Mann, Bates, et al. (1981). The stages of language generation for a given application, resulting in speech output, are shown in Figure 4.1.

This section discusses the following:

- the overall state of the art in generation,
- ignificant gaps of knowledge, and
- new developments and infrastructure.

For more detail, it then turns to two major areas of generation theory and practice: single-sentence generation (also called realization or tactical generation) and multisentence generation (also called text planning or strategic generation).

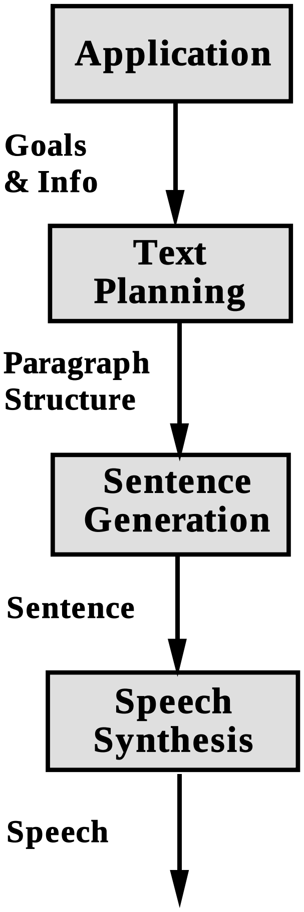

Figure 4.1: The stages of language generation.

4.1.1 State of the Art

No field of study can be described adequately using a single perspective. In order to understand NLG it is helpful to consider independently the *tasks* of generation and the *process* of generation. Every generator addresses one or more tasks and embodies one (or sometimes two) types of process. One can identify three types of generator *task*: text planning, sentence planning, and surface realization. Text planners select from a knowledge pool what information to include in the output, and out of this create a text structure to ensure coherence. On a more local scale, sentence planners organize the content of each sentence, massaging and ordering its parts. Surface realizers convert sentence-sized chunks of representation into grammatically correct sentences. Generator *processes* can be classified into points on a range of sophistication and expressive power, starting with inflexible canned methods and ending with maximally flexible feature combination methods. For each point on this range, there may be various types of implemented algorithms. 18 The simplest approach, **canned text systems**, is used in the majority of software: the system simply prints a string of words without any change (error messages, warnings, letters, etc.). The approach can be used equally easily for single-sentence and for multi-sentence text generation. Trivial to create, the systems are very wasteful. **Template systems**, the next level of sophistication, are used as soon as a message must be produced several times with slight alterations. Form letters are a typical template application, in which a few open fields are filled in specified constrained ways. The template approach is used mainly for multisentence generation, particularly in applications whose texts are fairly regular in structure such as some business reports. The text planning components of the U.S. companies CoGenTex (Ithaca, NY) and Cognitive Systems Inc. (New Haven, CT) enjoy commercial use. On the research side, the early template-based generator ANA (Kukich, 1983) produced stock market reports from a news wire by filling appropriate values into a report template. More sophisticated, the multisentence component of TEXT (McKeown, 1985) could dynamically nest instances of four stereotypical paragraph templates called schemas to create paragraphs. TAILOR (Paris, 1993a) generalized TEXT by adding schemas and more sophisticated schema selection criteria.

Phrase-based systems employ what can be seen as generalized templates, whether at the sentence level (in which case the phrases resemble phrase structure grammar rules) or at the discourse level (in which case they are often called text plans). In such systems, a phrasal pattern is first selected to match the top level of the input (say, [SUBJECT VERB OBJECT]), and then each part of the pattern is expanded into a more specific phrasal pattern that matches some subportion of the input (say, [DETERMINER ADJECTIVES HEAD-NOUN MODIFIERS]), and so on; the cascading process stops when every phrasal pattern has been replaced by one or more words. Phrase-based systems can be powerful and robust, but are very hard to build beyond a certain size, because the phrasal interrelationships must be carefully specified to prevent inappropriate phrase expansions. The phrase-based approach has mostly been used for single-sentence generation (since linguists' grammars provide well-specified collections of phrase structure rules). A sophisticated example is MUMBLE (McDonald, 1980; Meteer, McDonald, et al., 1987), built at the University of Massachusetts, Amherst. Over the past five years, however, phrase-based multisentence text structure generation (often called text planning) has received considerable attention in the research community, with the development of the RST text structurer (Hovy, 1988), the EES text planner (Moore, 1989), and several similar systems (Dale, 1990; Cawsey, 1989; Suthers, 1993), in which each so-called text plan is a *phrasal* pattern that specifies the structure of some portion of the discourse, and each portion of the plan is successively refined by more specific plans until the single-clause level is reached. Given the lack of understanding of discourse structure and the paucity of the discourse plan libraries, however, such planning systems do not yet operate beyond the experimental level.

Feature-based systems represent, in a sense, the limit point of the generalization of phrases. In feature-based systems, each possible minimal alternative of expression is represented by a single feature; for example, a sentence is either POSITIVE or NEGATIVE, it is a QUESTION or an IMPERATIVE or a STATEMENT, its tense is PRESENT or PAST and so on. Each sentence is specified by a unique set of features. Generation proceeds by the incremental collection of features appropriate for each portion of the input (either by the traversal of a feature selection network or by unification), until the sentence is fully determined. Feature-based systems are among the most sophisticated generators built today. Their strength lies in the simplicity of their conception: any distinction in language is defined as a feature, analyzed, and added to the system. Their strength lies in the simplicity of their conception: any distinction in language can be added to the system as a feature. Their weakness lies in the difficulty of maintaining feature interrelationships and in the control of feature selection (the more features available, the more complex the input must be). No feature-based multisentence generators have been built to date. The most advanced single-sentence generators of this type include PENMAN (Matthiessen, 1983; Mann & Matthiessen, 1985) and its descendant KPML (Bateman, Maier, et al., 1991), the Systemic generators developed at USC/ISI and IPSI; COMMUNAL (Fawcett, 1992) a Systemic generator developed at Wales; the Functional Unification Grammar framework (FUF) (Elhadad, 1992) from Columbia University; SUTRA (Von Hahn, Höppner, et al., 1980) developed at the University of Hamburg; SEMTEX (Rösner, 1986) developed at the University of Stuttgart; and POPEL (Reithinger, 1991) developed at the University of the Saarland. The two generators most widely distributed, studied, and used are PENMAN/KPML and FUF. None of these systems is in commercial use.

4.1.2 Significant Gaps and Limitations

It is safe to say that at the present time one can fairly easily build a single-purpose generator for any specific application, or with some difficulty adapt an existing sentence generator to the application, with acceptable results. However, one cannot yet build a general-purpose sentence generator or a non-toy text planner. Several significant problems remain without sufficiently general solutions:

- lexical selection
- sentence planning
- discourse structure

- domain modeling
- generation choice criteria

Lexical Selection: Lexical selection is one of the most difficult problems in generation. At its simplest, this question involves selecting the most appropriate single word for a given unit of input. However, as soon as the semantic model approaches a realistic size, and as soon as the lexicon is large enough to permit alternative locutions, the problem becomes very complex. In some situation, one might have to choose among the phrases *John's car, John's sports car, his speedster, the automobile, the red vehicle, the red Mazda* for referring to a certain car. The decision depends on what has already been said, what is referentially available from context, what is most salient, what stylistic effect the speaker wishes to produce, and so on. A considerable amount of work has been devoted to this question, and solutions to various aspects of the problem have been suggested (see for example Goldman (1975); Elhadad and Robin (1992); McKeown, Robin, et al. (1993)). At this time no general methods exist to perform lexical selection. Most current generator systems simply finesse the problem by linking a single lexical item to each representation unit. *What is required:* Development of theories about and implementations of lexical selection algorithms, for reference to objects, event, states, etc., and tested with large lexica.

Discourse Structure: One of the most exciting recent research developments in generation is the automated planning of paragraph structure. The state of the art in discourse research is described in chapter 6. So far no text planner exists that can reliably plan texts of several paragraphs in general. *What is required:* Theories of the structural nature of discourse, of the development of theme and focus in discourse, and of coherence and cohesion; libraries of discourse relations, communicative goals, and text plans; implemented representational paradigms for characterizing stereotypical texts such as reports and business letters; implemented text planners that are tested in realistic non-toy domains.

Sentence Planning: Even assuming the text planning problem solved, a number of tasks remain before well-structured multisentence text can be generated. These tasks, required for planning the structure and content of each sentence, include: pronoun specification, theme signaling, focus signaling, content aggregation to remove unnecessary redundancies, the ordering of prepositional phrases, adjectives, etc. An elegant system that addressed some of these tasks is described in (Appelt, 1985). While to the nonspecialist these tasks may seem relatively unimportant, they can have a significant effect and make the difference between a well-written and a poor text. *What is required:* Theories of pronoun use, theme and focus selection and signaling, and content aggregation; implemented sentence planners with rules that perform these operations; testing in realistic domains.

Domain Modeling: A significant shortcoming in generation research is the lack of large well-motivated application domain models, or even the absence of clear principles by which to build such models. A traditional problem with generators is that the inputs are frequently hand-crafted, or are built by some other system that uses representation elements from a fairly small hand-crafted domain model, making the generator's inputs already highly oriented toward the final language desired. It is very difficult to link a generation system to a knowledge base or database that was originally developed for some non-linguistic purpose. The mismatches between the representation schemes demonstrate the need for clearly articulated principles of linguistically appropriate domain modeling and representational adequacy (see also Meteer, 1990). The use of high-level language-oriented concept taxonomies such as the Penman Upper Model (Bateman, Moore, et al., 1990) to act as a *bridge* between the domain application's concept organization and that required for generation is becoming a popular (though partial) solution to this problem. *What is required:* Implemented large-size (over 10,000 concepts) domain models that are useful both for some non-linguistic application and for generation; criteria for evaluating the internal consistency of such models; theories on and practical experience in the linking of generators to such models; lexicons of commensurate size.

Generation Choice Criteria: Probably the problem least addressed in generator systems today is the one that will take the longest to solve. This is the problem of guiding the generation process through its choices when multiple options exist to handle any given input. It is unfortunately the case that language, with its almost infinite flexibility, demands far more from the input to a generator than can be represented today. As long as generators remain fairly small in their expressive potential then this problem does not arise. However, when generators start having the power of saying *the same thing* in many ways, additional control must be exercised in order to ensure that appropriate text is produced. As shown in Hovy (1988) and Jameson (1987), different texts generated from the same input carry additional, non-semantic import; the stylistic variations serve to express significant interpersonal and situational meanings (text can be formal or informal, slanted or objective, colorful or dry, etc.). In order to ensure appropriate generation, the generator user has to specify not only the semantic content of the desired text, but also its pragmatic—interpersonal and situational—effects. Very little research has been performed on this question beyond a handful of small-scale pilot studies. *What is required:* Classifications of the types of reader characteristics and goals, the types of author goals, and the interpersonal and situational aspects that affect the form and content of language; theories of how these aspects affect the generation process; implemented rules and/or planning systems that guide generator systems' choices; criteria for evaluating appropriateness of generated text in specified communicative situations.

4.1.3 Future Directions

Infrastructure Requirements: The overarching challenge for generation is scaling up to the ability to handle real-world, complex domains. However, given the history of relatively little funding support, hardly any infrastructure required for generation research exists today.

The resources most needed to enable both high-quality research and large-scale generation include the following:

- Large well-structured lexicons of various languages. Without such lexicons, generator builders have to spend a great deal of redundant effort, collecting standard morphological and syntactic information to include in lexical items. As has been shown recently in the construction of the Penman English lexicon of 90,000+ items, it is possible to extract enough information from online dictionaries to create lexicons, or partial lexicons, automatically.
- Large well-structured knowledge bases. Paralleling the recent knowledge base construction efforts centered around WordNet (Miller, 1985) in the U.S., a large general-purpose knowledge base that acts as support for domain-specific application oriented knowledge bases would help to speed up and enhance generator porting and testing on new applications. An example is provided by the ontology construction program of the Pangloss machine translation effort (Hovy & Knight, 1993).
- Large grammars of various languages. The general availability of such grammars would free generator builders from onerous and often repetitive linguistic work, though different theories of language naturally result in very different grammars. However, a repository of grammars built according to various theories and of various languages would constitute a valuable infrastructure resource.
- Libraries of text plans. As discussed above, one of the major stumbling blocks in the ongoing investigation of text planning is the availability of a library of tested text plans. Since no consensus exists on the best form and content of such plans, it is advisable to pursue several different construction efforts.

Longer-term Research Projects: Naturally, the number and variety of promising long-term research projects is large. The following directions have all been addressed by various researchers for over a decade and represent important strands of ongoing investigation:

- stylistically appropriate generation

- psycholinguistically realistic generation
- reversible multilingual formalisms and algorithms
- continued development of grammars and generation methods
- generation of different genres/types of text

Near- and Medium-term Applications with Payoff Potential: Taking into account the current state of the art and gaps in knowledge and capability, the following applications (presented in order of increasing difficulty) provide potential for near-term and medium-term payoff:

- **Database Content Display:** The description of database contents in natural language is not a new problem, and some such generators already exist for specific databases. The general solution still poses problems, however, since even for relatively simple applications it still includes unsolved issues in sentence planning and text planning.
- **Expert System Explanation:** This is a related problem, often however requiring more interactive ability, since the user's queries may not only elicit more information from a (static, and hence well-structured) database, but may cause the expert system to perform further reasoning as well, and hence require the dynamic explanation of system behavior, expert system rules, etc. This application also includes issues in text planning, sentence planning, and lexical choice.
- **Speech Generation:** Simplistic text-to-speech synthesis systems have been available commercially for a number of years, but naturalistic speech generation involves unsolved issues in discourse and interpersonal pragmatics (for example, the intonation contour of an utterance can express dislike, questioning, etc.). Only the most advanced speech synthesizers today compute syntactic form as well as intonation contour and pitch level.
- **Limited Report and Letter Writing:** As mentioned in the previous section, with increasingly general representations for text structure, generator systems will increasingly be able to produce standardized multiparagraph texts such as business letters or monthly reports. The problems faced here include text plan libraries, sentence planning, adequate lexicons, and robust sentence generators.
- **Presentation Planning in Multimedia Human-Computer Interaction:** By generalizing text plans, Hovy and Arens (1991) showed that it is possible also to control some forms of text formatting, and then argued that further generalization

will permit the planning of certain aspects of multimedia presentations. Ongoing research in the WIP project at Saarbrücken (Wahlster, André, et al., 1991) and the COMET project at Columbia University (Feiner & McKeown, 1990) have impressive demonstration systems for multimedia presentations involving planning and language generation.

- **Automated Summarization:** A somewhat longer-term functionality that would make good use of language generation and discourse knowledge is the automated production of summaries. Naturally, the major problem to be solved first is the identification of the most relevant information.

During the past two decades, language generation technology has developed to the point where it offers general-purpose single-sentence generation capability and limited-purpose multisentence paragraph planning capability. The possibilities for growth and development of useful applications are numerous and exciting. Focusing new research on specific applications and on infrastructure construction will help turn the promise of current text generator systems and theories into reality.

4.2 Syntactic Generation

Gertjan van Noord[a] & Günter Neumann[b]

[a] Alfa-informatica RUG, The Netherlands
[b] Deutsches Forschungzentrum für Künstliche Intelligenz, Saarbrücken, Germany

In a natural language generation module, we often distinguish two components. On the one hand it needs to be decided *what* should be said. This task is delegated to a *planning component*. Such a component might produce an expression representing the content of the proposed utterance. On the basis of this representation the syntactic generation component produces the actual output sentence(s). Although the distinction between planning and syntactic generation is not uncontroversial, we will nonetheless assume such an architecture here, in order to explain some of the issues that arise in syntactic generation.

A (natural language) grammar is a formal device that defines a relation between (natural language) utterances and their corresponding meanings. In practice this usually means that a grammar defines a relation between strings and logical forms. During natural language understanding, the task is to arrive at a logical form that corresponds to the input string. Syntactic generation can be described as the problem to find the corresponding string for an input logical form.

We are thus making a distinction between the grammar which *defines* this relation, and the procedure that *computes* the relation on the basis of such a grammar. In the current state of the art unification-based (or more general: constraint-based) formalisms are used to express such grammars, e.g., Lexical Functional Grammar (LFG) (Bresnan, 1982), Head-Driven Phrase-Structure Grammar (HPSG) (Pollard & Sag, 1987) and constraint-based categorial frameworks (cf. Uszkoreit, 1986 and Zeevat, Klein, et al., 1987).

Almost all modern linguistic theories assume that a natural language grammar not only describes the correct sentences of a language, but that such a grammar also describes the corresponding semantic structures of the grammatical sentences. Given that a grammar specifies the relation between phonology and semantics it seems obvious that the generator is supposed to use this specification. For example, Generalized Phrase Structure Grammars (GPSG) (Gazdar, Klein, et al., 1985) provide a detailed description of the semantic interpretation of the sentences licensed by the grammar. Thus one might assume that a generator based on GPSG constructs a sentence for a given semantic structure, according to the semantic interpretation rules of GPSG. Alternatively, Busemann (1990) presents a generator, based on GPSG, which does not take as its input a logical form, but rather some kind of control expression which merely instructs the grammatical component which rules of the grammar to apply. Similarly, in the conception of Gardent and Plainfossé (1990), a generator is provided with some kind of *deep structure* which can be interpreted as a control expression instructing the grammar which rules to apply. These approaches to the generation problem clearly *solve* some of the problems encountered in generation—simply by pushing the problem into the conceptual component (i.e., the planning component). In this overview we restrict the attention to the more ambitious approach sketched above.

The success of the currently developed constraint-based theories is due to the fact that they are purely declarative. Hence, it is an interesting objective—theoretically and practically—to use one and the same grammar for natural language understanding and generation. In fact the potential for reversibility was a primary motivation for the introduction of Martin Kay's Functional Unification Grammar (FUG). In recent years interest in such a *reversible* architecture has led to a number of publications.[1]

4.2.1 State of the Art

The different approaches towards the syntactic generation problem can be classified according to a number of dimensions. It is helpful to distinguish between

- Definition of the search space
    - Left-right vs. Bidirectional processing
    - Top-down vs. Bottom-up processing
- Traversal of the search space

A generator proceeds from left to right if the elements of the right-hand-side of a rule are processed in a left-to-right order. This order is very common for parsing, but turns out to be unsuitable for generation. For example, Shieber (1988) presents an Earley-based generation algorithm that follows a left-to-right scheduling. It has been shown that such a strategy leads to a very inefficient behavior when applied for generation. The reason is that the important information that guides the generation process, namely the logical forms, is usually percolated in a different manner. Therefore,

---

[1] See for example Strzalkowski, Carballo, et al. (1995); Strzalkowski (1994) which is a collection of papers based on the 1991 ACL workshop 'Reversible Grammars in Natural Language Processing'; some other references are Appelt (1987); Jacobs (1988); Dymetman and Isabelle (1988). However, it is currently a matter of debate, whether one and the same grammar should actually be employed at run-time by both processes without any change (e.g., Shieber, 1988; Shieber, Pereira, et al., 1990; VanNoord, 1993; Neumann, 1994) or whether two separate grammars should better be compiled out of a single source grammar (e.g., Block, 1994; Dymetman, Isabelle, et al., 1990; Strzalkowski, 1989.)

semantic-head-driven generation approaches have become popular, most notably the algorithm described in Shieber, Pereira, et al. (1990); VanNoord (1990); VanNoord (1993), but see also Calder, Reape, et al. (1989); Gerdemann and Hinrichs (1990); Gerdemann (1991); Neumann (1994). Such approaches aim at an order of processing in which an element of the right-hand-side of a rule is only processed once its corresponding logical form has been determined.

As in parsing theory, generation techniques can be classified according to the way they construct the derivation trees. Bottom-up and top-down traversals have been proposed as well as mixed strategies. For example, bottom-up generation strategies are described in Shieber (1988); VanNoord (1993), top-down approaches are described in Wedekind (1988); Dymetman, Isabelle, et al. (1990), and mixed strategies are described in Shieber, Pereira, et al. (1990); Gerdemann (1991); Neumann (1994).

As in parsing, bottom-up approaches solve some non-termination problems that are encountered in certain top-down procedures.

The above mentioned two dimensions characterize the way in which derivation trees are constructed. A particular choice of these parameters defines a non-deterministic generation scheme, giving rise to a search space that is to be investigated by an actual generation algorithm. Hence, generation algorithms can be further classified with respect to the search strategy they employ. For example, a generation algorithm might propose a depth-first backtrack strategy. Potentially more efficient algorithms might use a *chart* to represent successfully branches of the search space, optionally combined with a breadth-first search (see for example, Gerdemann, 1991; Calder, Reape, et al., 1989). Moreover, there also exist chart-based agenda driven strategies which allow the modeling of preference-based *best-first* strategies (e.g., Den, 1994; Neumann, 1994).

4.2.2 Future Directions

Syntactic generation is one of the most elaborated and investigated fields in the area of natural language generation. In particular, due to the growing research in the Computational Linguistics area, syntactic generation has now achieved a methodological status comparable to that of natural language parsing. However, there are still strong limitations which weakens their general applicability for arbitrary application systems. Probably the most basic problems are:

- Lexical and grammatical coverage
- Re-usability
- Limited functional flexibility

None of the syntactic generators process grammars whose size and status would go beyond that of a *laboratory* one. The newly proposed approaches in Computational Linguistics are in principle capable of processing declaratively specified grammars, and hence are potentially open to grammars which can be incrementally extented. However, as long as the grammars do not achieve a critical mass, the usability of the approaches for very large grammars is a speculation. The same is true for the status of the lexicons. Currently, generators only use small lexicons. Consequently most of the systems trivialize the problem of lexical choice as being a simple look-up method. However, if very large lexicons were to be used then the lexical choice problem would require more sophisticated strategies.

Of course, there exists some generators whose grammatical coverage is of interest, most notably those from the Systemic Linguistics camp (see section 4.1). However, these generation grammars have a less transparent declarative status, and hence are limited with respect to re-usability and adaptation to other systems.

All known syntactic generators have a limited degree of functionality. Although some approaches have been proposed for solving specific problems, such as generating ellipsis (e.g., Jameson & Wahlster, 1982); generation of paraphrases (e.g., Meteer & Shaked, 1988; Neumann, 1994); generation of referential expressions (Dale, 1990); or incremental generation (e.g., DeSmedt & Kempen, 1987), there exists currently no theoretical and practical framework, which could serve as a platform for combining all these specific operational issues.

Taking these limitations as a basis, important key research problems specific to syntactic generation are:

Large Grammars and Lexicons: These are needed for obtaining reasonable linguistic competence. As a prerequisite, grammatical knowledge must be specified declaratively in order to support the re-usability, not only for other systems, but also for integrating different specific generation performance methods.

Reversibility: If we want to obtain realistic generation systems then interleaving natural language generation and understanding will be important, e.g., for text revision. It is reasonable to assume that for the case of grammatical processing reversible grammars as well as uniform processing methods are needed. Such a uniform framework might also serve as a platform for integrating generation and understanding specific performance methods.

Incremental Processing: Rather than generating on the basis of a single complete logical form, some researchers have investigated the possibility of generating incrementally. In such a model small pieces of semantic information are provided to the tactical generator one at the time. Such a model might better explain certain psycholinguistic observations concerning human language production (cf. for example DeSmedt & Kempen, 1987).

Producing a non-Ambiguous Utterance: The generation procedures sketched above all come up with a possible utterance for a given meaning representation. However, given that natural language is very ambiguous the chances are that this proposed utterance itself is ambiguous, and therefore might lead to undesired *side-effects*. Some preliminary techniques to prevent the production of ambiguous utterances are discussed in Neumann and van Noord (1994); Neumann (1994).

Integration of Template- and Grammar-based Generation: This will be important in order to obtain efficient but flexible systems. This would allow competence grammar to be used in those cases where prototypical constructions (i.e., the templates) are not appropriate or even available.

Logical Equivalence: An important theoretical and practical problem for natural language generation is the problem of logical form equivalence. For a discussion of this problem we refer to Shieber (1993).

4.3 Deep Generation

John Bateman

GMD, IPSI, Darmstadt, Germany

Although crucial to the entire enterprise of automatic text generation, deep generation remains a collection of activities lacking a clear theoretical foundation at this time. The most widely accepted views on what constitutes deep generation are already exhausted by a small number of techniques, resources and algorithms revealing as many problems as they can really claim to solve. For these reasons, recent research work in text generation centers on aspects of deep generation and it is here that serious breakthroughs are most needed. Whereas the goal of deep generation is to produce specifications of sufficiently fine granularity and degree of linguistic abstraction to drive surface generators, how it is to do so, and from what starting point, remains unclear.

4.3.1 State of the Art

Although deep generation is most often seen as notionally involving two subtasks—selecting the content for a text and imposing an appropriate linear order on that content's expression—it is now usually accepted that this decomposition is problematic. The subtasks are sufficiently interdependent as to make such a decomposition questionable. Linear order is achieved by the intermediate step of constructing a recursive text structure, typically the province of *text planning.* The two standard methods for constructing text structure, *text schemata* (e.g., McKeown, 1985; McCoy, 1986; Rambox & Korelsky, 1992; Paris, 1993b) and *rhetorical structuring* (e.g., Mann & Thompson, 1987; Hovy, 1993; Moore & Paris, 1993), both combine content selection and textual organization.

Text schemata describe text on the model of constituency. A text is defined in terms of a *macro structure* with constituents given by *rhetorical predicates*, such as *Identification*, *Constituency*, and *Analogy.* Individual rhetorical predicates generally include both constraints on the information they express and particular surface realization constraints. Rhetorical predicates are combined in fixed configurations, the text schemata. The most commonly cited problems with text schemata are their rigidity and lack of intentional information (cf. Moore & Paris, 1993): i.e., if an *identification* predicate appears, there is no record as to why a speaker has selected this predicate. This is particularly problematic for dialogue situations where breakdowns can occur. Despite these problems, however, schemata are still sometimes selected on the basis of their simplicity and ease of definition (cf. Rambox & Korelsky, 1992).

In contrast to text schemata, rhetorical structures define the relational structure of a text. They show how a text can be recursively decomposed into smaller segments. These component segments are related to one another by means of a small set of *rhetorical relations*, such as *elaboration*, *solutionhood*, *volitional cause*, etc. Each such rhetorical relation is defined in terms of a distinctive set of constraints on the information presented in the segments related and in those segments' combination, on the speaker/hearer belief states, and on the effect that the speaker is attempting to achieve with the relation. It is generally assumed that imposing a rhetorical organization enables the information to be presented to be segmented into sufficiently small-scale chunks as to admit expression by surface generators. Rhetorical organization is typically constructed by using a top-down goal-oriented planning strategy with the rhetorical relation definitions as plan operators. However, while earlier rhetorical structure approaches tended to equate rhetorical relations with discourse intentions, this does not appear equally appropriate for all rhetorical relations. Those relations that are based on the informational content of the segments related underconstrain possible discourse intentions; for example, a *circumstance* relation can be given for many distinct discourse purposes. The most well developed versions of rhetorical structure-based text planning therefore separate out at least discourse intentions and rhetorical relations and allow a many-to-many relation between them, as defined by the system's planing operators.

An example of such a plan operator from the system of Moore and Paris (1993) is the following:

```
EFFECT: (PERSUADED ?hearer (DO ?hearer ?act))
CONSTRAINTS: (AND (STEP ?act ?goal)
                  (GOAL ?hearer ?goal)
                  (MOST-SPECIFIC ?goal)
                  (CURRENT-FOCUS ?act)
                  (SATELLITE))
NUCLEUS: (FORALL ?goal
                  (MOTIVATION ?act ?goal))
SATELLITES: nil
```

The successful application of this operator has the effect that a state of the hearer being *persuaded* (a discourse intention) to do some act is achieved. The operator may be applied when the specified constraints hold. When this is the case, a rhetorical structuring involving *motivation* is constructed. Information selection is thus achieved as a side-effect of binding variables in the operator's constraints. Further such plan operators then decompose the rhetorical relation *motivation* until sequences of surface speech acts are reached. The Moore and Paris system contains approximately 150 such plan operators and is considered sufficiently stable for use in various application systems.

Particular text schemata are associated with specific communicative intentions (such as answering a specified user-question or constructing a specified text-type) directly. Rhetorical relations are included as the possible expansions of plan operators with communicative intentions as their effects. The intentions employed are typically defined by an application system or a research interest—for example, Suthers (1991) presents a useful set for generating pedagogically adequate explanations, others (McKeown, 1985; Reiter, Mellish, et al., 1992) adopt sets of possible responses to questions addressed to databases. The lack of clear definitions for what is to be accepted as an *intention* constitutes a substantial theoretical problem.

Whereas text schemata, which are now generally interpreted as pre-compiled plan sequences, and rhetorical structuring impose text structure on information, there are cases where it is argued that it is better for the information to be expressed to impose its structure more freely on text. Such *data-driven* approaches (cf. Hovy, 1988; Kittredge, Korelsky, et al., 1991; Suthers, 1991; Meteer, 1991; McDonald, 1992), allow an improved opportunistic response to the contingencies of particular generation situations. Data-driven critics can be combined with the top-down planning of rhetorical structures in order to improve structures according to *aggregation* rules (Hovy, 1993) or text heuristics (Scott & de Souza, 1990). A variation on data-driven content selection is offered by allowing transformation of the information itself, by means of logical inference rules defined over the knowledge base (e.g., Horacek, 1990).

Finally, a further active area of research is the addition of dynamic constraints on the construction of rhetorical structures. Two examples of such constraints are the use of *focus* (McCoy & Cheng, 1991) and the use of *thematic development* (Hovy, Lavid, et al., 1992) to direct selection among alternative rhetorical organizations.

4.3.2 Limitations

Although an increasing number of systems find the use of rhetorical relations, augmented in the ways described above, an effective means of planning text, unclarities in the definitions of rhetorical relations and weaknesses in their processing schemes result in some inherent limitations. These limitations are often hidden in specific contexts of use by hardwiring decisions and constraints that would in the general case need to be explicitly represented as linguistic resources and decisions. Success in the particular case should therefore always be re-considered in terms of the cost of re-use.

The selection of appropriate granularities for the presentation of information remains an unsolved problem. Information will be packaged into units depending on contingencies of that information's structure, on the text purpose, on the expected audience, on the writer's biases, etc. This general aggregation problem requires solutions that go beyond specific heuristics.

Also problematic is the assumption that a rhetorical structure can decompose a text down to the granularity of inputs required for surface generators. Current systems impose more or less *ad hoc* mappings from the smallest segments of the rhetorical structure to their realizations in clauses. Much fine-scaled text flexibility is thus sacrificed (cf. Meteer, 1991); this also reduces the multilingual effectiveness of such accounts.

Finally, algorithms for deep generation remain in a very early stage of development. It is clear that top-down planning is not sufficient. The interdependencies between many disparate kinds of information suggest the application of constraint-resolution techniques (Paris & Maier, 1991) (as shown in the example plan operator given above) , but this has not yet been carried out for substantial deep generation components. The kinds of inferences typically supported in deep generation components are also limited, and so more powerful inference techniques (e.g., abduction Lascarides & Oberlander, 1992; decompositional, causal-link planning Young, Moore, et al., 1994) may be appropriate.

4.3.3 Future Directions

Computational components responsible for deep generation are still most often shaped by their concrete contexts of use, rather than by established theoretical principles. The principal problem of deep generation is thus one of uncovering the nature of the necessary decisions underlying textual presentation and of organizing the space of such decisions appropriately. It is crucial that methodologies and theoretical principles be developed for this kind of linguistic description.

Furthermore, current work on more sophisticated inferencing capabilities need to be brought to bear on deep generation. Important here, however, is to ensure that this is done with respect to sufficiently complex sources of linguistic constraint. Approaches rooted in *mainstream* (computational) linguistics posit fewer linguistic constraints in favour of more powerful inferencing over common sense knowledge. Shieber (1993), for example, divides generation generally into the *generator* (i.e., surface generator: mapping semantics to syntax) and the *reasoner* (the rest: pragmatics), whereby inferences are allowed to blend into common sense reasoning. This leaves no theoretically well-specified space of linguistic decisions separate to general inferential capabilities. The consequences of this for generation are serious; it is essential that more structured sources of constraint are made available if generation is to succeed.

Very rich, but computationally underspecified, proposals in this area can be found in functional approaches to language and text (cf. Martin, 1992); results here suggest that the space of linguistic text organizational decisions is highly complex—similar to the kind of complexity found within grammars and lexicons. One methodology to improve the status of such accounts is then to use the control requirements of grammars and semantics as indications of the kinds of distinctions that are required at deeper, more abstract level of organization (cf. Matthiessen, 1987; Bateman, 1991; McDonald, 1993). The richer the grammatical and semantic starting points taken here, the more detailed hypotheses concerning those deeper levels become. This then offers an important augmentation of the informationally weak approaches from structural linguistics. Sophisticated inferential capabilities *combined with* strong sources of theoretically motivated linguistic constraints appear to offer the most promising research direction. This is also perhaps the only way to obtain an appropriate balance between fine detail and generality in the linguistic knowledge proposed. New work in this area includes that of the ESPRIT Basic Research Action DANDELION (EP6665).

A further key problem is the availability of appropriately organized knowledge representations. Although in research the generation system and the application system are sometimes combined, this cannot be assumed to be the case in general. The information selected for presentation will therefore be drawn from a representational level which may or may not have some linguistically relevant structuring, depending on the application or generation system architecture involved. This information must then be construed in terms that can be related to some appropriate linguistic expression and, as McDonald (1994) points out with respect to application systems providing only raw numerical data, this latter step can be a difficult one in its own right. More general techniques for relating knowledge and generation intentions can only be provided if knowledge representation is guided more by the requirements of natural language. It is difficult for a knowledge engineer to appreciate just how inadequate a domain model that is constructed independently of natural language considerations—although possibly highly elegant and inferentially-adequate for some application—typically reveals itself when natural language generation is required (cf. Novak, 1991). If text generation is required, it is necessary for this to be considered at the outset in the design of any knowledge-based system; otherwise expensive redesign or limited text generation capabilities will be unavoidable.

4.4 Chapter References

ANLP (1992). *Proceedings of the Third Conference on Applied Natural Language Processing*, Trento, Italy.

Appelt, D. E. (1985). *Planning English Sentences.* Cambridge University Press.

Appelt, D. E. (1987). Bidirectional grammars and the design of natural language generation systems. In Wilks, Y., editor, *Theoretical Issues in Natural Language Processing-3*, pages 185–191. Erlbaum, Hillsdale, New Jersey.

Bateman, J., Maier, E., Teich, E., and Wanner, L. (1991). Towards an architecture for situated text generation. In *Proceedings of the ICCICL*, Penang, Malaysia.

Bateman, J. A. (1991). Uncovering textual meanings: a case study involving systemic-functional resources for the generation of Japanese texts. In Paris, C. L., Swartout, W. R., and Mann, W. C., editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics.* Kluwer Academic.

Bateman, J. A. and Hovy, E. H. (1992). An overview of computational text generation. In Butler, C., editor, *Computers and Texts: An Applied Perspective*, pages 53–74. Basil Blackwell, Oxford, England.

Bateman, J. A., Moore, J. D., and Whitney, R. A. (1990). Upper modeling: A level of semantics for natural language processing. In IWNLG, editor, *Proceedings of the Fifth International Workshop on Natural Language Generation*, Pittsburgh, Pennsylvania. Springer-Verlag.

Block, H.-U. (1994). Compiling trace & unification grammar. In *Reversible Grammar in Natural Language Processing*, pages 155–174. Kluwer Academic Publishers.

Bresnan, J., editor (1982). *The Mental Representation of Grammatical Relations.* MIT Press, Cambridge, Massachusetts.

Busemann, S. (1990). *Generierung natürlicher Sprache mit Generalisierten Phrasenstruktur–Grammatiken.* PhD thesis, University of Saarland (Saarbrücken).

Calder, J., Reape, M., and Zeevat, H. (1989). An algorithm for generation in unification categorial grammar. In *Proceedings of the Fourth Conference of the European Chapter of the Association for Computational Linguistics*, pages 233–240, Manchester. European Chapter of the Association for Computational Linguistics.

Cawsey, A. (1989). *Generating Explanatory Discourse: A Plan-Based, Interactive Approach.* PhD thesis, University of Edinburgh.

COLING (1988). *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest.

Dale, R. (1990). Generating receipes: An overview of epicure. In Dale, R., Mellish, C. S., and Zock, M., editors, *Current Research in Natural Language Generation*, pages 229–255. Academic Press, London.

Dale, R., Hovy, E. H., Rösner, D., and Stock, O., editors (1992). *Aspects of Automated Natural Language Generation.* Number 587 in Lecture Notes in AI. Springer-Verlag, Heidelberg.

Dale, R., Mellish, C. S., and Zock, M., editors (1990). *Current Research in Natural Language Generation.* Academic Press, London.

Den, Y. (1994). Generalized chart algorithm: An efficient procedure for cost-based abduction. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, New Mexico. Association for Computational Linguistics.

DeSmedt, K. and Kempen, G. (1987). Incremental sentence production, self–correction and coordination. In Kempen, G., editor, *Natural Language Generation*, pages 365–376. Martinus Nijhoff, Dordrecht.

Dymetman, M. and Isabelle, P. (1988). Reversible logic grammars for machine translation. In *Proceedings of the Second International Conference on Theoretical and Methodological issues in Machine Translation of Natural Languages*, Pittsburgh, Pennsylvania.

Dymetman, M., Isabelle, P., and Perrault, F. (1990). A symmetrical approach to parsing and generation. In Karlgren, H., editor, *Proceedings of the 13th International Conference on Computational Linguistics*, pages 90–96, Helsinki. ACL.

Elhadad, M. (1992). *Using Argumentation to Control Lexical Choice: A Functional Unification-Based Approach.* PhD thesis, Computer Science Department, Columbia University.

Elhadad, M. and Robin, J. (1992). Controlling content realization with functional unification grammars. In Dale, R., Hovy, E. H., Rösner, D., and Syock, O., editors, *Aspects of Automated Natural Language Generation*, pages 89–104. Springer, Heidelberg.

Fawcett, R. P. (1992). The state of the craft in computational linguistics: A generationist's viewpoint. Technical Report COMMUNAL Working Papers No. 2, Cardiff Computational Linguistics Unit, University of Wales.

Feiner, S. and McKeown, K. R. (1990). Coordinating text and graphics in explanation generation. In *Proceedings of the AAAI-90*, pages 442–449, Boston. American Association for Artificial Intelligence.

Gardent, C. and Plainfossé, A. (1990). Generating from a deep structure. In Karlgren, H., editor, *Proceedings of the 13th International Conference on Computational Linguistics*, pages 127–132, Helsinki. ACL.

Gazdar, G., Klein, E., Pullum, G., and Sag, I. (1985). *Generalized Phrase Structure Grammar*. Blackwell.

Gerdemann, D. and Hinrichs, E. W. (1990). Functor-driven generation with categorial-unification grammars. In Karlgren, H., editor, *Proceedings of the 13th International Conference on Computational Linguistics*, pages 145–150, Helsinki. ACL.

Gerdemann, D. D. (1991). *Parsing and Generation of Unification Grammars*. PhD thesis, University of Illinois. Also Cognitive Science Technical Report CS-91-06.

Goldman, N. (1975). Conceptual generation. In Schank, R., editor, *Conceptual Information Processing*. North-Holland, Amsterdam.

Horacek, H. (1990). The architecture of a generation component in a complete natural language dialogue system. In Dale, R., Mellish, C. S., and Zock, M., editors, *Current Research in Natural Language Generation*, pages 193–227. Academic Press, London.

Hovy, E., Lavid, J., Maier, E., Mittal, V., and Paris, C. (1992). Employing knowledge resources in a new text planner architecture. In *Aspects of automated natural language generation*, pages 57–72. Springer-Verlag, Berlin.

Hovy, E. H. (1988). *Generating Natural Language under Pragmatic Constraints*. Lawrence Erlbaum, Hillsdale, New Jersey.

Hovy, E. H. (1988). Planning coherent multisentential text. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*, SUNY, Buffalo, New York. Association for Computational Linguistics.

Hovy, E. H. (1993). Automated discourse generation using discourse relations. *Artificial Intelligence*, 63:341–385.

Hovy, E. H. and Arens, Y. (1991). Automatic generation of formatted text. In *Proceedings of the 8th AAAI Conference*, Anaheim, California. American Association for Artificial Intelligence.

Hovy, E. H. and Knight, K. (1993). Motivation for shared ontologies: An example from the Pangloss collaboration. In *Proceedings of the Workshop on Knowledge Sharing and Information Interchange*, Chambery, France.

Jacobs, P. S. (1988). Achieving bidirectionality. In *Proceedings of the 12th International Conference on Computational Linguistics*, pages 267–274, Budapest.

Jameson, A. (1987). How to appear to be conforming to the 'maxims' even if you prefer to violate them. In Kempen, G., editor, *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics*, pages 19–42. Kluwer Academic, Boston, Dordrecht.

Jameson, A. and Wahlster, W. (1982). User modelling in anaphora generation: Ellipsis and definite description. In *Proceedings of the 1982 European Conference on Artificial Intelligence*, pages 222–227, Orsay, France.

Karlgren, H., editor (1990). *Proceedings of the 13th International Conference on Computational Linguistics*, Helsinki. ACL.

Kempen, G., editor (1987). *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics.* Kluwer Academic, Boston, Dordrecht.

Kittredge, R., Korelsky, T., and Rambow, O. (1991). On the need for domain communication knowledge. *Computational Intelligence*, 7(4):305–314.

Kukich, K. (1983). *Knowledge-Based Report Generation: A Knowledge-Engineering Approach.* PhD thesis, University of Pittsburgh.

Lascarides, A. and Oberlander, J. (1992). Abducing temporal discourse. In *Proceedings of the Sixth International Workshop on Natural Language Generation*, pages 167–182, Trento, Italy. Springer-Verlag. Also in Dale, Hovy, et al. (1992).

Mann, W. C., Bates, M., Grosz, B. J., McDonald, D. D., McKeown, K. R., and Swartout, W. R. (1981). Text generation: The state of the art and the literature. Technical Report RR-81-101, USC/Information Sciences Institute.

Mann, W. C. and Matthiessen, C. M. I. M. (1985). Nigel: A systemic grammar for text generation. In Benson, R. and Greaves, J., editors, *Systemic Perspectives on Discourse: Selected Papers from the Ninth International Systemics Workshop.* Ablex, London.

Mann, W. C. and Thompson, S. A. (1987). Rhetorical structure theory: description and construction of text structures. In Kempen, G., editor, *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics*, pages 85–96. Kluwer Academic, Boston, Dordrecht.

Martin, J. R. (1992). *English text: systems and structure.* Benjamins, Amsterdam.

Matthiessen, C. M. I. M. (1983). Systemic grammar in computation: The Nigel case. In *Proceedings of the First Conference of the European Chapter of the Association for Computational Linguistics*, Pisa, Italy. European Chapter of the Association for Computational Linguistics.

Matthiessen, C. M. I. M. (1987). Notes on the organization of the environment of a text generation grammar. In Kempen, G., editor, *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics.* Kluwer Academic, Boston, Dordrecht.

McCoy, K. F. (1986). The ROMPER system: Responding to object-related misconceptions using perspective. In *Proceedings of the 24th Annual Meeting of the Association for Computational Linguistics*, Columbia University, New York. Association for Computational Linguistics.

McCoy, K. F. and Cheng, J. (1991). Focus of attention: constraining what can be said next. In Paris, C. L., Swartout, W. R., and Mann, W. C., editors, *Natural Language Generation in Artificial Intelligence and Computational Linguistics.* Kluwer Academic.

McDonald, D. D. (1980). *Natural Language Production as a Process of Decision Making Under Constraint.* PhD thesis, Department of Computer Science and Electrical Engineering, Massachusetts Institute of Technology.

McDonald, D. D. (1992). Type-driven suppression of redundancy in the generation of inference-rich reports. In *Aspects of automated natural language generation*, pages 72–88. Springer-Verlag, Berlin.

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

McDonald, D. D. (1994). Reversible NLP by linking the grammar to the knowledge base. In *Reversible Grammar in Natural Language Processing*, pages 257–292. Kluwer Academic Publishers.

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

McKeown, K. R., Robin, J., and Tanenblatt, M. (1993). Tailoring lexical choice to the user's vocabulary in multimedia explanation generation. In *Proceedings of the 31st*

Annual Meeting of the Association for Computational Linguistics, pages 226–234, Ohio State University. Association for Computational Linguistics.

McKeown, K. R. and Swartout, W. R. (1987). Language generation and explanation. *Annual Reviews of Computer Science*, 2:401–449.

Meteer, M. (1990). *The "Generation Gap": The Problem of Expressibility in Text Planning*. PhD thesis, University of Massachusetts at Amherst.

Meteer, M., McDonald, D. D., Anderson, S., Foster, D., Gay, L., Huettner, A., and Sibun, P. (1987). Mumble-86: Design and implementation. Technical Report COINS-87-87, University of Massachusetts at Amherst.

Meteer, M. M. and Shaked, V. (1988). Strategies for effective paraphrasing. In *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest.

Meteer, M. W. (1991). Bridging the generation gap between text planning and linguistic realization. *Computational Intelligence*, 7(4):296–304.

Miller, G. A. (1985). Wordnet: A dictionary browser. In *Information in Data: Proceedings of the 1st Conference of the UW Centre for the New Oxford Dictionary.* University of Waterloo, Canada.

Moore, J. D. (1989). *A Reactive Approach to Explanation in Expert and Advice-Giving Systems*. PhD thesis, University of California at Los Angeles.

Moore, J. D. and Paris, C. L. (1993). Planning texts for advisory dialogues: Capturing intentional and rhetorical information. *Computational Linguistics*, 19(4):651–694.

Neumann, G. (1994). *A Uniform Computational Model for Natural Language Parsing and Generation*. PhD thesis, Universität des Saarlandes, Germany.

Neumann, G. and van Noord, G. (1994). Reversibility and self-monitoring in natural language generation. In *Reversible Grammar in Natural Language Processing*, pages 59–96. Kluwer Academic Publishers.

Novak, H.-J. (1991). Integrating a generation component into a natural language understanding system. In Herzog, O. and Rollinger, C.-R., editors, *Text understanding in* LILOG*: integrating computational linguistics and artificial intelligence, Final report on the IBM Germany* LILOG*-Project*, pages 659–669. Springer-Verlag, Berlin, Heidelberg, New York. Lecture notes in artificial intelligence, 546.

Paris, C. L. (1993a). *The Use of Explicit Models in Text Generation.* Francis Pinter, London.

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

Paris, C. L. and Maier, E. A. (1991). Knowledge sources of decisions? In *Proceedings of the IJCAI-91 Workshop on Decision Making Throughout the Generation Process*, pages 11–17, Sydney, Australia.

Paris, C. L., Swartout, W. R., and Mann, W. C., editors (1990). *Natural Language Generation in Artificial Intelligence and Computational Linguistics.* Kluwer Academic, Boston.

Paris, C. L., Swartout, W. R., and Mann, W. C., editors (1991). *Natural Language Generation in Artificial Intelligence and Computational Linguistics.* Kluwer Academic.

Pollard, C. and Sag, I. A. (1987). *An Information-Based Approach to Syntax and Semantics: Fundamentals.* Number 13 in Center for the Study of Language and Information (CSLI) Lecture Notes. Stanford University Press and Chicago University Press.

Rambox, O. and Korelsky, T. (1992). Applied text generation. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 40–47, Trento, Italy.

Reiter, E., Mellish, C. S., and Levine, J. (1992). Automatic generation of on-line documentation in the IDAS project. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 64–71, Trento, Italy.

Reithinger, N. (1991). *Eine Parallele Architektur zur Inkrementellen Generierung Multimodaler Dialogbeiträge.* PhD thesis, University of the Saarland.

Rösner, D. (1986). *Ein System zur Generierung von Deutschen Texten aus Semantischen Repräsentationen.* PhD thesis, University of Stuttgart.

Scott, D. and de Souza, C. S. (1990). Getting the message across in RST-based generation. In Dale, R., Mellish, C. S., and Zock, M., editors, *Current Research in Natural Language Generation*, pages 47–73. Academic Press, London.

Shieber, S. M. (1988). A uniform architecture for parsing and generation. In *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest.

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

Shieber, S. M., Pereira, F. C. N., van Noord, G., and Moore, R. C. (1990). Semantic-head-driven generation. *Computational Linguistics*, 16:30–42.

Strzalkowski, T. (1989). Automated inversion of a unification parser into a unification generator. Technical Report 465, Courant Institute of Mathematical Sciences, New York University.

Strzalkowski, T. (1994). *Reversible Grammar in Natural Language Processing.* Kluwer Academic Publishers.

Strzalkowski, T., Carballo, J. P., and Marinescu, M. (1995). Natural language information retrieval: TREC-3 report. In *National Institute of Standards and Technology Special Publication on the The Third Text REtrieval Conference (TREC-3)*, Washington, DC. National Institute of Standards and Technology, U.S. Department of Commerce, U.S. Government Printing Office.

Suthers, D. (1993). *An Analysis of Explanation and its Implications for the Design of Explanation Planners.* PhD thesis, University of Massachusetts at Amherst.

Suthers, D. D. (1991). Task-appropriate hybrid architectures for explanation. *Computational Intelligence*, 7(4):315–333.

Uszkoreit, H. (1986). Categorial unification grammars. In *Proceedings of the 11th International Conference on Computational Linguistics*, Bonn. ACL.

VanNoord, G. (1990). An overview of head-driven bottom-up generation. In Dale, R., Mellish, C. S., and Zock, M., editors, *Current Research in Natural Language Generation*, pages 141–165. Academic Press, London.

VanNoord, G. J. M. (1993). *Reversibility in Natural Language Processing.* PhD thesis, University of Utrecht, The Netherlands.

Von Hahn, W., Höppner, W., Jameson, A., and Wahlster, W. (1980). The anatomy of the natural language dialogue system HAM-RPM. In Bolc, L., editor, *Natural Language Based Computer Systems.* McMillan, Münich.

Wahlster, W., André, E., Graf, W., and Rist, T. (1991). Designing illustrated texts: How language production is influenced by graphics production. In *Proceedings of the Fifth Conference of the European Chapter of the Association for Computational Linguistics*, pages 8–14, Berlin. European Chapter of the Association for Computational Linguistics.

Wedekind, J. (1988). Generation as structure driven derivation. In *Proceedings of the 12th International Conference on Computational Linguistics*, Budapest.

Young, R. M., Moore, J. D., and Pollack, M. E. (1994). Towards a principled representation of discourse plans. In *Proceedings of the Sixteenth Conference of the Cognitive Science Society*, Atlanta.

Zeevat, H., Klein, E., and Calder, J. (1987). An introduction to unification categorial grammar. In Haddock, J. N., Klein, E., and Morrill, G., editors, *Edinburgh Working Papers in Cognitive Science, volume 1: Categorial Grammar, Unification Grammar, and Parsing*, volume 1 of *Working Papers in Cognitive Science*. Centre for Cognitive Science, University of Edinburgh.