

TWO TYPES OF PLANNING IN LANGUAGE GENERATION

Eduard H. Hovy
USC/Information Sciences Institute
4676 Admiralty Way, Suite 1001
Marina del Rey, CA 90292-6695, U.S.A.
HOVY@VAXA.ISI.EDU

Abstract

As our understanding of natural language generation has increased, a number of tasks have been separated from realization and put together under the heading "text planning". So far, however, no-one has enumerated the kinds of tasks a text planner should be able to do. This paper describes the principal lesson learned in combining a number of planning tasks in a planner-realizer: planning and realization should be interleaved, in a limited-commitment planning paradigm, to perform two types of planning: prescriptive and restrictive. Limited-commitment planning consists of both prescriptive (hierarchical expansion) planning and of restrictive planning (selecting from options with reference to the status of active goals). At present, existing text planners use prescriptive plans exclusively. However, a large class of planner tasks, especially those concerned with the pragmatic (non-literal) content of text such as style and slant, is most easily performed under restrictive planning. The kinds of tasks suited to each planning style are listed, and a program that uses both styles is described.

1 Introduction

PAULINE (Planning And Uttering Language In Natural Environments) is a language generation program that is able to realize a given input in a number of different ways, depending on how its pragmatic (interpersonal and situation-specific)

This work was done while the author was at the Yale University Computer Science Department, New Haven

This work was supported in part by the Advanced Research Projects Agency monitored by the Office of Naval Research under contract N00014-82-K-0149. It was also supported by AFOSR contract F49620-87-C-0005.

goals are set by the user. The program consists of over 12,000 lines of T, a dialect of LISP developed at Yale University.

PAULINE addresses simultaneously a wider range of problems than has been tried in any single language generation program before (with the possible exception of [Clippinger 74]). As is to be expected, no part of PAULINE provides a satisfactorily detailed solution to any problem; to a larger or smaller degree, each of the questions it addresses is solved by a set of simplified, somewhat ad hoc methods. However, this is not to say that the program does not provide some interesting insights about the nature of language generation and the way that generators of the future will have to be structured.

One insight pertains to the problems encountered when the various tasks of generation — both of text planning and of realization — are interleaved to provide planning-on-demand rather than strict top-down planning (which has been the approach taken so far). The planning tasks that are best performed on demand tend to have short-range effects on the text (compared to those best performed in full before realization). In order to achieve the types of communicative goals such tasks usually serve, the planner must ensure that they work together harmoniously so that their effects support one another rather than conflict. This requirement imposes constraints on the organization and architecture of a generation system.

This paper describes PAULINE's architecture, the text planning tasks implemented, and how the tasks are managed. Unfortunately many details have to be left unsaid; the interested reader is referred to relevant material at appropriate points. Overview descriptions appear in [Hovy 87a, 87b].

1.1 The Problem

Depending on how the user sets the communicative goals, PAULINE produces over 100 variations of an episode that took place at Yale University in April 1986 (it also produces multiple versions of episodes in two other domains; see [Hovy 86a, 86b]). In each case, PAULINE is also given a description of the hearer and the same three principal topics from a single underlying representation network.

As a quick informal description of the episode, PAULINE says:

Example 1. YALE UNIVERSITY PUNISHED A NUMBER OF STUDENTS FOR BUILDING A SHANTYTOWN ON BEINECKE PLAZA BY ARRESTING 76 STUDENTS AND TEARING IT DOWN ONE MORNING IN EARLY APRIL. THE STUDENTS WANTED YALE TO DIVEST FROM COMPANIES DOING BUSINESS IN SOUTH AFRICA. FINALLY, THE UNIVERSITY GAVE IN AND ALLOWED THE STUDENTS TO REBUILD IT.

This is the kind of description one may hear from a passerby. In contrast, when PAULINE is given the goals to be more formal and to slant its description of the episode toward the protesters, it produces:

Example 2. AS A REMINDER TO YALE UNIVERSITY TO DIVEST FROM COMPANIES DOING BUSINESS IN SOUTH AFRICA, A LARGE NUMBER OF CONCERNED STUDENTS ERECTED A SHANTYTOWN --- NAMED WINNIE MANDELA CITY --- ON BEINECKE PLAZA IN EARLY APRIL. THE LOCAL COMMUNITY EXPRESSED SUPPORT FOR THE STUDENTS' ACTION. LATER, AT 5:30 AM ON APRIL 14, THE SHANTYTOWN WAS DESTROYED BY OFFICIALS; ALSO, AT THAT TIME, THE POLICE ARRESTED 76 STUDENTS. THE STUDENTS REQUESTED THAT YALE UNIVERSITY GIVE THEM PERMISSION TO REASSEMBLE IT ON BEINECKE PLAZA; CONCURRENTLY, SEVERAL LOCAL POLITICIANS AND FACULTY MEMBERS EXPRESSED CRITICISM OF YALE'S ACTIONS. FINALLY, YALE PERMITTED THE STUDENTS TO RECONSTRUCT THE SHANTYTOWN.

Of course, in face-to-face conversations a protester would probably be less formal and more openly opinionated. PAULINE (although now speaking in support of the university) illustrates:

Example 3. IT PISSES ME OFF THAT A FEW SHIFTLISS STUDENTS WERE OUT TO MAKE TROUBLE ON BEINECKE PLAZA ONE DAY. THEY TOOK OVER BEINECKE PLAZA

AND BUILT A SHANTYTOWN, WINNIE MANDELA CITY, BECAUSE THEY WANTED YALE UNIVERSITY TO PULL THEIR MONEY OUT OF COMPANIES WITH BUSINESS IN SOUTH AFRICA. THE UNIVERSITY ASKED THE STUDENTS TO BUILD THE SHANTYTOWN ELSEWHERE, BUT THEY REFUSED. I AM HAPPY THAT OFFICIALS REMOVED THE SHANTYTOWN ONE MORNING. FINALLY, YALE GAVE IN AND LET THE IDIOTS PUT IT UP AGAIN. AT THE SAME TIME YALE SAID THAT A COMMISSION WOULD GO TO SOUTH AFRICA IN JULY TO CHECK OUT THE SYSTEM OF APARTHEID, BECAUSE THE UNIVERSITY WANTED TO BE REASONABLE.

The construction of such texts is beyond the capabilities of most generators written to date. Though many generators would be capable of producing the individual sentences, some of the pre-realization planning tasks have never been attempted, and others, though studied extensively (and in more detail than implemented in PAULINE) have not been integrated into a single planner under pragmatic control.

This paper involves the questions: what are these planning tasks? How can they all be integrated into one planner? How can extralinguistic communicative goals be used to control the planning process? What is the nature of the relation between text planner and text realizer?

2 Interleaving or Top-Down Planning?

2.1 The Trouble with Traditional Planning

In the text planning that has been done, two principal approaches were taken. With the *integrated* approach, planning and generation is one continuous process: the planner-realizer handles syntactic constraints the same way it treats all other constraints (such as focus or lack of requisite hearer knowledge), the only difference being that syntactic constraints tend to appear late in the planning-realization process. Typically, the generator is written as a hierarchical expansion planner (see [Sacerdoti 77]) — this approach is exemplified by KAMP, Appelt's planner-generator ([Appelt 81, 82, 83, 85]). With the *separated* approach, planning takes place in its entirety before realization starts; once planning is over, the planner is of no further use to the realizer. This is the case in the generation systems of [McKeown 82], [McCoy

85], [Rösner 86, 87], [Novak 87], [Bienkowski 86], [Paris 87], and [McDonald & Pustejovsky 85].

Neither approach is satisfactory. Though conceptually more attractive, the integrated approach makes the grammar unwieldy (it is spread throughout the plan library) and is slow and impractical — after all, the realization process proper is not a planning task — and furthermore, it is not clear whether one could formulate all text planning tasks in a sufficiently homogeneous set of terms to be handled by a single planner. (This argument is made more fully in [Hovy 85] and [McDonald & Pustejovsky 85].) On the other hand, the separated approach typically suffers from the stricture of a one-way narrow-bandwidth interface; such a planner could never take into account fortuitous syntactic opportunities — or even be aware of any syntactic notion! Though the separation permits the use of different representations for the planning and realization tasks, this solution is hardly better: once the planning stage is over, the realizer has no more recourse to it; if the realizer is able to fulfill more than one planner instructions at once, or if it is unable to fulfill an instruction, it has no way to bring about any replanning. Therefore, in practice, separated generators perform only planning that has little or no syntactic import — usually, the tasks of topic choice and sentence order.

Furthermore, both these models both run counter to human behavior: When we speak, we do not try to satisfy only one or two goals, and we operate (often, and with success) with conflicting goals for which no resolution exists. We usually begin to speak before we have planned out the full utterance, and then proceed while performing certain planning tasks in bottom-up fashion.

2.2 A Solution: Interleaving

Taking this into account, a better solution is to perform *limited-commitment* planning — to defer planning until necessitated by the realization process. The planner need assemble only a partial set of generator instructions — enough for the realization component to start working on — and can then continue planning when the realization component requires further guidance. This approach interleaves planning and realization and is characterized by a two-way communication at the realizer's decision points. The advantages are: First, it allows the separation of planning and realization tasks, enabling them to be handled in appropriate terms. (In fact, it even allows the separation of special-purpose planning tasks with

idiosyncratic representational requirements to be accommodated in special-purpose planners.) Second, it allows planning to take into account unexpected syntactic opportunities and inadequacies. Third, this approach accords well with the psycholinguistic research of [Bock 87], [Rosenberg 77], [Danks 77], [De Smedt & Kempen 87], [Kempen & Hoenkamp 78], [Kempen 77, 78], and [Levelt & Schriefers 87]. This is the approach taken in PAULINE.

But there is a cost to this interleaving: the type of planning typically activated by the realizer differs from traditional top-down planning. There are three reasons for this. 1. Top-down planning is prescriptive: it determines a series of actions over an extended range of time (i.e., text). However, when the planner cannot expand its plan to the final level of detail — remember, it doesn't have access to syntactic information — then it has to complete its task by planning in-line, during realization. And in-line planning usually requires only a single decision, a selection from the syntactically available options. After in-line planning culminates in a decision, subsequent processing continues as realization — at least until the next set of unprovided-for options. Unfortunately, unlike hierarchical plan steps, subsequent in-line planning options need not work toward the same goal (or indeed have any relation with each other); the planner has no way to guess even remotely what the next set of options and satisfiable goals might be.

2. In-line planning is different for a second reason: it is impossible to formulate workable plans for common speaker goals such as pragmatic goals. A speaker may, for example, have the goals to impress the hearer, to make the hearer feel socially subordinate, and yet to be relatively informal. These goals play as large a role in generation as the speaker's goal to inform the hearer about the topic. However, they cannot be achieved by constructing and following a top-down plan — what would the plan's steps prescribe? Certainly not the sentence "I want to impress you, but still make you feel subordinate"! Pragmatic effects are best achieved by making appropriate subtle decisions during the generation process: an extra adjective here, a slanted verb there. Typically, this is a matter of in-line planning.

3. A third difference from traditional planning is the following: Some goals can be achieved, flushed from the goal list, and forgotten. Such goals (for example, the goal to communicate a certain set of topics) usually activate prescriptive plans. In contrast, other goals cannot ever be

fully achieved. If you are formal, you are formal throughout the text; if you are friendly, arrogant, or opinionated, you remain so — you cannot suddenly be “friendly enough” and then flush that goal. These goals, which are pragmatic and stylistic in nature, are well suited to in-line planning.

Generation, then, requires two types of planning. Certain tasks are most easily performed in top-down fashion (that is, under guidance of a hierarchical planner, or of a fixed-plan (schema or script) applier), and other tasks are most naturally performed in a bottom-up, selective, fashion. That is, some tasks are *prescriptive* — they act over and give shape to long ranges of text — and some are *restrictive* — they act over short ranges of text, usually as a selection from some number of alternatives. Prescriptive strategies are formative: they control the construction and placement of parts in the paragraph and the sentence; that is, they make some commitment to the final form of the text (such as, for example, the inclusion and order of specific sentence topics). Restrictive strategies are selective: they decide among alternatives that were left open (such as, for example, the possibility of including additional topics under certain conditions, or the specific content of each sentence). A restrictive planner cannot simply plan *for*, it is constrained to plan *with*: the options it has to select from are presented to it by the realizer.

2.3 Planning Restrictively: Monitoring

Since there is no way to know which goals subsequent decisions will affect, restrictive planning must keep track of all goals — conflicting or not — and attempt to achieve them all in parallel. Thus, due to its bottom-up, run-time nature, planning with restrictive strategies takes the form of execution monitoring (see, say, [Fikes, Hart & Nilsson 72], [Sacerdoti 77], [Miller 85], [Doyle, Atkinson & Doshi 86], [Broverman & Croft 87]); we will use the term *monitoring* here, appropriate for a system that does not take into account the world’s actual reaction (in generation, the hearer’s actual response), but that trusts, perhaps naively, that the world will react in the way it expects. Monitoring requires the following:

- checking, updating, and recording the current satisfaction status of each goal
- determining which goal(s) each option will help satisfy, to what extent, in what ways
- determining which goal(s) each option will thwart, to what extent, and in what ways

- computing the relative priority of each goal in order to resolve conflicts (to decide, say, whether during instruction to change the topic or to wait for a socially dominant hearer to change it)

When the planner is uncertain about which long-term goals to pursue and which sequence of actions to select, the following strategies are useful:

- prefer *common intermediate* goals (subgoals shared by various goals [Durfee & Lesser 86])
- prefer *cheaper* goals (more easily achieved goals; [Durfee & Lesser 86])
- prefer *discriminative intermediate* goals (goals that most effectively indicate the long-term promise of the avenue being explored) ([Durfee & Lesser 86])
- prefer *least-satisfied* goals (goals furthest from achievement)
- prefer *least-recently satisfied* goals (goals least recently advanced)
- *combine the latter two* strategies (a goal receives higher priority the longer it waits and the fewer times it has been advanced)

3 Planning in PAULINE

3.1 Program Architecture, Input and Opinions

The user provides PAULINE with input topics and a set of pragmatic goals, which activate a number of intermediate rhetorical goals that control the style and slant of the text. Whenever planning or realization require guidance, queries are directed to the activated rhetorical goals and their associated strategies (see Figure 1).

Prescriptive planning is mostly performed during topic collection and topic organization and restrictive planning is mostly performed during realization. Restrictive planning is implemented in PAULINE in the following way: None of the program’s rhetorical goals (opinion and style) are ever fully achieved and flushed; they require decisions to be made in their favor throughout the text. PAULINE keeps track of the number of times each such goal is satisfied by the selection of some option (of course, a single item may help satisfy a number of goals simultaneously). For conflict resolution, PAULINE uses the *least-satisfied* strategy: the program chooses the option helping the goals with the lowest total satisfaction status. In order to do this, it must know which goals each option will help satisfy. Responsibility for providing this

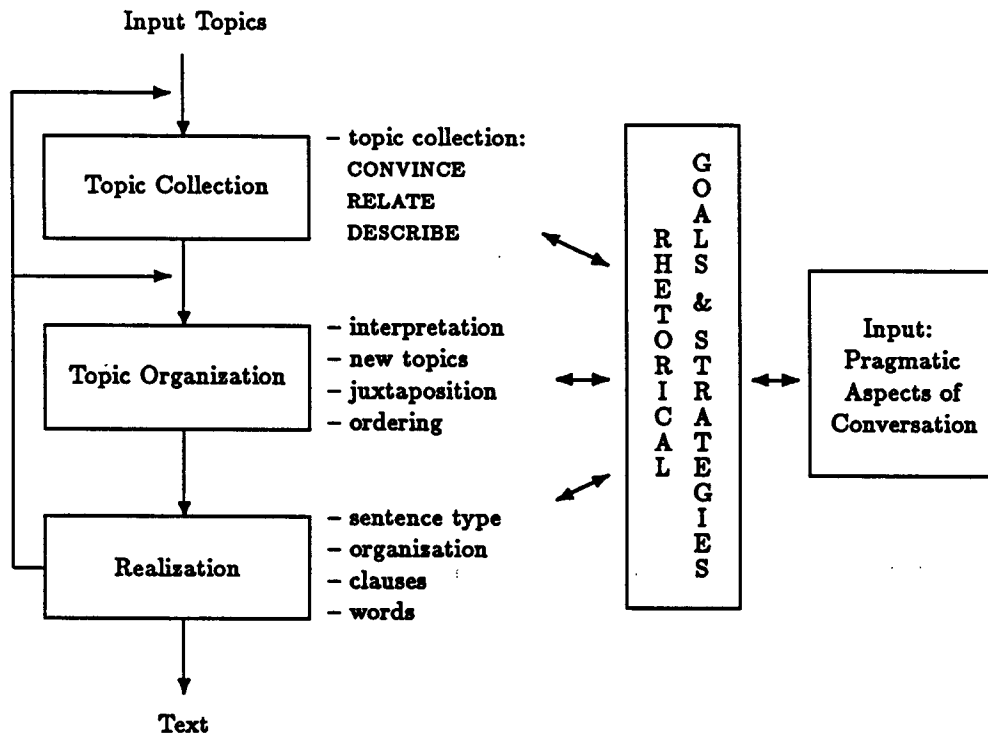


Figure 1: Program Architecture

information lies with whatever produces the option: either the lexicon or the language specialist functions in the grammar.

PAULINE's input is represented in a standard case-frame-type language based on Conceptual Dependency ([Schank 72, 75], [Schank & Abelson 77]) and is embedded in a property-inheritance network (see [Charniak, Riesbeck, & McDermott 80], [Bobrow & Winograd 77]). The shantytown example consists of about 120 elements. No intermediate representation (say, one that varies depending on the desired slant and style) is created.

PAULINE's opinions are based on the three affect values GOOD, NEUTRAL, and BAD, as described in [Hovy 86b]. Its rules for affect combination and propagation enable the program to compute an opinion for any representation element. For instance, in example 2 (where PAULINE speaks as a protester), its sympathy list contains the elements representing the protesters and the protesters' goal that Yale divest, and its antipathy list contains Yale and Yale's goal that the university remain in an orderly state.

3.2 Text Planning Tasks

This section very briefly notes the text planning tasks that PAULINE performs: topic collection, topic interpretation, additional topic inclusion, topic juxtaposition, topic ordering, intrasentential slant, and intrasentential style.

Topic Collection (Prescriptive): This task — collecting, from the input elements, additional representation elements and determining which aspects of them to say — is pre-eminently prescriptive. Good examples of topic collection plans (also called schemas) can be found in [McKeown 82], [Paris & McKeown 87], and [Rösner 86]. In this spirit PAULINE has three plans — the DESCRIBE plan to find descriptive aspects of objects, the RELATE plan to relate events and state-changes, and the CONVINCING plan to select topics that will help convince the hearer of some opinion. Whenever it performs topic collection, PAULINE applies the prescriptive steps of the appropriate collection plan to each candidate topic, and then in turn to the newly-found candidate topics, for as long as its pragmatic criteria (amongst others, the amount of time available) allow. The CONVINCING plan (described in [Hovy 85]) contains,

amongst others, the steps to *say good intentions*, *say good results*, and *appeal to authority*. Example 1 presents the topics as given; in example 2, the CONVINCE plan prescribes the inclusion of the protesters' goal and the support given by the local community and faculty; and in example 3, with opposite sympathies, the same plan prescribes the inclusion of Yale's request and of the announcement of the investigation commission.

Topic Interpretation (Prescriptive and Restrictive): As described in [Hovy 87c], generators that slavishly follow their input elements usually produce bad text. In order to produce formulations that are appropriately detailed and/or slanted, a generator must have the ability to aggregate or otherwise interpret its input elements, either individually or in groups, as instances of other representation elements. But finding new interpretations can be very difficult; in general, this task requires the generator (a) to run inferences off the input elements, and (b) to determine the expressive suitability of resulting interpretations. Though unbounded inference is not a good idea, limited inference under generator control can improve text significantly. One source of control is the generator's pragmatic goals: it should try only inferences that are likely to produce goal-serving interpretations. In this spirit, PAULINE has a number of prescriptive and restrictive strategies that suggest specific interpretation inferences slanted towards its sympathies. For example, in a dispute between "we" (the program's sympathies) and "they", some of its strategies call for the interpretations that

- coercion: they coerce others into doing things for them
- appropriation: they use ugly tactics, such as taking and using what isn't theirs
- conciliation: we are conciliatory; we moderate our demands

Interpretation occurred in examples 1 and 3: the notions of *punishment* in example 1, and of *appropriation* ("took over Beinecke Plaza") and *conciliation* ("Yale gave in") in example 3, did not appear in the representation network.

Additional Topic Inclusion (Restrictive): During the course of text planning, the generator may find additional candidate topics. When such topics serve the program's goals, they can be included in the text. But whether or not to include these instances can only be decided when such topics are found; the relevant strategies are therefore restrictive. For example, explicit statements of opinion may be interjected where appro-

priate, such as, in example 3, the phrases "It pisses me off" and "I am happy that".

Topic Juxtaposition (Restrictive): By juxtaposing sentence topics in certain ways, one can achieve opinion-related and stylistic effects. For example, in order to help slant the text, PAULINE uses multi-predicate phrases to imply certain affects. Two such phrases are "Not only X, but Y" and "X; however, Y"; depending on the speaker's feelings about X, these phrases attribute feelings to Y, even though Y may really be neutral (for more detail [Hovy 86b]). With respect to stylistic effect, the juxtaposition of several topics into a sentence usually produces more complex, formal-sounding text. For example, consider how the phrases "as a reminder", "also, at that time", and "concurrently" are used in example 2 to link sentences that are separate in example 3. The task of topic juxtaposition is best implemented restrictively by presenting the candidate topics as options to strategies that check the restrictions on the use of phrases and select suitable ones. (The equivalent prescriptive formulation amounts to giving the program goals such as [find in the network two topics that will fit into a "Not only but" phrase], a much less tractable task.)

Topic Ordering (Prescriptive): The ordering of topics in the paragraph is best achieved prescriptively. Different circumstances call for different orderings; newspaper articles, for instance, often contain an introductory summarizing sentence. In contrast to the abovementioned schemas ([McKeown 82], etc.), steps in PAULINE's topic collection plans are not ordered; additional plans must be run to ensure coherent text flow. PAULINE uses one of two topic-ordering plans which are simplified scriptifications of the strategies discussed in [Hobbs 78, 79] and [Mann & Thompson 83, 87].

Intrasentential Slant (Restrictive): In addition to interpretation, opinion inclusion, and topic juxtaposition, other slanting techniques include the use of stress words, adjectives, adverbs, verbs that require idiosyncratic predicate contents, nouns, etc. Due to the local nature of most of these techniques and to the fact that options are only found rather late in the realization process, they are best implemented restrictively. In example 2, for example, the protesters are described as "a large number of concerned students". This is generated in the following way: The generator's noun group specialist produces, amongst others, the goals to say adjectives of number and of opinion. Then the specialist that controls the real-

ization of adjectives of number collects all the alternatives that express number attributively (such as "a few", "many", a number) together with the connotations each carries. The restrictive strategies activated by the rhetorical goals of opinion then select the options of "many" and "a large number" for their slanting effect. Finally, the restrictive strategies that serve the rhetorical goals determining formality select the latter alternative. The opinion "concerned" is realized similarly, as are the phrases "as a reminder" and, in example 3, "a few shiftless students" and "idiots".

Intrasentential Style (Restrictive): Control of text style is pre-eminently a restrictive task, since syntactic alternatives usually have relatively local effect. PAULINE's rhetorical goals of style include haste, formality, detail, simplicity (see [Hovy 87d]). Associated with each goal is a set of restrictive strategies or plans that act as criteria at relevant decision points in the realization process. Consider, for example, the stylistic difference between examples 2 and 3. The former is more formal: the sentences are longer, achieved by using conjunctions; they contain adverbial clauses, usually at the beginnings of sentences ("later, at 5:30 am one morning"); adjectival descriptions are relativized ("named Winnie Mandela City"); formal nouns, verbs, and conjunctions are used ("erected, requested, concurrently, permitted"). In contrast, example 3 seems more colloquial because the sentences are shorter and simpler; they contain fewer adverbial clauses; and the nouns, verbs, and conjunctions are informal ("built, asked, at the same time, let"). Indications of the formality of phrases, nouns, and verbs are stored in discriminations in the lexicon (patterned after [Goldman 75]).

4 Conclusion

The choices distributed throughout the generation process are not just a set of unrelated ad hoc decisions; they are grammatically related or, through style and slant, convey pragmatic information. Therefore, they require control. Since traditional top-down prescriptive planning is unable to provide adequate control, a different kind of planning is required. The limited-commitment planning organization of PAULINE illustrates a possible solution.

Text planning provides a wonderfully rich context in which to investigate the nature of prescriptive and restrictive planning and execution monitoring — issues that are also important to general AI planning research.

5 Acknowledgement

Thanks to Michael Factor for comments.

6 References

1. Appelt, D.E., 1982. *Planning Natural Language Utterances to Satisfy Multiple Goals*. Ph.D. dissertation, Stanford University.
2. Appelt, D.E., 1982. Planning Natural-Language Utterances. *Proceedings of the Second AAAI Conference*, Pittsburgh.
3. Appelt, D.E., 1983. Telegram: A Grammar Formalism for Language Planning. *Proceedings of the Eighth IJCAI Conference*, Karlsruhe.
4. Appelt, D.E., 1985. *Planning English Sentences*. Cambridge: Cambridge University Press.
5. Bienkowski, M.A., 1986. A Computational Model for Extemporaneous Elaborations. Princeton University Cognitive Science Laboratory Technical Report no. 1.
6. Bobrow, D.G. & Winograd, T., 1977. An Overview of KRL, a Knowledge-Representation Language. *Cognitive Science*, 1(1), 3-46.
7. Bock, J.K., 1987. Exploring Levels of Processing in Sentence Production. In *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics*, Kempen G. (ed), 351-364. Boston: Kluwer Academic Publishers.
8. Broverman, C.A. & Croft, W.B., 1987. Reasoning about Exceptions during Plan Execution Monitoring. *Proceedings of the Sixth Conference of AAAI*, Seattle.
9. Charniak, E., Riesbeck, C.K. & McDermott, D.V., 1980. *Artificial Intelligence Programming*. Hillsdale: Lawrence Erlbaum Associates.
10. Clippinger, J.H., 1974. *A Discourse Speaking Program as a Preliminary Theory of Discourse Behavior and a Limited Theory of Psychoanalytic Discourse*. Ph.D. dissertation, University of Pennsylvania.
11. Danks, J.H., 1977. Producing Ideas and Sentences. In *Sentence Production: Developments in Research and Theory*, Rosenberg S. (ed), 229-258. Hillsdale: Lawrence Erlbaum Associates.
12. De Smedt, K. & Kempen, G., 1987. Incremental Sentence Production. In *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics*, Kempen G. (ed), 356-370. Boston: Kluwer Academic Publishers.
13. Doyle, R.J., Atkinson, D.J. & Doshi, R.S., 1986. Generating Perception Requests and Expectations to Verify the Execution of Plans. *Proceedings of the Fifth Conference of AAAI*, Philadelphia.
14. Durfee, E.H. & Lesser, V.R., 1986. Incremental Planning to Control a Blackboard-Based Problem Solver. *Proceedings of the Eighth Conference of the Cognitive Science Society*, Amherst.

15. Fikes, R.E., Hart, P.E. & Nilsson, N.J., 1972. Learning and executing generalized robot plans. *Artificial Intelligence*, 3, 251-288.
16. Goldman, N.M., 1975. Conceptual Generation. In *Conceptual Information Processing*, Schank, R.C. (ed), 289-371. Amsterdam: North-Holland Publishing Company.
17. Hobbs, J.R., 1978. Why is Discourse Coherent? SRI Technical Note 176.
18. Hobbs, J.R., 1979. Coherence and Coreference. *Cognitive Science*, 3(1), 67-90.
19. Hovy, E.H., 1985. Integrating Text Planning and Production in Generation. *Proceedings of the Ninth IJCAI Conference*, Los Angeles.
20. Hovy, E.H., 1986a. Some Pragmatic Decision Criteria in Generation. In *Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics*, Kempen G. (ed), 3-18. Boston: Kluwer Academic Publishers, 1987.
21. Hovy, E.H., 1986b. Putting Affect into Text. *Proceedings of the Eighth Conference of the Cognitive Science Society*, Amherst.
22. Hovy, E.H., 1987a. *Generating Natural Language under Pragmatic Constraints*. Ph.D. dissertation, Yale University.
23. Hovy, E.H., 1987b. Generating Natural Language under Pragmatic Constraints. *Journal of Pragmatics*, 11(6), 689-719.
24. Hovy, E.H., 1987c. Interpretation in Generation. *Proceedings of the Sixth Conference of AAAI*, Seattle.
25. Hovy, E.H., 1987d. What Makes Language Formal? *Proceedings of the Ninth Conference of the Cognitive Science Society*, Seattle.
26. Kempen, G., 1976. Directions for Building a Sentence Generator which is Psychologically Plausible. Unpublished paper, Yale University.
27. Kempen, G., 1977. Conceptualizing and Formulating in Sentence Production. In *Sentence Production: Developments in Research and Theory*, Rosenberg S. (ed), 259-274. Hillsdale: Lawrence Erlbaum Associates.
28. Kempen, G. & Hoenkamp, E., 1978. A Procedural Grammar for Sentence Production. University of Nijmegen Technical Report, Nijmegen.
29. Levelt, W.J.M. & Schriefers, H., 1987. Stages of Lexical Access. In *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics*, Kempen G. (ed), 395-404. Boston: Kluwer Academic Publishers.
30. Mann, W.C. & Thompson, S.A., 1983. Relational Propositions in Discourse. USC/Information Sciences Institute Research Report RS-83-115.
31. Mann, W.C. & Thompson, S.A., 1987. Rhetorical Structure Theory: Description and Construction of Text Structures. In *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics*, Kempen G. (ed), 85-96. Boston: Kluwer Academic Publishers.
32. McCoy, K.F., 1985. The Role of Perspective in Responding to Property Misconceptions. *Proceedings of the Ninth IJCAI Conference*, Los Angeles.
33. McDonald, D.D. & Pustejovsky, J.D., 1985. Description-Directed Natural Language Generation. *Proceedings of the Ninth IJCAI Conference*, Los Angeles.
34. McKeown, K.R., 1982. *Generating Natural Language Text in Response to Questions about Database Queries*. Ph.D. dissertation, University Of Pennsylvania.
35. Miller, D.P., 1985. *Planning by Search Through Simulations*. Ph.D. dissertation, Yale University.
36. Novak, H-J., 1987. Strategies for Generating Coherent Descriptions of Object Motions in Time-Varying Imagery. In *Natural Language Generation: Recent Advances in Artificial Intelligence, Psychology, and Linguistics*, Kempen G. (ed), 117-132. Boston: Kluwer Academic Publishers.
37. Paris, C.L. & McKeown, K.R., 1987. Discourse Strategies for Descriptions of Complex Physical Objects. In *Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics*, Kempen G. (ed), 97-116. Boston: Kluwer Academic Publishers.
38. Paris, C.L., 1987. *The Use of Explicit User Models in Text Generation: Tailoring to a User's Level of Expertise*. Ph.D. dissertation, Columbia University.
39. Rosenberg, S., 1977. Semantic Constraints on Sentence Production: An Experimental Approach. In *Sentence Production: Developments in Research and Theory*, Rosenberg S. (ed), 195-228. Hillsdale: Lawrence Erlbaum Associates.
40. Rösner, D., 1986. *Ein System zur Generierung von Deutschen Texten aus Semantischen Repräsentationen*. Ph.D. dissertation, Universität Stuttgart.
41. Rösner D., 1987. The Automated News Agency SEM-TEX — a Text Generator for German. In *Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics*, Kempen G. (ed), 138-148. Boston: Kluwer Academic Publishers.
42. Sacerdoti, E., 1977. *A Structure for Plans and Behavior*. North-Holland Publishing Company.
43. Schank, R.C., 1972. 'Semantics' in Conceptual Analysis. *Lingua* 30(2), 101-139. Amsterdam: North-Holland Publishing Company.
44. Schank, R.C., 1975. *Conceptual Information Processing*. Amsterdam: North-Holland Publishing Company.
45. Schank, R.C. & Abelson, R.P., 1977. *Scripts, Plans, Goals and Understanding*. Hillsdale: Lawrence Erlbaum Associates.