Meaning Representation and Text Planning.

Christine DEFRISE
IRIDIA
Université Libre de Bruxelles

Sergeï NIRENBURG
Center for Machine Translation
Carnegie Mellon University

The data flow in natural language generation (NLG) starts with a 'world' state, represented by structures of an application program (e.g., an expert system) that has text generation needs and an impetus to produce a natural language text. The output of generation is a natural language text. The generation process involves the tasks of a) delimiting the content of the eventual text, b) planning its structure, c) selecting lexical, syntactic and word order means of realizing this structure and d) actually realizing the textusing the latter. In advanced generation systems these processes are treated not in a monolithic way, but rather as components of a large, modular generator. NLG researchers experiment with various ways of delimiting the modules of the generation process and control architectures to drive these modules (see, for instance, McKeown, 1985, Hovy, 1987 or Meteer, 1989). But regardless of the decisions about general (intermodular) or local (intramodular) control flow, knowledge structures have to be defined to support processing and facilitate communication among the modules.

The natural language generator DIOGENES(e.g., Nirenburg et al., 1989) has been originally designed for use in machine translation. This means that the content delimitation stage is unnecessary, as the set of meanings to be realized by the generator is obtained in machine translationas a result of source text analysis. The first processing component in DIOGENES is, therefore, its text planner which, takes as input a *text meaning representation* (TMR) and a set of static pragmatic factors (similar to Hovy's (1987) rhetorical goals) and produces a *text plan* (TP), a structure containing information about the order and boundaries of target language sentences; the decisions about reference realization and lexical selection.[1] At the next stage, a set of semantics-to-syntax *mapping rules* are used to produce a set of target-language syntactic structures (we are using the f-structures of LFG — see, e.g., Nirenburg and Levin, 1989). Finally, a syntactic realizer produces a target language text from the set of f-structures.

To produce texts of adequate quality, natural language generation needs a sufficiently expressive input language. In this paper we discuss several important aspects of the knowledge and the processing at the text planning stage of a generation system. First, we describe a comprehensive language processing paradigm which underlies work on both generation and analysis of natural language in our environment. Next, we illustrate the features of our meaning representation languages, the text meaning representation language TAMERLAN and the text plan representation language TPL. Finally, we describe the mechanism of text planning in DIOGENES and illustrate the formalism and the strategy for acquiring text planning rules.

1 Text Meaning in Analysis & Generation.

Understanding a text involves determining its propositional content as well as detecting pragmatic meaning elements, those pertaining to the speech situation and to the nature of the discourse producer's goals.[2] A high-quality natural language generator must be capable of expressing the entire set of meaning types in a natural language. The input to a generator must, therefore, express all of the kinds of meanings to be rendered in the output text. This requires a convenient knowledge representation scheme for the input, as well as actual knowledge (in the form of heuristic selection rules) about interactions between the input and elements of the output.

The acquisition of this knowledge must be informed by systematic field work on such linguistic phenomena as focus, topic and comment, speech acts, speaker attitudes, or reference-related phenomena (anaphora, deixis, ellipsis, definite description, etc.). When these phenomena are studied in computational linguistics, they are usually approached from the standpoint of language *understanding*. The types of activities in *generation* often differ from those in analysis (see, e.g., Nirenburg and Raskin, 1987). To highlight the requirements of natural language generation, we will describe a generation-oriented approach to speech acts and attitudinal knowledge.

1.1 Speech Acts.

We model a text producer/consumer as having, in addition to knowledge about the grammar and lexis of a natural language and knowledge about the world, an inventory of world-related and communication-related goals and plans that are known to lead to the achievement of these goals. Producing text, then, is understood as a planning process to achieve a goal. Being able to achieve goals through communication can be seen as broadening the range of tools an agent has for attaining its goals. It is important to distinguish clearly between the producer's goal and plan inventory and a set of active goal and plan instances which constitute the text producer's *agenda*. When the decision is made to try to achieve a goal by rhetorical means, one of the available rhetorical plans corresponding to this goal is selected for processing. Based on knowledge recorded in this plan structure, the agent produces a new, language-dependent, structure, a *text plan*, as an intermediate step in producing a text.

The process of natural language *understanding* in our approach will be modeled as a plan recognition process in a similar conceptual architecture. In understanding, it is necessary to determine not only the propositional content but also the illocutionary force of each utterance. This involves reconstructing (on the basis of the propositional content and the knowledge of the speech situation) the text

[1] Text planning in DIOGENES is described in detail Defrise and Nirenburg, 1989; the language for writing planning rules, in Nirenburg et al., in preparation; the lexical selection in DIOGENES is described, e.g., in Nirenburg and Nirenburg, 1988.

[2] We decided to use the term 'producer' for the author of a text and the speaker in a dialog; and the term 'consumer' to indicate the reader of a text or a hearer in a dialog.

producer's goals, in order to decide if an utterance constitutes a direct or an indirect speech act, calculating any added non-propositional, implicit elements of meaning.

In generation, as we have seen, the task is to decide, on the basis of the communication situation and knowledge of the producer's intentions, whether to achieve a given goal via rhetorical means. If so, it is also necessary to decide whether it will be realized in the target text directly or indirectly. If a direct speech act is chosen, one must further determine whether to lexicalize the speech act through the use of a performative verb (e.g., *promise*, *order*, etc.).

The treatment of speech acts is thus always accomplished using links between the set of producer's current goals and the propositional content. The order of operations, however, differs depending on whether this treatment is a part of analysis or generation. To illustrate, suppose an agent is cold and has a goal to be warm. To achieve this goal, we construct a plan tree, with the goal *be-warm* as the root, and various plans of action as subtrees (see Figure 1).

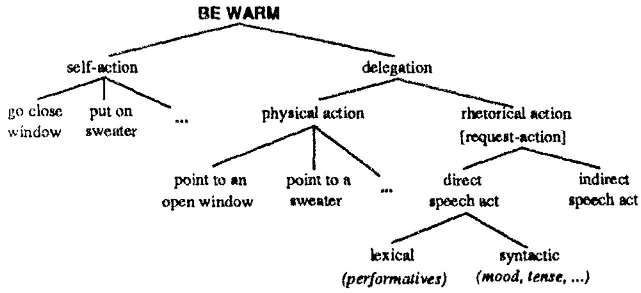

Figure 1: **The Plan Tree for the Goal 'Be-Warm.'**

In analysis, the tree is traversed bottom-up: the consumer has direct access to the input utterance and uses it as a clue to reconstruct the producer's goals. If complete extraction of meaning is to be achieved, it is crucial to know whether an utterance realizes a direct or an indirect speech act, In generation, the tree will be traversed top-down. If the producer's choice is to achieve his goal through rhetorical means, he can generate any of the following three utterances:

(1) I order you to close the window.
(2) Close the window, please.
(3) It's cold in here.

Now, even though (3) differs in propositional meaning from (1) and (2), the distinction between direct speech act (1) or (2) and indirect speech act (3) does not matter from the point of view of the producer's goals. In TAMERLAN the representation of both *Close the window* and *It is cold in here* will contain a pointer (in TAMERLAN the filler of the *producer-intention* slot of every *clause* frame) to the same plan node, in this case, 'request-action.' The fact that in the above utterances, the request-action is realized as a command or a statement pertains to text, not domain planning.

1.2 Producer Attitude.

Our reasons for introducing attitudes as an explicit part of the representation of the meaning of a natural language clause are manifold. In what follows we will review three (partially interconnected) reasons. Representing attitudes helps to a) support reasoning about producer goals; b) highlight the argumentative structure of a discourse; c) provide a convenient vehicle for representing modal meanings.

Almost all spoken and written discourse involves the participants' opinions, so much so that producing a perfectly objective text is an almost impossible task. Within the set of possible goals relating to generating text, the introduction (explicit or implicit, lexicalized or not) of the producer's opinions and points of view serves two goals:

- modifying the consumer's model of the producer by stating facts (including opinions) about the latter which are not in principle observable by the consumer
- modifying the consumer's opinions by stating producer's opinions about facts of the world (the latter can in principle be observed by the consumer)

The above distinctions only become visible if one decides to represent attitudes overtly. Once this decision is made, it becomes clear that it brings about better description possibilities for additional linguistic phenomena, such as the argumentative structure of discourse. It has been observed (e.g., Anscombre and Ducrot, 1983) that texts have a well-defined *argumentative structure* which a) reflects the producer's current goals and b) influences such processes as the ordering of text components and lexical selection in generation. The argumentative structure of a text is realized (or, in text understanding, detected) through linguistic means such as the use of scalar adverbs ('only', 'even', 'almost', etc.), connectives ('but', 'since'), adjectives ('unbearable', 'fascinating', etc.). Sets of such lexical items may have to be considered equivalent from a purely semantic point of view, but different in a facet of their pragmatic effect known as *argumentative orientation.* For example, to illustrate the interplay between semantic content and argumentative orientation (i.e. the producer's attitude towards an event), compare (4) and (5), which have opposite truth conditions, but the same pragmatic value — from both (4) and (5) the consumer will infer that the producer regards Burma as an inefficient sleuth. In this example it is sufficient to retain pragmatic information concerning the producer's judgment of Burma while the semantic differences (induced by the use of 'few' versus 'none at all') can be disregarded. However, in other contexts the semantics will matter much more — consider, for instance, (6) for which there can be no paraphrase with 'no clues at all.'

(4) Nestor Burma found few clues. Nobody was surprised.
(5) Nestor Burma found no clues at all. Nobody was surprised.
(6) Nestor Burma found few clues. But it was still better than having none at all.

The difference between (7) and (8), whose truth conditions are similar, is purely argumentative (or attidudinal) — (7) expresses a positive (optimistic!) attitude, (8) the opposite point of view. This example shows how crucial the extraction of the argumentative structure is, since it is the only clue for the (in)acceptability of (9).

(7) Nestor has a little money.
(8) Nestor has little money.
(9) *Nestor has little money. He wouldn't mind spending some on chocolate.

Finally, we use the attitude markers as a means of expressing modality. Traditionally, formal semanticists have extended first order logic to modal logic in order to account for modals. This places the modals at a purely semantic level, as a result of which the it is difficult to distinguish between what is observable for both producer and consumer and what is not (opinions, beliefs, etc.). We consider that expressions like 'perhaps,' 'possibly,' 'it is almost certain that' are clues as to what the producer's beliefs and attitudes are towards facts of the world and help the consumer modify or update his model of the producer. It is for the above reasons that we decided to include a detailed specification of producer attitudes into the input specification for generation.

2 The Structure of TAMERLAN.

TAMERLAN is a frame-based representation language that has the following basic entity types: *text, clause, relation, proposition, attitude* and *pointer* (to the producer's current plan). A text frame indexes a set of clauses and a set of relations comprising an input text. TAMERLAN clauses delimit the propositional and pragmatic content of target language utterances. Relations represent links among events, objects, or textual objects. In what follows, we will illustrate the structure of TAMERLAN concentrating on the representation of attitudes and agenda pointers (corresponding to speech acts). An exhaustive definition and description of TAMERLAN is given in Defrise and Nirenburg (in preparation).

2.1 Representation of Attitudes.

Each TAMERLAN clause contains one or several attitude slots. Each is composed of four facets: the type facet, the value facet, the scope facet, and the 'attributed-to' facet.

The possible **types** of attitudes are :

- *epistemic* (with values taken from the {0,1} interval to account for expressions like *perhaps*; the endpoints of the interval intuitively correspond to the values of *impossible* and *necessary*);
- *evaluative* (with values taken from a similar scale, with the endpoints interpreted as, roughly, 'the best' 'the worst,' the midpoint as 'neutral,' and other intermediate points used to account for expressions like 'fairly interesting');
- *deontic* (ranging from 'unfair' to 'fair');
- *expectation* (ranging from 'expected' to 'surprise').

The organization of the above types is similar — their value ranges are all one type of scale. The differences among them are semantic. The above classification is an enhancement of Reichman's treatment of "context spaces" (1985: 56). We use the terminology (if not exactly the spirit) of her distinction among the epistemic, evaluative and deontic *issue-type context spaces*. Context space is Reichman's term for a discourse segment. The issue context space roughly corresponds to our attitude component, while the non-issue context space provides a shallow taxonomy for discourse segment types (Reichman lists comment, narrative support, and nonnarrative support as the non-issue type values).

Every attitude type has a **scale of values** associated with it. The *value* component specifies a region on the appropriate scale. Though the semantics of all scales is different, the set of values is the same — we represent all attitude scales as {0,1} intervals. Thus, the value (> 0.8) on the scale of *evaluative-saliency* will be lexically realizable through the modifier *important(ly)*, while the same value on the *deontic* scale will end up being realized as *should* or *ought to*.

The **attributed-to** component of the attitude simply binds the attitude to a particular cognitive agent (which may be the producer of the utterance or some other known or unknown agent), who is responsible for the content of the utterance. This is important for understanding reported speech, and more generally the polyphony phenomena, in the sense of Ducrot (1984). Ducrot's theory of polyphony, an approach to extended reported speech treatment, provides a framework for dealing with the interpretation of a number of semantic and pragmatic phenomena, e.g., the difference in meaning and use between 'since' and 'because,' certain particularities of negative sentences, etc.

The **scope** of the attitude representation pinpoints the entity to which this attitude is expressed. The values of the scope can be the whole proposition, a part of it or another attitude value, with its scope. In understanding the text the consumer notes the attitudes of the producer to the content. The attitudes can be expressed toward events, objects, properties or other attitudes (see 10 — 13, respectively).

(10) The train, unfortunately, left at 5 p.m.
(11) This book is interesting.
(12) The meeting was reprehensibly short.
(13) Unfortunately, I ought to leave.

McKeown and Elhadad (1989) also treat argumentative scales and attitudinals in a generation environment. They, however, consider these phenomena as part of syntax, thus avoiding the need to add a special pragmatic component to their system. This decision is appropriate from the point of view of minimizing the changes in an existing generator due to the inclusion of attitude information. However, if compatibility is an overriding concern, introducing a separate component is a more appropriate choice.

2.2 Representation of Producer's Goals.

We overtly refer to the producer's goals in each TAMERLAN clause, using the 'producer-intention' slot. The filler of this slot is a *pointer* to an action or a plan in the producer agenda. The producer agenda, which is a necessary background component of text planning, contains representations of active goal and plan instances. Since we are interested in discourse situations, we take into account only the goals and plans that a) presuppose the situation with at least two cognitive agents and b) relate to rhetorical (and not physical-action) realizations of goals.

To illustrate our use of the 'producer-intention' slot in text generation, consider the task of generating from a TMR which we will gloss as 'The speaker promises to return to his current location at 10 o'clock.' Depending on the context and other parameters, the producer may decide to generate *I will return at 10* or *I promise to return at 10*. In the latter case the decision is made to realize the meaning of the speech act lexically. The mechanism for this is as follows: traversing the producer-intention slot the producer gets to the relevant point in the agenda, which is the (primitive) plan PROMISE.[3] Since the realization

[3] or THREAT, etc., as the case may be

rules for speech acts prescribe their realization as first-person-singular clauses with the lexical realizations of the names of appropriate speech plans (acts), the natural language clause *I promise X* is produced, and, eventually, X is expanded into the subordinate natural language clause *to return at 10*. The central point is that the former natural language clause is the realization of a pointer in the input, not an entire text representation clause.[4]

3 The Text Plan Language (TPL).

The text plan is an intermediate data structure which is the obtained through the application of the text planning rules to TMRs. TPs serve as input to the semantics-to-syntax mapping rules that produce f-structures which in turn serve as input to the syntactic realization module. While TMR does not specify the boundaries and order of sentences in the output text, the ways of treating co-reference or the selection of the appropriate lexical units (both open-class items and such closed-class ones as realizations of producer attitudes and discourse cohesion markers, connectives) for the target text, TP contains this information. Additionally, the text planning stage also produces values for such features as definiteness, tense and mood.

A text plan is an hierarchically organized set of frames. The root of this hierarchy is the *text structure* frame which, notably, contains the slot "has-as-part" whose value is an ordered set of *plan sentence* frames, S_i. A plan sentence frame contains a slot "subtype" whose values include "simple," "complex," and "and." Another slot contains an ordered list of *plan clause frames* comprising the sentence. A plan clause frame contains the realization, through lexical choice and feature values, of both propositional and pragmatic meanings in the input. It lists the realizations of the head of the proposition and pointers to *plan-role* frames that contain realizations of the case roles of this proposition. (Sometimes the filler of a case role slot in a plan clause will be a pointer to another plan clause.) Realizations of producer attitudes pertaining to the clause and of cohesion markers which realize some of TAMERLAN relations are also included, as is an indication of whether a given clause is a subordinate clause or a matrix clause in a complex sentence. The plan role frames are composed in a similar fashion.

4 The Mechanism of Text Planning.

Text planning can be understood as a mapping problem between sets of expressions in TAMERLAN and TPL. The mapping is achieved through the application of heuristic rules of the situation-action kind. The rules take as input elements of the input representation and mapped them into elements of a text plan based the various meaning components in the input and their combinations. In addition, the heuristic rules take into account the state of affairs in the communication situation, modelled in our system as a static set of pragmatic factors that determine the stylistic slant of a text. Our pragmatic factors are a subset of the rhetorical goals suggested by Hovy (1987) and include such factors as simplicity, formality, colorfulness, etc. Finally, the heuristic rules can take into account elements of the text plan under construction — those elements that were produced by previously triggered rules.

4.1 Control.

In our generator, we adopt a blackboard model of control. This means that the work in the system is performed by a set of semi-autonomous knowledge sources which apply various heuristic rules or run specialized algorithms in a distributed fashion. There is no centralized sequential model of control. The activated knowledge source instances are collected in the processing agenda(s). After a knowledge source instantiation "fires," its results are listed on one of several public data structures, *blackboards*, supported by our system. The balckboards are public in the sense that subsequent knowledge source instantiations can draw knowledge needed for their application from the blackboard spaces in which outputs of other knwolcdge source instances is recorded. Elements of TMR and TP are recorded on one of the blackboards, as are various intermediate processing results, such as those produced in the process of lexical selection.

Efficiency of a computational architecture can be controlled and improved by introducing special control knowledge sources which perform manipulations of the contents of the processor agendas, including such operations as obviation and retraction. See Nirenburg, Nyberg and Defrise, 1989 for a sketch of the text plan controller in our generation system.

5 From Meaning Representations to Plans.

In this section we show how decisions concerning the treatment of agenda pointers are reflected in the text plan and how decisions about realization of attitudinals are made. We will illustrate the treatment of attitudinals by showing the TMRs and their corresponding TPs for a set of examples. The treatment of attitudinals will be illustrated by listing the attitude-related text planning rules (a much more comprehensive list of text planning rules is included in Defrise and Nirenburg, 1989). In this fashion we will be able to illustrate both knowledge representation languages and the planning rule language.

5.1 Planning the Realization of Agenda Pointers.

Consider (14) — (16) as the desired target for generation.

(14) I will definitely be back by 10.
(15) I will be back by 10.
(16) I promise to be back by 10.

The corresponding set of (somewhat simplified) TAMERLAN structures is in (17) and (18), since the structures for (15) and (16) will be identical — indeed, the sentences are synonymous and differ only in what is known as *register* characteristics, in this case, the level of formality. Taking the above sets as input, the text planner will produce the text plans in (19) — (21). The differences between the realizations of the input (18) as either (15) or (16) will be reflected in the text plans, see (20) and (21).

(17)

```
(make-frame text
  (clauses (value clausel))
  (relations
       (value relationl)))
(make-frame clausel
```

[4] In natural language understanding, one will expect to obtain as input some full-fledged natural language clauses whose real meaning is purely pragmatic. Consider, for instance, the sentence "I hereby inform you that X." The meaning of "I hereby inform you" will represented as a filler of the "producer-intention" slot or as an attitude.

```
  (proposition (value #return1))
  (attitude (value attitude1))
  (producer-intention (value #statement1)))
(make-frame #return1
  (is-token-of (value *return))
  (phase (value begin))
  (iteration (value 1))
  (duration (value 1))
  (time (value (< 10))
  (agent (value *producer*))
  (destination (value #statement1.space)))
(make-frame attitude1
  (epistemic (value 1)
             (scope #return1)
             (attributed-to *producer*)))
(make-frame #statement1
  (is-token-of (value *statement))
  (time (value #statement1.time))
  (space (value #statement1.space)))
(make-frame relation1
  (type (value intention-before))
  (arguments
       (first (value  #statement1.time))
       (second (value #return1.time))))
```

(18)

```
(make-frame text
  (clauses (value clause1))
  (relations
      (value relation1)))
(make-frame clause1
  (proposition (value #return1))
  (producer-intention (value #promise1)))
(make-frame #return1
  (is-token-of (value *return))
  (phase (value begin))
  (iteration (value 1))
  (duration (value 1))
  (time (value (< 10))
  (agent (value *producer*))
  (destination (value #statement1.space)))
(make-frame #promise1
  (is-token-of (value *promise))
  (time (value #promise1.time))
  (space (value #promise1.space)))
(make-frame relation1
  (type (value intention-before))
  (arguments
       (first (value  #promise1.time))
       (second (value #return1.time))))
```

(19)

```
(TSF
 (has-as-part S1))
(S1
 (type simple)
 (clauses C1))
(C1
  (head be_back)
  (time
    (head 10)
    (features
       (time-relation before)))
  (modifier
      (head definitely))
  (features
      (tense future)
      (mood declarative))
  (agent r1)
  (destination r2))
(r1    (head PRO)
      (features
          (number singular)
          (person first)))
(r2    (head *ellided*))
```

(20)

```
(TSF
 (has-as-part S1))
(S1
 (type simple)
 (clauses C1))
(C1
  (head be_back)
  (features
      (tense future)
      (mood declarative))
  (time
    (head 10)
    (features
       (time-relation before)))
  (agent r1)
  (destination r2))
(r1    (head PRO)
      (features
          (number singular)
          (person first)))
(r2    (head *ellided*))
```

(21)

```
(TSF
 (has-as-part S1))
(S1
 (type complex)
 (clauses (C1 C2)))
(C1
 (head promise1)
  (features
      (tense present)
      (mood declarative))
  (agent r1)
  (theme C2))
(r1  (head PRO)
    (features
       (number singular)
       (person first)))
(C2
  (head be_back)
  (time
    (head 10)
    (features
       (time-relation before)))
  (features
      (tense future)
      (mood declarative))
  (agent r2)
  (destination r3))
(r2    (head PRO)
      (features
          (number singular)
          (person first)))
(r3    (head *ellided*))
```

5.2 Text Planning Rules for Attitudes.

Text planning rules in DIOGENES deal with a variety of phenomena. Some are devoted to text structure proper — the number and order of sentences and clauses to express the meanings of input; clause dependency structures, etc. Others deal with treatment of reference — pronominalization, ellipsis, etc. Still others take care of lexical selection, determine tense and mood features of the target text, etc. A set of text planning rules devoted to realization of *producer attitudes* is presented in Figure 2.

```
A1.  IF   (and (= clause_i.attitude.type  evaluative)
               (= clause_i.attitude.value  low)
               (= clause_i.attitude.scope  clause_i.proposition))
     THEN (add-unit-filler C_i 'attitude 'unfortunately)
A2. IF  (and (= clause_i.attitude.type  epistemic)
             (= clause_i.attitude.value  1)
             (= clause_i.attitude.scope  clause_i.proposition))
    THEN  (add-unit-facet-filler C_i 'features 'mood 'declarative)
A3. IF  (and (= clause_i.attitude.type  epistemic)
             (= clause_i.attitude.value  0)
             (= clause_i.attitude.scope  clause_i.proposition))
    THEN   (add-unit-facet-filler C_i 'features 'mood 'negative)
A4. IF  (and (= clause_i.attitude.type  epistemic)
             (= clause_i.attitude.value  0.5)
             (= clause_i.attitude.scope  clause_i.proposition))
    THEN (add-unit-filler C_i 'attitude 'perhaps)
```

Figure 2: **Text planning rules for 'producer attitudes'.**

Rule A1 deals with an attitude of the evaluative type; rules A2 through A4 with attitudes of the epistemic type. In the rules, the *if* clauses check the values in TMR and, depending onthe match, either add features to TP or add a lexical realization for the attitudinal meaning (as in Rule A4).

6 Status and Future Work.

In the DIOGENES project we adopt the methodological attitude of developing the generator functionalities in a breadth-first fashion. In other words, unlike many other projects, DIOGENES do not tend to describe exhaustively a specific linguistic phenomenon (e.g., negation, anaphora, aspect, scope of quantifiers) or type of processing (e.g., text planning, lexical selection, syntactic realization) before proceeding to the next one. We prefer instead to go for a complete functioning system which contains all (or, in practice, most) of the above components and covers all (or most) of the above phenomena. It is clear that, at the beginning, each (or many) of these components is somewhat incomplete, and not every phenomenon is described in sufficient detail. However, this methodology allows us to benefit from a complete experimentation environment and an open-ended architecture that facilitates the addition of knowledge to the system as well as testing and debugging. At present we have a working prototype text planning and generation system with narrow coverage. We are working on expanding the knowledge needed for achieving a deeper level of analysis of each of the linguistic phenomena covered in the system.

Acknowledgements.

Many thanks to the members of the DIOGENES project, especially Eric Nyberg, and to Ken Goodman for useful discussions about the material and its presentation. The first author was supported in conducting the research reported in this document by the Belgian National Incentive Program for Fundamental Research in Artificial Intelligence initiated by the Belgian state — Prime Minister's Office — Science Policy Programming. The scientific responsibility is assumed by the authors.

Bibliography.

Anscombre, J.-C. and O. Ducrot. 1983. **L'argumentation dans la langue.** Brussels: Mardaga.

Defrise, C. and S. Nirenburg. 1989. Aspects of Text Planning. CMU-CMT Memorandum. August.

Defrise, C. and S. Nirenburg (in preparation). Aspects of Text Meaning. Center for Machine Translation, Carnegie Mellon University.

Ducrot, O. 1984. Polyphonie. In *Lalies*, 4.

Hovy, E. 1987. Generating Natural Language under Pragmatic Constraints. Yale University Ph.D. Dissertation.

McKeown, K. 1985. **Text Generation.** Cambridge: Cambridge University Press.

McKeown, K. and M. Elhadad. 1989. A Comparison of Surface Language Generators: A Case Study in Choice of Connectives. MS. Columbia University.

Meteer. M. 1989. The Spokesman Natural Language Generation System. Technical Report 7090. Bolt Beranek and Newman Inc. July.

Nirenburg, S. and V. Raskin. 1987. The Subworld Concept Lexicon and the Lexicon Management System. *Computational Linguistics*, Volume 13, Issue 3-4.

Nirenburg, S., E. Nyberg, R. McCardell, S. Huffman, E. Kenschaft and I. Nirenburg. 1988. Diogenes-88. Technical Report CMU-CMT-88-107. Carnegie Mellon University. June.

Nirenburg, S. and L. Levin. 1989. Knowledge Representation Support. *Machine Translation*, 4, pp. 25 - 52.

Nirenburg, S. and I. Nirenburg. 1988. A Framework for Lexical Selection in Natural Language Generation. *Proceedings of COLING-88.*

Nirenburg, S., E. Nyberg and C. Defrise. 1989. Text Planning with Opportunistic Control. Technical Report CMU-CMT-88-113. Carnegie-Mellon University. June.

Nirenburg, S., E. Nyberg and C. Defrise. (In preparation) The DIOGENES Natural Language Generation System.

Reichman, R. 1985. **Getting Computers to Talk Like You and Me.** Cambridge, MA: MIT Press.