Natural language generation:
the state of the art and literature

Athanasios Daradoumis

Report LSI–90–46

NATURAL LANGUAGE GENERATION: THE STATE OF THE ART AND LITERATURE

Athanasios Daradoumis
Facultad d´ Informática de Barcelona
Pau Gargallo, 5. Barcelona 08028. **SPAIN**

ABSTRACT

The purpose of this paper is to present the main issues and concepts of the generation process as well as the principal achievements accomplished in this field up to date. Since generation is a relatively new developing research area, our objective is to sort out its goals and identify its problems in an attempt to establish a coherent framework in which to fit the various approaches adopted by generation researchers. As a result, we identify and clarify standard components and terminology considered by these approaches in order to present an integrated and interrelated study of the young but continually growing generation technology.

May 30, 1990

INDEX

1. INTRODUCTION

What is natural language generation? As a field, natural language generation is part of artificial intelligence and computational linguistics. It is not concerned simply with making computers able to print out fluent texts by using simple techniques based on pre-stored text and uncomplicated substitutions. Instead, it looks ahead to the time when machines will think complex thoughts and need to communicate them to their human users in a natural way. Consequently, it is the process of deliberately constructing a natural-language **text** in order to meet specified communicative goals. "**Text**" is a term that has been widely used in the generation field [Mann et al. 1982]. However, it is a general, recursive term that can apply to utterances or part of utterances of any size, spoken or written. Depending on the goals to be achieved -- with these goals coming from several sources such as an expert reasoning system, a dialog system or an ICAI tutor -- the texts that are produced may range from a single phrase given in answer to a question, two or three sentences at a time in order to satisfy multiple goals, through multisentence remarks and questions within a dialog up to full-page explanations.

Text generation has been studied seriously in artificial intelligence and computational linguistics only in the last eight or ten years, and so it is still sorting out its goals and identifying its problems. Up to that time the major research effort was concentrated on its complementary companion, the natural language understanding. Only around 1980, the growing interest in discourse and pragmatics led to increased concern with developing systems that could understand as well as produce multisentence texts. After that, a long term view has been established that is the process of creating a technology for building computer programs that can function as authors or speakers. However, in the text generation programs in existence today there is very little that deserves the title of "author" or "speaker". This is because, writing and speaking are regarded as complex arts capable of high refinement and great intellectual achievement. In contrast, programs today reflect only fragments of the most basic skills.

As a result of this research, NL generation widened into a much larger problem, comprising three facets:

- **Content Determination**
- **Text Organization (Planning)**
- **Realization as a NL Text**

The first two deal with the problem of **deciding what to say** (and **when to say it**) and constitute the **strategic**(or **planning**) **component** of the generation system, while the third deals with the problem of **deciding how to say it** effectively and that makes up the **tactical component** [McKeown 1985] or, as others call it, the **linguistic component** [McDonald 1983]. As we shall see, there is **no** clear division between the two components, neither from the point of view of system implementation nor from the point of view of terminology.

As such, *content determination* concerns what to include in a text and depends both on general properties of the system's domain and task and on the specific discourse context.

Text planning concerns organizing the content to be communicated so that the resulting discourse is appropriate to at least (1) the function of the text and (2) the intended audience.

Realization involves mapping the structured content proposed by *content determination* and *text planning* processes into a NL text. It is concerned with both lexical and syntactic choices, as well as with other decisions such as reference, ellipsis, etc.

In this paper, we first point out the acceptance of NL generation as an important and exclusive area of research and justify the need for a growing effort and further work for the development of a computational theory of text generation and communication (section 2). Then, in section 3, we discuss the ways in which NL generation differs from NL understanding as well as the adapted or shared techniques that could be exploited by the two processes.

In section 4, we make a first attempt to sort out issues and problems on generation terminology and components by presenting arguments from the point of view of several researchers. In the following sections, we point out various related research topics. In particular, in section 5, we discuss *text organization* -- discourse structure and coherence -- and generation and we mainly focus on two important works done in this area.One has been done by McKeown who developed a computational model of discourse strategies (abstracted from previous texts), and focus rules to guide the generation process about *what to say next*. The other is a work by Mann and Thompson who developed more general discourse strategies than those of McKeown, and which are derived from goal pursuit rather than past texts.

In section 6, we present two other important works on *planning and generation* . First, we discuss the work done by Appelt who developed a theory of language generation based on planning, and whose system reasons from speaker's goals to produce a plan which in turn is refined until an English sentence is completely specified. Then, we describe a distinct work done in this area by Mellish and Evans who start with a non-linear plan structure produced by an AI planning program and generate natural language text explaining how to execute the plan.

Section 7 is entirely devoted to the work of McDonald who viewed generation as a decision-making process, with the decisions characterized as *indelible,* and intended to develop a generation system that could serve as a psychological model of the human behavior.

In section 8, we examine two main issues arised from knowledge needed for generation: the content of a knowledge base and the knowledge representation formalisms used. Next, in section 9, we discuss the nature of a generator's lexicon and particularly, the use of a phrasal lexicon -- a conceptual extension of a standard word-based lexicon -- that demonstrated the ability of using specialized knowledge in generation (such as, complex, "fixed", or idiomatic terms). More specifically, three generation research projects are described: The PHRED generator of Jacobs that uses a pattern language to organize a phrasal lexicon as a shared linguistic knowledge base (section 9.1), the generation system of Kukich that organizes a phrasal lexicon as a set of production rules that integrate semantic and linguistic knowledge (section 9.2), and finally, Hovy´s generator who groups all linguistic knowledge in a phrasal lexicon as a set of syntax specialist procedures (section 9.3).

In section 10, we discuss the important role that grammars play in generation. In particular, in section 10.1, we examine two grammatical formalisms: the ATN and Systemic grammars that have been used to thoroughly control the generation process. Then, in section 10.2, we discuss how the use of the TAG grammars succeeds in specifying all the valid surface structures that texts can have, and consequently representing a surface structure as an intermediate level of representation between the levels of the "message" and the words of the text. Finally, in section 10.3, we present the FUG formalism that is used to give a generator a modular, independent way of supplying the purely linguistic information that the process must have and do so without imposing specific demands on its control structure.

In section 11, we entirely concentrate on the work done by Danlos who considered text generation in Roman languages. Thus, taking into account the rich inflectional system of Romance languages -- an issue which has not been seriously considered in English -- she had to specify the way of integrating conceptual, linguistic, syntactic and morphological decisions. She did that by developing a "discourse grammar", and by using a "lexicon grammar" as well as syntactic and morphological operations in an interacting way.

Finally, in section 12, we look at somewhat related areas to NL generation such as: the area of explanation, the area of psycholinguistics as well as the translation and the database (question/answering) systems.

Note: A very good and general-purpose overview of NL generation was done by McDonald and can be found in [Shapiro 1987]. However, our paper concentrates on the generation process itself (and not on the description and evaluation of particular generation systems) in an attempt to determine and level out the diversity of its approaches depending on the issues each approach has focused on. This is done by examining common points and differences between various research projects as regards the terminology they used, and the way they conceived generation´s issues and divided it into components. Doing this way, we try to give the feeling of a more global and comprehensive treatment of the whole generation process. A separate, complementary paper deals exclusively with NL generation systems in an evaluative and comparative manner as there exist quite interesting and competent generation programs developed during the last decade.

2. THE NEED FOR AN EXTENSIVE RESEARCH IN NL GENERATION

Historically, NL generation is a much more recent domain than NL understanding: since 1950's we have seen numerous programs of understanding, while genereation has just started its development since the last decade. Nevertheless, programs have been printing natural-language messages at their users for as long as there have been computers, for example an error message *File DOC.TXT does not exist* after a print command of the file DOC.TXT. These messages are pre-stored in the computer in the form of phrases that support variables (e.g., *File* F_x *does not exist*); at the right moment the variables are substituted by a value and the complete phrases are displayed on the screen, without any linguistic computation to be performed. The simplicity of the method of pre-stored phrases and the fact that they satisfied the needs of a specific moment resulted in the belief that text production in natural language was an easy task.

Let's see now how that belief was overrulled as more systems have been developed which required the capability for sophisticated communication with their users [Danlos 1985]. We suppose that after a command to print a file that does not exist on the current directory, one would like to have messages indicating the files that could correspond to the one the user wants to print, for example, after the print command of the file DOC.TXT, to have messages such as:

(1) There is no file DOC.TXT, but there is a file DOC.PAS
Do you want to print DOC.PAS? **(Answer Y/N)**

(2) There is a file DOC-MIN.TXT and a file DOC-MAX.TXT.
Do you want to print DOC-MIN.TXT? **(Answer Y/N)**
Do you want to print DOC-MAX.TXT? **(Answer Y/N)**

(3) The file DOC.TXT is not in your current directory, but it is in your directory <DOCUMENTS>.
Do you want to print it anyway? **(Answer Y/N)**

The production of these messages brings up the first main issue of the generation process: to determine the information that should be transmitted to the user **(decide what to say)**. That is, the messages (1) - (3) are not satisfied by only indicating the files that could correspond to DOC.TXT: they also support an error explanation and one or more questions. So, having determined the information that should appear in the messages, one could try to use pre-stored phrases for every information and juxtapose them without performing any linguistic computation. Let's use the following pre-stored phrases:

The file F_x *does not exist* (for the error explanation)

The file F_y *is in your directory* D_i (for the existence of a file similar to the one to be printed)

Do you want to print F_y *?* (for the question).

This method will result in the following messages corresponding to (1) -(3) respectively:

(1´) The file DOC.TXT does not exist. The file DOC.PAS is in your current directory.
Do you want to print DOC.PAS?

(2´) The file DOC.TXT does not exist. The file DOC-MIN.TXT is in your current directory.
The file DOC-MAX.TXT is in your current directory.
Do you want to print DOC-MIN.TXT?
Do you want to print DOC-MAX.TXT?

(3´) The file DOC.TXT does not exist. The file DOC.TXT is in your directory <DOCUMENT>.
Do you want to print DOC.TXT?

As we see, these messages are awkward, and this is true of all messages obtained by juxtaposition of pre-stored phrases. One could improve the method by introducing a **semantic computation**, for example by proposing for the error explanation two pre-stored phrases:

The file F_x is not in your current directory.

The file F_x does not exist.

with the choice between the two forms depending on the existence of a file with identical name to the one to be printed. This refinement results in the text corresponding to case (3):

(3*) The file DOC.TXT is not in your current directory. The file DOC.TXT is in your directory <DOCUMENTS>.
Do you want to print DOC.TXT?

which is less awkward than (3´), but it is still a poor one. Therefore, the method of "elaborated" pres-stored phrases is not sufficient.

On the other hand, the formation of texts that look natural, pass through **syntactic operations** (e.g., pronominalization, coordination, omission of complements, etc.). So, one could try to perform syntactic operations on "elaborated" pre-stored phrases. But from one part the addition of syntactic operations would require that the pre-stored phrase to be annotated with linguistic information. For example, the substitution of a noun phrase by a pronoun requires that we know the syntactic function of the noun phrase (subject, object, etc.) and the gender and number of the head noun, so that we determine the form of the pronoun (he, she, they, him, etc.). From the other part, the application of syntactic operations does not have the simplicity of the pre-stored phrases method. The conditions of applying these operations are very complex, in particular they are not local: in general, it is not sufficient to examine whether the element, to which we apply an operation, satisfies certain characteristics; instead, we should also examine several parameters of a varying nature (pragmatic, semantic, syntactic and lexical).

Finally, even though we are able to apply syntactic operations on elaborated pre-stored phrases annotated with linguistic information that considerably improve the quality of the obtained texts, we can not expect to have high-level results. This is because the formation of a text consisting of a set of information $I_1, I_2, \ldots, I_n$ does not result in a simple juxtaposition of phrases $P_1, P_2, \ldots, P_n$ expressing $I_1, I_2, \ldots I_n$ respectively. We have to take into account phenomena such as the existence of a phrase P_i that expresses I_k and I_i simultaneously, or the fact that P_i can be inferred by P_m. In otherwords, we must make an appeal for a global apreciation of the information to be transmitted as well as the means to express it. As a result, the production of well-prepared texts requires the abandonment of the pre-stored (or "canned") text method to the profit of automatic generation techniques.

The actual goal of research in natural language generation

Research on natural language generation has then as its goal not just competent performance by a computer but the development of a *computational theory* of the human capacity for language and the processes that engage it. For generation this focuses on the explanation of two key matters: *versatility* (varying the texts in form and emphasis to fit an enormous range of speaking situations), and *creativity* (with the potential to express any object or relation in one´s mind as a natural language text). But, what do people know about their language, what processes do they employ that enable them to be versatile and creative? The need to accomodate these capabilities is the prime organizing force behind generation theories.

In other words, research on generation is often aimed at purely scientific concerns about the nature of language and language use in people. As such, its methods are "synthetic" rather than "analytical". In AI studies of generation, one experiments by constructing computer programs, observing their behavior, and comparing it to the behavior of the natural system under study. This approach has both strengths and weaknesses:

Since the computer is the main medium for the definition and modeling of processes, we can be much more concrete than other disciplines in modeling psychological phenomena as interacting processes. We are also sure for the internal coherence of our theories that many other disciplines lack, since we are forced to have complete, executable modules before our computer programs will run.

At the same time, however, there is the weakness that we continually make a step into the unknown: in order to give our programs something interesting to work from, we are forced to consider the nature of thought and intention -- the bases of language -- and must set our considerations down in a usable notation. In this respect our problems in generation are quite different from those of its sister area in AI, NL understanding, a difference that is worth looking at, as we see in the next section.

3. COMPARING NATURAL LANGUAGE UNDERSTANDING AND GENERATION

Before going into an explicit presentation of the organization of the generation process and a detailed study of its components and issues, it would be helpful to make a brief comparison of it with its more studied complementary process, NL understanding. This will not only help us to understand the different organizational structure of generation, but also to see how we can take profit of techniques already used in one field in order to apply them to the other.

Differences Between Natural Language Understanding and Generation

When the task is understanding language it is very clear what one is starting from. NL understanding is commonly factored into morphology, syntax, semantics, discourse analysis and pragmatics. However, the generation process has a fundamentally different character. This fact follows directly from the differences in the information flow in the two processes. *Understanding* proceeds from texts to intentions; *generation* does the opposite. *In understanding* , the "known" is the text itself and from its wording the process constructs and deduces the propositional content ("meaning") conveyed by the text and the probable intentions of the speaker in producing it. Its primary effort is to scan the words of the text in sequence, during which the form of the text gradually unfolds.

Generation has the opposite information flow. It proceeds from content to form, from intentions and perspectives to linearly arranged words and syntactic markers. Its "known" is its awareness of its intentions, its plans, and the text it has already produced. By the means of its model of the audience, the situation, and the discourse, it is possible to make choices among the alternative wordings (*lexical choices*) and constructions (*syntactic choices*) that the language provides -- the primary effort in constructing a text deliberately. For instance, a generator must decide that although both the active and passive forms may be possible, the active, for example, is better than the passive.

Furthermore, the generator must have a principled reason for making that decision, which it can use in all similar cases. Research in NL understanding needs to use the evidence provided by the text to examine the limited set of available options to determine the option actually taken (i.e., a system would use the evidence that a verb -- like "produce" -- occurs in the active form to determine that a noun phrase is the object "being produced" and another one is the agent that "does the producing"). However, research in generation must specify why one option is better than others in various situations.

Generation addresses two principal questions: **what to say** and **how to say it**. Under the first heading come such questions as:

- *What does the other participant need to know?* When users ask questions, they need to know the answer or to know that their question does not make sense. They may also need to know things they haven´t thought to ask. Or, the system may have recognized that the user has made a mistake or is doing something the hard way. The user should be told this, without having asked.
- *In what terminology should the information be conveyed?* For example, if we tell medical personel that a drug therapy program is trying to "check sensitivities", will they know what that means or must our explanation be given in terms of what "checking sensitivities" means?
- *How explicit does a text have to be?* This is a generator´s problem to choose from its oversupplied sources how to adequately signal intended inferences to the audience and what information to omit from explicit mention in the text. For example, in explaining a causal chain, is it sufficient to just give the endpoints and assume that the user will be able to make the connection, or is it necessary to explicitly describe the intermediate links?

Under the question of "**how to say it**" fall such questions as:

- *What is to be presented explicitly?* A system should be able to identify what it can ommit from its explicit message because the user already knows the information or because the information will be communicated implicitly through lexical, syntactic or organizational choices.
- *What linear order should it be presented in?* We should present a good formation of the sentence and its constituents according to the rules of the grammar of the object language. Moreover, linearization into sentences should include the choice between juxtaposition, subordination, relativization or coordination as a sentence linearization procedure.
- *What underlying structure should it reflect?* We would like a system to make use of appropriate sentence and paragraph structures.
- *According to whose point of view should it be presented?* Ideally we would like a system to describe things from the user´s point of view so as to make the description easier to understand.

Besides the two main questions addressed by generation, one could possibly consider a third one: **When to say something** [McKeown 1985] which can either be placed between the *"what to say"* and *"how to say it"* questions or simply make it part of the *"what to say"* question. The *"when to say"* question contributes to the ordering of the information (*text planning*), i.e., the organization of individual sentences in a text in a specific order so that to obtain a connected text (and not single isolated sentences). On the other hand, by including only that information which is relevant to the discourse goal, we achieve that a coherent and well organized text to be generated.

In conclusion, a language generation system must be able to decide **what** information to communicate, **when** to say what, and **which** words and syntactic structures best express its content.

Yet in generation one has to deal with a deeper set of questions or one will not be doing serious work. For instance: How does the form of the conceptual representation that the process starts with affect the final text and the intermediate processing? Why will one text be effective and another not be, even though they are very similar in form? And of course the deepest question that we also identified above: How do we achieve our goals through the use of language? Just as AI research on generation must step into the unknown to situate the representational structures from which its process starts, understanding research must ultimately do the same to determine the representation of "meaning" with which it presumably must end.

The conceptual (knowledge) representation that generation process starts with, is provided by what is uniformly referred to as the *underlying program*. This can be almost any type of AI system one can imagine: cooperative database, expert diagnostician, ICAI tutor, commentator, apprentice, advisor, mechanical translator. The nature of the underlying program presently has no significant influence on the generator´s design. Today most generator researchers work most often with underlying programs that are expert advisors [McDonald 1985]. Advisor programs and intelligent machine tutors are likely to have a good understanding of what their human interlocutors are thinking. These features make them able to motivate fairly sophisticated texts, which makes them attractive to those generation researchers who are looking for already developed programs to work with.

Thus the generation process starts within the underlying program when some event leads to a need for the program to "speak". In the simplest case this may be the need to answer a question from the user; with a sophisticated discourse controller it may be the perception of a need to interrupt the user´s activities in order to point out an actual problem; or it may be the need to generate a well-formed, multisentence text of a newspaper´s style. Whatever the objective is, once the process is initiated, three kinds of activities must be carried out. These activities actually factor generation into three stages:

- **Content Determination** (*identifying the goals* the utterance is to achieve)
- **Text Planning** (*planning how the goals may be achieved* , including evaluating the situation and the available communicative resources)
- **Realization** (*realizing the plans as a NL text*)

Content determination involves identifying information (the goals) that support general decisions about what to include in the text -- for example, what to include in an explanation, what to include in a persuasive argument or justification, what to include to show the similarities and differences between two kinds of things, what to include in a request for information so that it can be satisfied. These decisions on the part of a speaker (here, the system) can depend on such issues as: what the speaker believes the hearer wants to know, what the speaker believes the hearer already knows, and what the speaker believes the hearer needs to know. *Content determination* thus depends both on general properties of the system´s domain and task and on the specific discourse context.

Text planning concerns organizing the content to be communicated so that the resulting discourse is appropriate to at least (1) the function the text is to serve, (2) the intended audience, and (3) the amount of content to be conveyed. Planning actually involves the selection (and deliberate ommision) of the information units to appear in the text (e.g., concepts, relations, individuals) and the adoption of a coordinating rhetorical framework or schema -- a text structure, for the utterance as a whole (e.g., temporal progression, compare and contrast). In fact, text structures often conform to familiar patterns that are associated either with the content of the material to be presented or with different rhetorical styles. For example, explanation of physical processes are often presented with the overall stucture of a temporally ordered chain of causes and effects, with a definite start and a definite end, even when things are essentially simultaneous. Comparisons of things are done either by describing each one in turn or by interleaving their descriptions so that points of similarity and difference are revealed [McKeown 1985]. Rhetorically, a newspaper report of a crime will differ from a police account in the order and focus of the material presented. Finally, particular perspectives may be imposed on the information units to help in the signaling of intended inferences.

Realization involves mapping the stuctured content proposed by *content determination* and *text planning* processes into an NL text. It depends on a sophisticated knowledge of the language´s grammar and rules of discourse coherency, and typically constructs a syntactic description of the text as an intermediate representation. Decisions here include: (1) Appropriate *lexical choices* in how to describe or refer to things, depending on the intended audience (e.g., its level of expertise), on the function of the description within the utterance (e.g., whether it is only being used to refer or whether the description is significant to what is being affirmed) and on the preceding discourse, and (2) Appropriate *syntactic choices* , depending on such discourse-specific properties as the current focus of attention and syntactic choices made in previous sentences. For example, one speaks about choosing to "realize" a modification relationship as either an adjective or a relative clause. In such cases, we do not only pay attention to the linguistic form but also to the knowledge of the criteria that dictate how these forms are used. *Realization* must justify the reasons of choosing one option rather than the other.

Concluding, while syntax, semantics, discourse and pragmatics all play a role in NL generation, it seems that their operation must be equally integrated as in language understanding (*and in some cases even more*). We look at this point in more detail in the sections 6, 9, and 11.

Shared Techniques Between NL Understanding and Generation

In closing this section, we discuss how techniques developed in one area can be adopted in the other. This is actually the case for the generation process that has been able to take advantage of already studied and utilized techniques in NL understanding.

As we have seen above, rather than breaking along "traditional" linguistics lines of syntax, semantics, discourse, and pragmatics (which correspond to different level descriptions used in NL understanding), NL generation has broken along task-oriented lines -- the tasks required for creating and realizing a text. Each of these tasks may use many types of knowledge, linguistic and otherwise. Nevertheless, attempts have been made to reconcile this broad task-oriented view of NL generation (including everything from *content determination* to actual *realization*) with the knowledge-oriented view of NL understanding. As a result, there have been some instances of adapted or shared techniques in the following areas:

In the area of discourse , [McKeown 1985] has extended the rules of focus movement of [Sidner 1982] for use in both *text planning* (i.e., selecting what to talk about next in a way that maximizes text coherence) and *realization* (e.g., allowing the system to use anaphoric terms appropriately by choosing when to use pronouns and definite noun phrases). Similarly, [Appelt 1985] has applied other parts of Sidner´s work to planning description although he adopts the later notation of [Grosz et al. 1983].

In the area of syntax and lexical semantics (*realization*), [Simmons and Slocum 1972] were the first to treat NL generation as an inverse single sentence understanding problem by using an ATN grammar to traverse a semantic network structure based on the concept of verb-centered case frames by thus producing a corresponding english sentence in the process. Later on [McKeown 1985] and [Appelt 1985] have used the functional unification grammar of [Kay 1984] that was designed specifically for such sharing of syntactic and lexical semantic knowledge between NL generation and understanding (See also section 10.3).

As wee have seen in the beginning of this section, the two tasks of NL understanding and generation have been treated independently, principally because some of the hard problems in language production differ from those of language analysis. However, computer programs that can effectively communicate in natural language must be capable both of analyzing a range of utterances to derive their meaning or intent, and of producing appropriate and intelligible responses.

PHRED [Jacobs 1985a], a realization component for Berkeley´s UNIX Consultant system (UC), is based on sharing its declarative linguistic knowledge base (i.e., its syntactic and lexical patterns) with UC´s PHRAN component, which analyzes users´ NL queries to the system. PHRAN/PHRED´s knowledge base is organized into pattern-concept rules whose patterns embody both syntactic and lexical-semantic aspects and reflect a particular approach to NL processing, similar in many ways to semantic grammars. Other complete NL systems like PHRAN/PHRED that exploit knowledge shared between analyzer and generator are: The HAM-ANS question-answerer [Wahlster et al. 1983], [Busemann 1984] makes use of a shared lexicon. The VIE-LANG system [Buchberger and Horacek 1984] shares a "syntactico-semantic" lexicon, but the generator accesses this lexicon using a discrimination net with specialized choice knowledge.

There are a number of practical advantages of this approach: Having a knowledge base shared between analyzer and generator eliminates the redundancy of having separate grammars and lexicon for input and output. It avoids possibly awkward inconsistencies caused by such a separation, and allows for interchangeable interfaces, such as the English and Spanish versions that the UC system has implemented.

One area in which there has been little sharing or adaptation so far has been in applying the macro structures found in "schemata" or "text grammars" of the *text planning* process to discourse understanding.

With its opposite flow of information, one might assume that the generation process could be organized like an understanding process but with the stages in opposite order. To a certain extent this has been true; however, later generation research has shown that there can be strong interactions among the stages. In the next section, we examine this fact and attempt to identify these interactions and put some order in the way generation stages and components have been defined in literature and developed in research projects.

4. SORTING OUT ISSUES ON GENERATION TERMINOLOGY AND COMPONENTS

In this section, we make a first attempt to sort out issues that concern both generation components and teminology, by briefly presenting the points of view of three generator researches (Danlos, McDonald, and Appelt), who found interactions between the various generation stages. We also identify the different ways that researches have defined the control of the generation process.

The syntax and lexicon of a language become both resources and constraints, defining the elements available for the construction of the text and also the dependencies between them that determine their valid combinations. These dependencies, and the fact that they silently govern when the information, on which each decision depends, can become available are the fundamental reason why many generation programs do largely follow the conventional stages identified by linguists:

Identification of intention (goals) largely precedes any detailing of the conceptual information the audience should be given *(content selection and rhetorical planning).*

The *planning of the rhetorical structure* that will be imposed on the information largely precedes any *construction of syntactic structures* to realize it.

Finally, the *syntactic context of a word* must be fixed before the precise *morphological form* it should take can be known.

One could consider this kind of organization because it is a natural order in which to make **decisions:** decisions on the order of information, decisions to use certain words or syntactic constructions and decisions to place constraints on later decisions; it is **simpler** to go with the flow of the dependencies rather than jump ahead and take the chance that a premature decision will have to be undone because it later turns out to be inconsistent. Thus, *generation is above all a careful planning process.* It involves realizing goals in the presence of constraints and limitations on resources.

Issues on the Interaction of Generation Stages (Components) from Three Points of View

Given the stages that a generation program may follow in order to produce natural language text, it does not mean that the generation process must consider a strictly predetermined order as the one presented above. Later research on the generation of well-constructed texts -- in the style of a newspaper [Danlos 1984] -- has shown that it would be a mistake to stick at a fixed ordering like this. Certain decisions that naturally belong to a considered "low-level" stage may have to be decided "early", that is in a higher stage (for example, the passive transformation -- which is a syntactic operation -- is relevant to the discourse semantics and therefore should be decided "early");

and vice-versa: decisions of a "high-level" stage may have to be made in a "later" one (for example, transformations that affect the sentence form -- which are supposingly "high-level" decisions -- and/or that depend upon the pronominalization decisions should be decided "late"). In other words, decisions involved in various stages of the generation process are operations that depend on each other, none being "higher" than the other ones.

[McDonald 1985] has concentrated his research on understanding how best to represent what decisions are possible and the dependencies among them, as well as on how to represent the constraints and opportunities earlier decisions place on later ones as the process proceeds (for more detail see section 7).

His initial objective has been to model the production of spoken, and not written, english text based on the common experience that people usually start talking without knowing (or having totally decided) how they will finish. They may only know that they will try and speak about such and such a relation or make a certain reference. This motivated the use of indelible left-to-right decision-making procedures that increases the program´s efficiency. Decisions are indelible, in the sence that they cannot be taken back once they have been made. Once the generator has decided to use a certain construction and made it part of the surface structure, it cannot change its mind by aborting the entire process and starting over. Moreover, decision-makers have a limited lookahead in the sense that they never can have more than a vague idea of what decisions will happen after theirs.

Therefore, only a limited part of the text is planned at a time. It must therefore be possible to represent *planned but not yet uttered text and grammatical relations* , as well as *planned but unexecuted decisions* , since decision-making continues as a text is actually being uttered. As a result, planning proceeds in layers of refinement and must appreciate the linguistic consequences of its decisions. On the other hand, the realization of units in early layers creates a grammatical context that imposes constraints on the range of realizations that can be planned for later. Planning and realization are assumed to be asynchronous.

This design was therefore found to lead to behavioral properties of a virtual machine (i.e., to a psychological model that simulates human behavior) with important linguistic and psycholinguistic consequencies, such as those presented above, as well as the advantage of limiting the amount of processing that needs to be done to produce an utterance. McDonald has anyway focused his research on the process of deciding the linguistic form of the text and of actually uttering it, rather than on its content determination and organization. His linguistic component accepts as input a conceptual representation organized and produced by another program (a text planner). This means that he has not yet implemented a planning component, but rather he expects to develop it so that it produces a conceptual representation that has the same form as the one he is actually using.

Finally, [Appelt 1985] has been strongly opposed to the independence of the two principal questions of generation: *what to say?* and *how to say it?* and examined their interactions among all stages of the generation process. In fact, he made the point that there is no clean line between planning and realization: he showed that his planning formalism could be used not only for planning speech acts (*what to say*) but for determining the syntactic structure and lexical items of the text as well (*how to say it*). According to his theory, the planning of an action appropriate for a given situation necessitates consideration of several crucial elements: different kinds of goals that are satisfied by utterances; the knowledge of the hearer; general knowledge about the world; the constraints imposed by the syntax of the language. The utterance planner can **integrate** these diverse knowledge sources to arrive at a plan involving abstract specifications of speech acts and then, finally, produce English sentences. In other words, the planning at each level involves consideration of both linguistic rules and goal satisfaction. The distinction between "*what*" and "*how*" then becomes merely two points on a continuum between goal-satisfaction and rule-satisfaction processes, and no modularization based on that distinction is obvious.

Thus *content determination* need not be a monolithic operation, done before *text planning* and *realization* : decisions made by either of these two processes should be able to suggest revisions or additional content; in other words, goals may emerge or change in priority opportunistically as planning and even realization proceeds.

Concluding, we have seen that one should not consider the three stages of the generation process as independent modules or fixed in a certain order. There may be various interactions and dependencies among all of them that one has to take into account when designing a generation system. In fact, these components, interactions and dependencies are described in more detail in following sections (see sections 6, 9.3, and 11); for this reason, only a brief presentation of them has been made here.

In the next section, we identify and present the components in which the generation process **may** be "modularized" -- recalling that this feeling of modularity is quite relative. In other sections (as we just stated above: 6, 9.3, and 11), we shall clearly point out the differences between research projects as regards the modularity of the generation process from both implementation and terminology points of view.

Control of the Generation Process

Other differences result from the diversity of the ways researchers have defined the **control of the generation process:** What decides for the choices that have to be made in a given situation? What provides the basis for ordering them? How does one organize and represent the intermediate results? What awareness does the system have of the dependencies between choices? How are these dependencies represented and made to influence the control algorithms? As a consequence, the generation process may be directed by a **centralized control** (by a grammar, such as ATN´s and systemic grammars), a **distributed control** (by various components one of which is the grammar, such as a FUG grammar), and **control by progressive refinement of the "message"** with the grammar used to specify all the valid surface structures that text can have (message-driven approaches, TAG grammars) -- see section 10.

Furthermore, another important question is: What is a person´s model of the differences between syntactically or lexically similar versions of the same text and of the impact they will have on an audience? Alternative answers to these questions will be described throughout the rest of the paper.

Issues on Generation Terminology

The nonlinguistic plan or specification that directs realization is typically called the *message* ; some researchers talk about "*realizing the message* " and speak of the conceptual and rhetorical representations maintained by the *planning* and *goal-identification* processes as being at "*the message level* " (as opposed to *realization* activities at "*the surface-structure level* "). However, the metaphor of "*messages*" carries with it a notion of generation as a translation process where the audience reconstructs in its mind the very same expression -- the "*message* " -- as the speaker starts with. This metaphor is charged with both practical and philosophical difficulties, and has largely been abandoned in A.I. research [McDonald 1984], in favor of the view of the generation as a planning process with plan realization as its last phase. Anyway, some researchers still make use of the term "*message* " to refer to the representations that are passing over to the realization phase. Other conflicts caused by the use of coreferential terms will appear and be cleared out when the components of a generation system will have been identified and presented.

5. TEXT ORGANIZATION AND GENERATION

In this section, we present two important text generation theories and methods. The first, developed by McKeown, provides principles of discourse strategies and rules of focus that quide the generation process in its decisions about *what to say next* ; the second, developed by Mann and Thompson, is based on the interaction between speaker goals and discourse strategies, and has been an attempt to classify different discourse strategies in a manner more general than that of McKeown.

A modularized Generation System (Structure Derived from Previous Texts)

[McKeown 1985] has focused her research on issues concerning the content and organization of the generated text. She follows a rather clear division of the generation process into two main components: the **strategic component** that handles both content determination and text planning (i.e., guides the generation process in its decisions about **what to say next**), and the **tactical component** that uses a grammar and a dictionary to realize in English the message produced by the strategic component (i.e., **deciding how to say it**). Thus the approach she has taken clearly specifies how the planning of the text influences the realization of a message in natural language but not vice-versa.

In particular, the strategic component embodies both **semantic** and **structural** processes. Semantic processes determine relevancy: of all that could be said, the component must be capable of selecting that information that is relevant to a given discourse goal. Structural processes then, based on the discourse purpose and relevant information, select a *discourse* (organizational) *strategy* (or *schema*) that encodes partially ordered *rhetorical techniques* (or *predicate semantics*). This marks the beginning of **interaction** between the structural and semantic processes; here, semantics influence the structure selected for the answer. The rhetorical techniques quide the selection of propositions from the relevant information. A *focusing* mechanism monitors the use of rhetorical techniques to fully determine the content and order of the answer. It helps maintain *discourse coherency* by selecting, among the next possible propositions indicated by the discourse strategy, that proposition which ties in most appropriately with the previous discourse. Thus there is a full interaction between structural and semantic processes: the organizational strategies used in the text will affect its content, and the information that is determined to be relevant will influence the chosen organization of the text.

The output ("*message* ") of the strategic component is passed to a **dictionary**, a separate module that works as an interface between the strategic and tactical component. The dictionary translates each proposition in the message into a deep structure representation of the sentence to be generated. This involves translating the predicate into the sentence verb, and mapping the instantiated predicate arguments into the case roles of the verb. *This process entails the selection of appropriate lexical items for the instantiated arguments.*

Dictionary output is subsequently fed to the tactical component. The tactical component uses a functional grammar [Kay 1979] to translate the message into English. The grammar was designed so that it can use the focus information provided by the message to select appropriate syntactic constructions, for example, the passive construction over the active or a pronoun over a definite noun phrase.

Concluding, the main issue that stands out in this project is the **focus mechanism** that provides a basis not only for deciding what to say next when the discourse strategy allows for a choice, but also it provides information needed by the tactical component to decide how to express the text. Moreover, from the point of view of text structure, three features of McKeown´s schemas are particularly significant:

(1) The scale of schemas is the whole text. Texts are specified as whole units, whose structural posibilities are pre-specified rather than created for the individual instance. This occurs because schemas are abstracted from previous texts which are doing the same task. By studying texts from other knowledge domains, the high frequency patterns are identified and represented in the schemas.

(2) Optionality of elements is built in at the whole-text level.

(3) Order of elements is pre-specified.

Later on, [Paris and McKeown 1896] have further extended this work by examining discourse strategies for descriptions of complex physical objects.

Structure Derived from Goal Pursuit

[Mann and Moore 1981], [Mann 1982] were among the first people that did a quite important work on the computer generation of multi-paragraph English text. The system they developed however, had its serious limitations that led Mann, in particular, to reconsider the way he was facing the problem of text generation.

As a consequence, [Mann 1984] and [Mann and Thompson 1987] have been concerned with text organization (planning) by developing what they call **Rhetorical Structure Theory (RST)**, in an attempt to classify different discourse strategies in a manner more general than that of McKeown.

RST is a **structural account** of the nature of discourse, characterizing the kind of units that can make up an extended text and the relationships between them. At the same time, RST is a **functional theory**, with structural units characterized according to the role they play in the forwarding of a speaker´s goals.

Unlike McKeown's text structure that is implemented by a single schema, RST text structures are compositions of various schema applications. McKeown uses her schema for both the description of an adequate answer and text structure. RST separates these [Mann 1988]. Since Mckeown's model relies on predetermined text structures based on study of prior texts, it does not need to reason extensively about the need for particular text elements, nor about how to structure them. *Its schemas encode both task and the method for accomplishing the task.* This produces a workable but rigid text structuring method with the major advantages of being implementable and effective in its design domain.

On the other hand, RST structure is derived from **goal pursuit**, and thus it is based on the immediate needs of the text under construction, rather than past texts. For instance, a Conditional schema involves simple goal decompositions, whereas the use of others, e.g., an Evidence schema involves addition of goals. Therefore, the RST approach must reason much more about the state of the audience and the ways that text affects that state.

Finally, RST separates text structure building from other processes (e.g., processes for goal, knowledge and audience identification -- done before the building process -- as well as processes for lexical and syntactic choices, sentence scope or other grammatical realizations -- done after the building process). This has the obvious advantages of flexibility and greater generality, but produces a corresponding need to implement and coordinate a diversity of processes. For this reason, RST approach is at present still under development and its implementation depends on finding suitable reductions of it which meet the needs of particular domains. This need of reduction applies both to the structure building process and to the processes related to structure building. For better or worse, in a particular text generation application much of the reduction will be forced by the limitations of the current state of the art of knowledge representation (see section 8).

6. PLANNING AND GENERATION

Two significant works on planning and generation are presented in this section. First, Appelt developed a theory of language generation based on planning and on the necessity of reasoning about speaker´s and hearer´s beliefs when generating language. Second, Mellish and Evans present a distinct work in this area by generating natural language text directly from plans already produced by an AI planning program.

[Cohen 1978] was interested in the interaction between planning and generation. He addressed the problem of planning speech acts in response to a user´s question. Part of the problem involved determining which speech act (e.g., inform, request, etc.) was most appropriate for the response that satisfied a speaker´s goal. The implemented system, OSCAR, was capable of selecting speech acts that named the type of action to be performed, specifying the agents involved, and the propositional content of the act, but it did not produce actual English output. In later work, [Cohen 1981, 1984] proposes the use of the planning formalism for deciding upon appropiate referential descriptions, while [Allen and Perrault 1980], [Perrault and Allen 1980] and [Allen 1983, 1984] assumed that speech acts work through *intended plan recognition* rather than merely as a result of agents figuring out what other agents are trying to do. This enabled a rigorous speech act theory to be developed.

NL Generation Based on Planning (an Integrated Approach)

[Appelt 1985] extended Cohen´s ideas by examining the interaction between planning and generation at all stages of the generation process. He showed that the planning formalism could be used not only for planning speech acts but for determining the syntactic structure and lexical items of the text as well.

Thus Appelt, unlike McKeown, has not accepted a separation between the processes of *deciding what to say* and *how to say it.* Instead, these processes are tightly interleaved in the sense that segments of the utterance can be realized as enough constraints are specified by the planning process. This does not, of course, imply that modularization per se is undesirable, but that the dimension along which the modules are separated is not one of "*deciding what to say* " and "*deciding how to say it* ". Appelt´s model is a hierachical-planning system whose purpose has been to plan and produce one or more utterances that satisfy multiple goals (for example, the ability to inform, request, and flatter simultaneously). System´s behavior is therefore controlled by a goal satisfaction process, where the system has to construct a plan for satisfying one or more goals from available actions and these plans may involve interactions between physical actions (e.g., pointing) and linguistic actions (e.g., requesting, informing).

A hierachical design for an utterance planning was selected because it provides for separating the planning of domain-level goals and high-level actions and the planning of low-level linguistic actions, as well as for intermediate levels of abstraction that facilitate the integration of multiple goals into utterances.

The highest-level linguistic actions are called *illocutionary acts* , which are such speech acts as informing or requesting, and are represented at a very high level of abstraction, without any consideration given to an action´s ultimate linguistic realization. They are just actions that communicate to the hearer the speaker´s (system´s) goals or intentions. The next level consists of *surface speech-acts* , which are the abstract representations of sentences with particular syntactic structures, that is, an abstract representation of *what is to be said is* determined. One surface speech act can realize one or more illocutionary acts. The next level consists of *concept activation actions,* which involves selection of explicit descriptions that are mutually believed by both the speaker and the hearer to refer to objects in the world. Finally, at the lowest level, concept activation actions are expanded as *utterance* acts, at which point specific words and syntactic structures are chosen to realize the descriptors chosen for concept activation actions.

The generation process is then divided into two components: the **intention communication component** and the **linguistic realization component**, that are actually the components in which the *surface speech acts* are divided.

The *intention communication component* is concerned with how the intentions of the speaker get communicated to the hearer. This includes communication of the intention to refer to objects or to get the hearer to identify objects. In other words, it involves planning to get the hearer to recognize that the speaker wants to do something.

The *linguistic realization component* is concerned with taking the actions specified by the intention communication component and realizing them in a valid syntactic structure. There exists a two-way channel of communication between these two components. On the one hand, the means of communication of intention determines what linguistic structures must be chosen; on the other hand, the grammatical choices available restrict the possible means of intention communication. This channel of communication is mediated by a lexicon, a unification algorithm and a functional unification grammar, called TELEGRAM (see section 10.3).

Concluding, the task of the generation process, according to Appelt´s theory, is to reason from speaker´s goals to a plan involving both physical and linguistic actions that will satisfy these goals (*content determination* and *planning*) -- done by the *intention comminication component.* The linguistic actions are refined until an English sentence is completely specified (*realization*) -- done by the *linguistic realization component.*

Although sometimes two or three sentences can be generated at a time in order to satisfy multiple goals, the *intention communication component* does not emboby the kind of discourse-level planning and discourse strategies that are necessary to deal with the generation of a multi-sentential text in general, as the *strategic component* of McKeown's system does. On the other hand, there is rather an equivalence as regards the function of the *tactical* and *linguistic* components that McKeown and Appelt have respectively used for realization. Finally, both approaches involve a distributed control of the generation process, that is, the grammar is not used to control the generation process. Rather, other techniques (components) are used to produce partial semantic and syntactic structures, which are then processed by the grammar to fill in the remaining details.

Generating Natural Language Text from Plans

Dealing with the area of planning and generation, [Mellish and Evans 1989] addressed the problem of designing a system that accepts a plan of the sort generated by AI planning programs and produces natural language text explaining how to execute the plan, (i.e., explaining the actions to be performed and why things have to be done this way). Their objective in building this system has been to develop a clear model of a possible architecture for a language generation system that makes use of *simple* , *well-understood* , and *restricted* computational techniques.

As such, the core of their work concerns the mapping from plans to *messages* , which can be thought of as abstract descriptions of natural languages discourses. Their natural language system consists of four processing stages:

Message planning: is the interface between the generator and the outside world. At this stage the generator must decide *"what to say"* , i.e., which objects and relationships in the world are to be expressed in language and in roughly what order. The output of the message planner is an expression in a special *message language* and is called *the message* .

Message simplification: it justs "cleans up" and simplifies the resulting message by localized rewrite operations on the expression.

Compositional structure building: it deals with the *"how to say it"* question and it is a recursive descent traversal of the message, during which a simple *grammatical constraint satisfaction* system -- implemented efficiently as unification -- is used to enforce grammaticality and propagate the consequences of syntactic decisions. It initially builds a first (structural) description of the message as an intermediate representation of the text.

Linearization and output: a separate set of rules is used to produce a linear sequence of words from the structural description of the text.

The resulting system is similar to the model of [McDonald 1983] in that it is basically a direct production system that utilizes an intermediate syntactic representation and emphasizes on local processing (see section 7). However, it does not aim to produce a psychological model of their work as McDonald´s model does. Moreover, the results of their system are promising, but the texts still lack much of the smoothness of human-generated text, since the explanatory power of the system depends on what information is represented in the plan in the first place. In general, the range of possible natural language constructs may be artificially limited by the shallowness of the planner´s representations.

[Houghton 1989] is also another monograph devoted to the planning approach to language generation.

Finally, unlike the planning approach considered by the above researchers, [McDonald and Conklin 1982] looked at the problem of *deciding what to say* from a different perspective, by developing a model based on what is most relevant in a given situation.

7. NL GENERATION AS A DECISION-MAKING PROCESS

Dealing with the subject of the production of spoken text, [McDonald 1985] considered that NL generation is best characterized as a decision-making process where all decisions are indelible (they cannot be retracted once they have been made) and that only a limited part of a text is planned at a time. As such, his effort has been to produce a system that could be effective as a source of psycholinguistic hypotheses about the actual generation process that humans use (as we discussed in section 4).

He describes the generation process as consisting of an *expert system* which knows about problem solving in some domain, (but does not necessarily have to know anything about language), the *speaker component* which decides what to say about the domain, and the *linguistic component* which decides what words and syntactic structures to use. McDonald work´s addresses problems in the linguistic component, that is, he is concerned solely with *realization* (the "*how to say it* " question). His system can thus be classified as a tactical component -- according to McKeown´s terminology, since it does none of the initial planning of the content or organization of the text to be produced. Instead, his generator has been designed to produce text from a *realization specification* (1) -- plans -- developed by another program (e.g., an expert system).

As such, the linguistic component consists of three major modules: a **domain-independent interpreter (controller)**, and a **domain-dependent grammar** and **dictionary**. As described in [McDonald 1984], the generation process is based on the technique of "*description-directed control* " which is a special case of *data-directed control*, where the controlling data structure is the evolving description of the final product. In this case, there are two such descriptions: an abstract description which is the input to the system (*realization specification*), and a much more concrete, *linguistic description* that is built up from the first by successively realizing its components (the *surface structure*). **Both** are used to control the generation process, with the linguistic description being most important since it provides the primary constraints on realization and organizes the recursive decomposition of the original specification into its components so as to match the order in which the text is to be incrementally produced.

In particular, the controller treats the *realization specification* as though it were a program in a high-level language, interpreting its terms and relations as instructions which dictate the order of execution of "decision-procedures" (high-specialized programs fixed into the system) used to decide between alternative choices of linguistic realization, in other words to progressivelly refine the

(1)*Realization specification* can informally be taken to correspond to what several researchers have called the "*message* " -- the conceptual and rhetorical representations produced by the planning and goal-identification processes.

specification into a syntactic description of the text.

However, the speaker´s *realization specification* does not exhaustively determine all of the decisions that must be made. The demands of language´s grammar must be taken into account, as well as the constraints due to any text that has already been produced. This results in *embeding* the process of realizing the speaker´s specifications within *another* process -- also data-directed -- this second, *enfolding* process being responsible for ensuring that the ouput text is grammatical. The controlling data structure chosen for this second process is the *text´s own linguistic description* . This description is of course the output of the decision procedures as they realize the realization specifications. It is a kind of a tree-structured, *surface-level syntactic description* , based on the form of the text as it will actually be produced rather than as it would look at some more abstract linguistic level. The surface structure tree is created incrementally from top to bottom and from left to right as the embedded realization specifications are succesfully realized.

Part of what the surface structure does is to define the path of the traversal; it defines the linguistic "program" that will be executed in a fixed order. This is the function of the hierarchical pattern of the nodes and the sequential order of their immediate constituents. The other part of what the surface structure does is to define the "content" of this program, this being done by the grammatical labels that have been selected to annotate the parts of the tree. In general, the explanation of how a text´s surface structure is used to control its generation includes: (1) a specification of how the surface structure is traversed, (2) a specification of the kind of "modules" (programs) that exist in the system and how the terms in the surface structure map to them, and (3) a specification of the overall computational environment to which the modules have access -- can they for example examine other parts of the surface structure apart from those the controller is currently working on? Is there any independent record of past actions? Where do the modules put the results of their computations?

Language generation is then a two-phase process: The first phase expands the specification (the input "message") into an annotated tree which will eventually represent the complete surface structure of the utterance (as an *intermediate representation*). The second phase traverses the tree built by the first phase, printing words, annotating the grammatical context, recording the history of the process, and propagating grammatical constraints.

While traversing the syntactic-structure tree, the controller invokes procedurally encoded rules in the *grammar and dictionary* at appropriate times in order to determine what would be the best realization of a particular "message" element within its context. The *dictionary* uses a discrimination net to select a structure and lexical items for the elements. Both the dictionary and the grammar take into account decisions already made in the tree traversal, the previous discourse and knowledge about the audience in order to ascertain the best options in making decisions about such issues as pronominalization and in choosing among syntactic structures.

Furthermore, the "description-directed" technique imposes strong demands on the descriptive formalism (the grammar) used for representing surface structure. As a result, [McDonald and Pustejovsky 1985] found out that Tree Adjoining Grammars (TAG´s) could be perfectly incorporated into the above generation process and showed that the competence theory of TAG´s could be profitably projected to structures at the morphological level as well as at the present syntactic level (see section 10.2).

Concluding, faced with the difficulties under a "*message* "-*directed approach* of realizing conceptual representations directly as words, McDonald´s work has been important since he considered to interpose a surface structure as an intermediate level of representation between the levels of the "message" and the words of the text. This provides the capacity for producing texts whose hierarchical structures are different from that of the "message" that leads to them, the usual form of texts constructed under a "message"-driven control structure.

Other generation researchers have also elaborated a theory of surface structure as an intermediate representation -- perhaps at a lesser extent than that of McDonald; among them, we distinguish [Kempen and Hoenkamp 1982] and [Jacobs 1985b, 1986, 1987].

8. KNOWLEDGE NEEDED FOR GENERATION

In this section, we examine two main issues arised from knowledge needed for generation: the content of a knowledge base and the knowledge representation formalisms used.

The use of natural language as a communication means between humans and machines requires first af all that the computer understand the message of the user, then determine how to respond to this message, and finally formulate and produce the response in the user´s desired language. The first is the task of *language understanding* , the second of a (*nonlinguistic*) *reasoning module* (a planner or expert system), and the third of *language generation* (*realization*). In general, every system that produces a formal (knowledge) representation that reflects the analysis of a text is in the domain of *NL understanding* , while every system that produces a text from a knowledge representation is in the domain of *NL generation* . The generation process can be said to actually start with the reasoning module (program) when some event leads to a need for the program to "speak".

Knowledge representation is an important aspect in a knowledge base system. The information that is, and can be, represented in a knowledge base limits the overall semantic power of the generation system: *systems are limited in what they can say to what is included in the underlying knowledge base* . Unless the generation system includes a powerful inferencing mechanism, information that is not contained in the knowledge base cannot be expressed. Thus, in evaluating a generated text, it is important to take into account the information it has to start with.

Furthermore, the **formalism** chosen for the representation, although less clearly, limits the expressive power of a generation system. Although *form* can always be manipulated to produce the specific form necessary for the task at hand, there are situations where the manipulations required to do so are prohibitely expensive. Researchers in AI have been experimenting with the effectiveness of various formalisms including frame-based formalism (such as KL-ONE), semantic networks, first-order predicate calculus based formalisms, etc. Further development of research on issues of both **content** and **formalism** are extremely important to work done in NL generation.

Faced with the potential problems of using underlying programs (knowledge bases) built to suit independent concerns, generation researchers have adopted **various approaches:** [Appelt 1985], for instance, has developed his generator as stand-alone facility and concentrated on planning in isolation (see section 6). [Mann and Matthiessen 1985] have concentrated on studying grammar (see section 10.1). [Kukich 1984] has dedicated a great deal of her own development effort to building a task-based conceptual program on her generator and give it something substantive to talk about (see section 9.2). [McKeown 1985] has worked from an independently developed program (a database), but as this program "could not say more than it knew about", she implemented a knowledge base that provided an augmented meta-level representation of the database (including a generalization hierarchy on entities, a topic hierarchy on attributes and relationships). Moreover, she interposed some kind of independent *planning* system between the knowledge base and the realization component to even out further differences (see section 5).

It is no wonder then that the very best performances by generators may come from systems in which the generator researcher is also the person who develops the underlying program. That way one is sure that the underlying representation will offer the basis for any rhetorical attitudes or distinctions that the subject matter needs, as well as for a conceptual perspective by which to organize and order information units. Nevertheless, none of these approaches leads soon to a general-purpose generation facility that can be attached freely to any underlying program. Especially, when a system needs to express the same idea from multiple perspectives, or to present it to people with varying backgrounds or goals, a complete generation system, with its grammar, planning components and a large knowledge base, must be used.

9. THE NATURE OF A GENERATOR´S LEXICON

In this section, we discuss the nature of a generator´s lexicon and particularly, the use of a phrasal lexicon that demonstrated the ability of using specialized knowledge in generation (such as, complex, "fixed", or idiomatic terms). More specifically, three generation research projects are described: The PHRED generator of Jacobs that uses a pattern language to organize a phrasal lexicon as a shared linguistic knowledge base (section 9.1), the generation system of Kukich that organizes a phrasal lexicon as a set of production rules that integrate semantic and linguistic knowledge (section 9.2), and finally, Hovy´s generator who groups all linguistic knowledge in a phrasal lexicon as a set of syntax specialist procedures (section 9.3).

The use of a lexicon (or dictionary) during the *realization* phase of the generation process has been necessary, as we saw in the sections 5-7, for choosing an appropriate word in order to express a single term in natural language. In general, the use of particular lexical items requires decisions to be made about the best possible choice given the circumstances (e.g., who the audience is, what has already been said, etc.). Similar decisions may need to be made about the verb, which will change the deep structure of the generated sentence.

Furthermore, a *discrimination net* design, which was the formulation base of McKeown´s and McDonald´s dictionaries, invites the generation researcher to further include contextual factors like the speaker´s emotional perspective in the decisions. [Goldman 1795] was the first to use discrimination nets to determine the best words for realizing whole expressions on generation from conceptual-dependency representations [Schank 1975].

What words to associate with simple conceptual terms like "*book* " or "*reads* " is obvious; however, for more complex lexical phrases that consist of more than one concatenated or hyphenated words, or idiomatic phrases like "*kick the bucket* ", lexical choice can be more problematic. The problem is actually when these terms, which may encode entire sentences at once, have to be used in rhetorical contexts where they may need to be modified within adverbs or adjectives or elaborated by subordinated clauses.

A natural solution in this situation is to use a **phrasal lexicon** [Becker 1975]. This is another thread running through different research projects, in which the use of a phrasal lexicon has proved to be a very important tool of generation systems. In particular, the problem of lexical choice takes another dimension since complex, "fixed", or idiomatic terms, not derived compositionally the way a person could, are included in a lexicon -- a conceptual extension of a standard word-based lexicon -- as unanalized entities on the same semantic basis as words.

We are going to describe three generation research projects that employ a phrasal lexicon. This fact eliminates the distinction, made previously by McKeown and McDonald, between the contents of the lexicon and the contents of the grammar and integrates conceptual and linguistic knowledge in the sense that all entities that deal with the ordering of words and syntactic environments are incorporated into the lexicon and accessed in a unified manner.

9.1 Using a pattern language (*pattern-concept pairs*) to organize a phrasal lexicon as a shared linguistic knowledge base

One of the generation projects that made use of a phrasal lexicon has been PHRED [Jacobs 1985a], a system that served as a natural language interface to the UNIX Consultant System (UC). An unusual feature of PHRED has been the shared linguistic knowledge base with its counterpart component PHRAN, a NL analysis system [Wilensky et al. 1984] (see also section 3).

The knowledge base consists of **pattern-concept (PC)** pairs, where the *pattern* contains a linguistic structure and the *concept* its internal representation. Patterns are used to represent lexical entries, determiners and particles that refer to nothing (*general patterns*), as well as *very specific patterns* that refer to particular objects. For example, there is a specific pattern representing the phrase "the big apple" when used to refer to New York City, while this phrase can also be produced by a general pattern when used to refer to an apple.

Thus the PC pair makes it easy to encode specialized phrases and constructs to be used by the generator. It allows the generator to apply the same mechanisms to both general and specific constructions, and to choose PC pairs based on their conceptual attributes. Finally, the use of the PC pair results in **no** separation between grammatical and conceptual information: the "pattern" component of each PC pair may include conceptual information, and the properties associated with each PC pair may combine linguistic and conceptual attributes.

PHRED serves as the **linguistic component** to UC by producing natural languages responses to inquiries about the UNIX operating system. In other words, it is solely concerned with the *realization phase* of the generation process. UC passes the conceptual form of its responses, usually either questions or answers to questions, to the PHRED generator, which expresses them in the user´s language (which can be in either English or Spanish). The planning component of the system exists entirely within the task domain of UC. Independently of the language being used, the UC planner makes the choice of illocutionary act, speech act, and the "message" to be conveyed. PHRED expresses then the "message" in natural language.

The generation process that leads to the production of an utterance in PHRED is a recursive process that can be divided into three phases: **Fetching** is the retrieval of pattern-concept pairs -- associations between phrasal patterns (linguistic structures) and conceptual templates -- from the knowledge base. **Restriction** consists of validating a potential pattern-concept pair to confirm that it fulfills a given set of constraints and adding new constraints to the pattern. **Interpretation** is the generation of lexical items that match the constraints of the restricted pattern.

Anaphora are handled specially during *interpretation* . In the case of constituents for which PHRED has already produced references, the generator applies a set of heuristics that will remove the constituent entirely if it is not necessary to the utterance, pronominalize, or regenerate the entire constituent. The principal heuristics are:

(1) If the anaphoric constituent is optional, remove it from the current pattern, and

(2) pronominalize other anaphoric constituents wherever possible.

There are of course many cases in which an alternative reference would be preferable but the method used by PHRED is generally effective in producing coherent references. The heuristics lead, for example, to the production of "Mary was told by John that he wanted the book to be given to him" rather than "Mary was told by John that John wanted the book to be given to John by Mary".

Concluding, the phrasal approach to language processing realized by PHRED has proven helpful in generation as in analysis. PHRED commands the use of idioms, grammatical constructions, and canned phrases without a specialized mechanism or data structure. This is accomplished without restricting the ability of the generator to utilize more general linguistic knowledge. More specifically, PHRED served to illustrate the importance of the use of specialized constructs and the ability to use specialized knowledge in generation. Nevertheless, the knowledge used by such a program, tended to be **too** specialized. Thus, the problem was not only the lack of a clear representation; more importantly, the representation failed to support the interaction of general and specialized knowledge required for a **broadly** applicable system.

This fact led [Jacobs 1985b,1987] to redesign the linguistic knowledge representation used and replace the PHRED generator by one called KING. This system uses a specially developed knowledge representation, called ACE, which allows new information to be easily linked to existing knowledge, extending the power of the system in a way that is far easier than it was in PHRED. In general, the development of extensible and adaptable natural language systems depends on a knowledge representation framework within which generalizations are effectively exploited (see section 8).

Finally, the idea of a shared knowledge base (ACE) between the generator and a variety of other mechanisms (such as an analyzer) is still maintained in KING, as it was in PHRED, and constitutes an important feature of the system.

9.2 Organizing a phrasal lexicon as a set of production rules that integrate semantic and linguistic knowledge.

In contrast to McKeown´s, Appelt´s, Mann´s and McDonald´s approaches which concentrated on the task of identifying and examining in great detail a small set of knowledge processes which underlie text production, [Kukich 1984] focused on a larger set of processes in less detail. This made it possible to address another task in NL generation, that of understanding the interactions among knowledge processes that give rise to fluent text. She argues that the source of the difficulty in obtaining fluent texts is the **knowledge-intensive nature of language generation:** not only are a variety of types of knowledge required, such as **semantic** and **linguistic** (lexical, syntactic, grammar and discourse) knowledge, but those various forms of knowledge are used in a highly integrated faschion during the generation of text.

In particular, she argues that fluency is the result of the effective integration and control of some subset of knowledge processing skills, including *semantic skills* (e.g., interesting message choice), *lexical skills* (e.g., lexical variety), *syntactic skills* (e.g., appropriate syntactic choices), *grammar skills* (e.g., appropriate use of conjuctions), and *discourse skills* (e.g., appropriate use of pronouns, of ellipsis, etc.) -- where discourse knowledge includes all those processes involved in managing a body of text longer than a single sentence or clause. A sufficient subset of these various knowledge processing skills must be succesfully identified, implemented and integrated in order for fluent text to be generated; otherwise, fluency defects will be exhibited as a result of incomplete or poorly integrated knowledge processes.

The following text demonstrates both system´s ability to use pronouns appropriately and to elide a subject entirely when discourse structure warrants it (*discourse skills*):

Anaphora and Ellipsis Usage :

after the *stock market* crept upward early in the session yesterday, *it* slid downhill late in the day and *posted* a small loss. stock prices turned in a mixed showing in moderate trading.

The first clause, a subordinate sentence, calls for a subject from the subject class MKT (market), and the generator selects the phrase " the stock market". The generator recognizes that the second clause, a simple sentence, also requires a subject from the same class, and chooses to use the **pronoun** "it", which agrees with the subject it replaces in number. On recognizing that the third clause requires yet another subject from the same class, the generator **elides** the subject completely.

Generation System Description

Kukich divides the generation process into four modules: (1) a **fact generator** receives data from a database for the day´s stock market behavior and its only function is to perform arithmetic computations to transform the data into OPS5 format. A stream of facts is produced as output.

(2) a **message generator** takes these facts as input and performs the function of infering interesting messages from the facts. A stream of messages is produced as output -- *content determination* . It is in this module where *semantic knowledge processes* , such as inference rules, are used to create and instantiate semantic messages from the facts in the database.

(3) a **discourse organizer** takes the messages as input and performs the function of grouping messages according to topic, and combining and elininating messages where appropriate. A stream of ordered messages is produced as output -- *text planning* . Again, *semantic knowledge processes* , such as discourse organization rules are used to combine and order messages. The messages conveyed in the text are called *semantic knowledge constructs* .

(4) a **surface generator** takes the ordered messages as input and produces the final report as output -- *realization* . It does this by mapping messages into phrases in its **phrasal lexicon**, building clauses from phrases, and combining clauses into complex sentences according to its *clause-combining grammar* . Important *linguistic knowledge processes* , such as grammar rules, are used to select phrases for expressing messages, to select syntactic forms for shaping phrases into clauses, and to combine clauses into complex sentences in accordance with discourse guidelines, such as variation in sentence length, invariance of verb tense, and use of parallelism, synonyms, hyponyms, pronouns, and ellipsis. As a consequence, a variety of linguistic knowledge processes are integrated in the grammar rules. Moreover, important *linguistic knowledge constructs* include lexical phrases used to convey messages and syntactic forms used to shape phrases into clauses.

The modules operate in a linear, sequential fashion for the sake of computational manageability. However, the grammatical knowledge required to produce fluent natural language report is extremely situation-dependent. That is, at any decision point in the generation of text, the choice of a valid and appropriate grammatical item, such as a word, a phrase, or a syntactic form, is a function of a well- integrated set of rules embodying a variety of semantic and linguistic knowledge, including knowledge of the **lexicon**, syntax, discourse structure, rhetoric principles, and semantics, as well as other types of knowledge. In particular, there are specific production rules encoded in the clause-combining grammar which are used to select appropriate lexical phrases that match the semantic attributes of the message.

Kukich always works with large phrasal units, never assembling NPs or PPs from smaller concepts; that is, rather than dealing with semantic primitives, the approach she takes is to work with *whole semantic messages* that match whole lexical units. In the case of the stock market report these include: Closing market status messages, volume of trading messages, etc., paired directly with the stereotypical phrases of such announcements: "stock prices turned in a mixed showing", "in heavy trading", etc. Moreover, objects, actions, and time points are mapped directly into the appropriate word strings: "Wall Street securities market", " <be> pushed downhill", "late in the day". The compositional template driving the assembly of these phrases into a text was based on clauses, such as simple sentences, coordinate sentences, infinite clauses, adverbs, punctuation, etc., built out of the SUBJ-VERB-ADVP phrase: <market> <action> <time point>. The clauses are then combined by the clause-combining grammar rules to form compound-complex sentences.

9.3 Grouping all linguistic knowledge in a phrasal lexicon as a set of syntax specialist procedures

A quite distinct and interesting approach to NL generation has been taken by [Hovy 1988] who presents a model of generating text with the exclusive use of a phrasal lexicon. He particularly focuses on the form that a generator's linguistic knowledge should take. Applying a generation perspective to the customary breakdown into morphology, syntax, lexicon, and semantics, Hovy finds no sharp demarcations between them. He argues that in a functionally organized system, there is no reason why a priori distinction should be made between the contents of the lexicon and the contents of the grammar. From the generator's perspective, the difference between these sources is not important, so all linguistic knowledge -- all language -- should be contained in the lexicon.

In particular, he assumes that the generator's input is a list of structures built using a representation scheme (similar to Conceptual Dependency notation [Schank 1975]). These structures have been assembled by a *text planner* that plays a further role in the realization process only to guide decisions, as stated below.

There are three types of decisions involved in the generation process:

(1) **inclusion:** select which portions of the input to consider further (which portions will eventually appear in the text; not all the input need appear) -- *content determination* .

(2) **ordering:** select the order in which to consider them (in which order they will appear in the text) -- *text organization* .

(3) **casting:** select a syntactic class or type for each portion, that is, the form of its textual realization (how they will eventually appear in the text) -- *realization* .

The criteria for determining *inclusion* and *ordering* decisions reside in the *text planner* , which is assumed to have access to the speaker's goals with respect to the hearer, and to some characterization of the conversational situation and of the hearer [Hovy 1985,1986,1987a,1987b].

Furthermore, the form of each generated sentence is determined by the sequence of *inclusion*, *ordering* and *casting* decisions made in the realization process. At any point in the process, the generator needs information on which linguistic options exist -- which decision tasks it must/may perform on the current input.

This information is provided by the rules of grammar of the language, by the fixed and semi-fixed phrases of the language, and by idiosyncratic constraints associated with certain words in the language. Since, to a generator, no sharp line of demarcation exists, there is **no** forcing reason why some of this information should be contained in a body of rules called the grammar, and the rest of it in a collection of words (and even phrases) called the lexicon. Thus it makes sense to collect all information concerning the decisions in the lexicon. The lexicon should be then the sole repository of the patterns that make up language -- some very specific, some very general. This homogeniety enables the generator builder to add new forms of expression -- words, phrases or rules of grammar -- with ease.

Furthermore, just as it makes sense to associate information about idiosyncratic syntactic phenomena with the words, it makes sense to associate the pragmatic decisions with words and other lexicon elements as well. All this information can be considered and implemented as specialist functions, called *syntax specialists* , which determine the applicability of each phrase in context. Thus *syntax specialists* **control** the operation of the three types of decision. Sometimes the specialists are very simple -- so simple that they contain no procedural information -- and then they are implemented as patterns (phrasal templates). Alternatively, they may be quite complex -- prescribing a series of actions and tests to perform the three tasks (inclusion, ordering and casting) -- and then they are implemented as procedures.

Thus, the generation process (that is, its *realization component*) starts with the syntax goal to produce sentence(s) from the input topic. Initial syntax goals are produced by the planner. After that, each syntax goal (the goal to create some syntactic constituent) is achieved by a procedure, the *specialist* , that accepts a piece of input, performs the three tasks, and produces an ordered list of words and/or other syntactic goals, each associating another specialist with the piece of the input.

Hence, *inclusion* decisions are reflected on the choice of topic; *casting* decisions in the choice of specialist; and *ordering* decisions in the order of syntax goals.

The generation process applies specialist procedures to progressively "smaller" pieces of the input, and eventually arrives at words, which it outputs. In doing so, it implicitly creates, in depth-first fashion, the syntax tree of the sentence. Each node in the tree corresponds to a point at which a specialist was applied. Moreover, each specialist must know of the different ways its goal can be achieved, and must be able to select an appropriate alternative. The criteria by which these decisions are made are syntactic, semantic and pragmatic.

10. THE ROLE OF GRAMMAR IN GENERATION

In this section, we discuss the important role that grammars play in generation. In particular, in section 10.1, we examine two grammatical formalisms: the ATN and Systemic grammars that have been used to thoroughly control the generation process. Then, in section 10.2, we discuss how the use of the TAG grammars succeeds in specifying all the valid surface structures that texts can have, and consequently representing a surface structure as an intermediate level of representation between the levels of the "message" and the words of the text. Finally, in section 10.3, we present the FUG formalism that is used to give a generator a modular, independent way of supplying the purely linguistic information that the process must have and do so without imposing specific demands on its control structure.

Usually it is only the *linguistic component* of the generation process that has any direct knowledge of the grammar of the language being produced. What form this grammar takes is one of the points of greatest difference between generation projects (being a separate module from the lexicon -- see sections 5-7, or part of a phrasal lexicon -- see section 9). Anyway, all projects largely agree on **the function a grammar** should serve in generation: *to quide and constrain the process of generating a text with a specific content and goals in the presence of a specific audience, as well as to ensure that the texts that are produced follow the rules of the language -- that they are "grammatical "*.

The function (or role) of grammar has a significant effect on the form it takes. More importantly, it also strongly influences the information it must include: The grammar is now responsible for defining the choices that a language allows in form and vocabulary, and it must further include criteria of usage; that is, in what circumstances does one decide on one alternative rather than another, as well as what functions the various constructs of the language serve that make them appropriate for fulfilling a certain goal? Only by including such information can a grammar serve as a resource defining the options available to the text planner.

Furthermore, as we mentioned at the end of section 4, there has always been a binding between the choice of a formalism for representing the language´s grammar and the choice of the control protocol of the generation process. Depending on whether a partial or total amount of control is given to a grammar, there are three basic approaches that can be identified:

(1) Giving the grammar a complete control of the whole generation process as soon as a text plan has been constructed. The formalism representing this kind of grammar could be a traversable graph structure such as an ATN or a systemic grammar (**centralized approach**).

(2) Using the grammar to specify all the valid surface structures that texts can have, therefore providing an explicit , examinable representation of the grammatical context in which an element will appear, and thus make it available to constrain the choices open to realization and the text planner (**message-driven approaches, TAG grammars**).

(3) Stating the grammar as an independent body of statements and using it to fill in the remaining details once partial semantic and syntactic structures are produced by other generator´s modules. Such a grammar is the functional unification grammar (**distributed approach**).

We have mentioned in section 3 the attempts that have been made to share as much of the grammatical information as possible between both generation and understanding processes. In the following sections we are going to describe such "reversible" grammatical formalisms, i.e., grammars able to serve equally well as a controlling description in both generation and understanding. Another kind of grammar, a *discourse grammar* , is also mentioned in section 11.

10.1 Using ATN or Systemic grammars to thoroughly control the generation process

A first approach taken by several researchers involved a very general notion of grammar and used the grammar to drive the generation process. Early systems that relied on explicit grammars often used an **ATN** to drive the generation of text as in [Simmons and Slocum 1972], [Goldman 1975] and later on [Shapiro 1982] whose generator is the most elaborated of the group. All these systems have a similar design. They generated single sentences, whose content had already been specified in a semantic network formalism, and not texts. So, they only dealt with the realization component of the generation process by using an ATN formalism to handle syntactic procedures. Goldman, in particular, concentrated his research effort on the problem of lexical choice by developing a process to select particular words and idioms for a sentence.

The problem with using this approach to drive the entire generation process is that it forces a certain ordering of decisions about the surface structure of the sentence, that must be followed by the system. For example, the active-passive decision must be made at the start, even though more important decisions regarding the overall form of the content and the lexical choices might seem more appropriate to consider first. Thus a syntax-driven generator might be awkward to design or would involve considerable backtracking to redesign the sentence structure when new constraints are found.

Other researchers, however, have used another type of grammar, the **systemic grammar** [Halliday 1976, 1985], [Patten and Ritchie 1986] that proved to be much better suited for controlling the generation process. It has been succesfully employed in the realization component of the generation system of [Mann 1983a, 1983b]. Mann divided the generation process into four modules: the *acquisition module* that is used for content determination: it acquires from an external system a body of information that is to be communicated, the *text planner module* organizes a discourse -- both modules correspond to what McKeown has called the *strategic component* -- the *sentence generation module* generates the text -- this is equivalent to the *tactical* or *linguistic component* that performs realization. Finally, Mann incorporates a separate module, called the *improvement module* whose task is to criticize and improve the text after it is produced.

Most of the work has been concentrated on the *sentence generation module* and its associated grammar, NIGEL [Mann and Matthiessen 1985]. The NIGEL grammar is an extensive systemic grammar organized as sets of loosely related choice systems, where each choice adds some constraint to the final sentence form, giving constraints the opportunity to propagate and influence later decisions; however, the final structure of the sentence is determined only by the combination of all the different choices made.

Thus the decisions about the mood of the sentence (e.g., indicative or imperative) can be made either before or after decisions about what the new content of the sentence will be, or decisions about what objects should be in focus. The generation process can then be driven by these choice systems: It simply picks one of the systems and then proceeds to call whatever modules in the system can best make the decision. As a consequence, there is no a fixed order in which the modules of the generation process are executed. Rather, they are called as needed to make a decision (choice). The power of this framework is that the grammar translates decisions that a module makes into constraints of the syntactic form of the sentence. The module itself need have no knowledge of the syntactic consequences of its decisions. Mann´s system will finally generate a coherent multisentential text.

10.2 Applying the theory of TAG´s to NL generation

Tree Adjoining Grammars or TAG´s [Joshi 1985], [Kroch and Joshi 1985], [Vilay-Shankar and Joshi 1985] were developed as an alternative to the standard syntactic formalisms that are used in theoretical analyses of language. They are attractive because they may provide just the aspects of context sensitive expressive power that actually appear in human languages while otherwise remaining context free.

A tree adjoining grammar is defined as a set of **elementary** phrase structure trees, a **linking** relation that can be used to define grammatical dependency relations between two nodes within an elementary tree, such as subcategorization relationships or filler-gap dependencies, and an **adjoining** operation that combines trees under specifiable constraints. The elementary trees are divided into two sets: **initial** and **auxiliary**. *Initial trees* have only terminals at their leaves. *Auxiliary trees* are distinguished by having one non-terminal among their leaves; the category of this node must be the same as the category of the root. All elementary trees are "minimal" in the sense that they do not recurse on any non-terminal.

Sentence derivations start with an initial tree T, and continue via the *adjoining* of an arbitrary number of auxiliary trees A that are "spliced" into the initial tree T by displacing the subtree of T at the point of the *adjunction* up to the frontier of A. For example, we would take the initial tree:

$[_{S'}$ Who$_i$ does $[_S$ John like e_i]] (the subscript "i" indicates that the "who" and the trace "*e*" are linked) and adjoin to it the auxiliary tree:

$[_S$ Bill believes $_S]$ to produce the *derived* tree:

$[_{S'}$ Who$_i$ does $[_S$ Bill believe $[_S$ John likes e_i]]].

Adjunction may be "constrained". The grammar writer may specify which specific trees may be adjoined to a given node in an elementary tree; if no specification is given the default is that there is no constraint and that any auxiliary tree may be adjoined to the node.

As such, [McDonald and Pustejovsky 1985] applied the theory of TAG´s to NL generation. They have been attracted to TAG´s because TAG´s central operation -- the *adjoining* operation -- corresponds directly to certain central operations of their own performance-oriented theory. In particular, the surface structure trees of TAG´s correspond to the surface structure trees that McDonald used as an intermediate level of representation between the levels of the "message" and the words of the text (see section 7). Though they employ a non-conventional tree notation, the constituency pattern of the tree is used directly as the specification of a path -- top-down and left to right through the tree -- that controls the sequence and environment of realization and the order in which words appear. This representation is called *path notation* and formally the surface structure is not a tree but a uni-directional linked list whose formation rules obey the axioms of a tree.

The path consists of a stream of entities representing *phrasal nodes* (e.g., S, NP, VP, etc.), *constituent positions* (e.g., SENTENCE, SUBJECT, PREDICATE, etc.), *instances of information units* (e.g., <hit-by-missiles...>), *instances of words* (e.g., two, oil , tanker, etc.), and *activated attachment points* (labeled circles under constituent positions).

The attachment points correspond to the constraints on the *adjunction* operation of TAG´s, but rather than being simple constraints are actual objects interposed in the path notation of the surface structure. A list of the attachment points active at any moment is maintained by the *attachment process* (which corresponds to the *adjunction* operation of TAG´s), and is consulted whenever an *information unit* needs to be added. Most information units could be attached at any of several attachment points, with the decision being made on the basis of what would be most consistent with the desired prose style. When one of the points is selected it is instantiated, usually "splicing in" new surface structure in the process, and the new information unit is added at a designated position within the new structure.

Furthermore, an attachment point goes with any "choice" (elementary tree) that includes the labeled position of the attachment point. Thus McDonald´s treatment of attachment uses two structures, an attachment point and a choice, whereas a TAG would only use one structure, an auxiliary tree. This is because a TAG is a formalism that only needs to specify the syntactic structure of the grammatical strings of a language, while McDonald's generation model has to show explicitly how conceptual information units are represented into texts as part of a psycholinguistically plausible process. *This suggests that the competence theory of TAG's can be profitably projected to structures at the morphological level as well as the present syntactic level*.

Finally, the attachment (or "splicing") operation inherits a natural set of constraints on the kinds of distortions that it can perform, since by the *indelibility* property of previously made decisions, existing position sequences can not be destroyed or rearranged. However, although it seems that these constraints will turn out to be formally the same as those of a TAG, there has not yet been carried out the detailed analysis to confirm this.

10.3 Obtaining a distributed control of the generation process by using a FUG grammar

Functional Unification Grammars or FUG's [Kay 1984, 1985] have been shown to give a generator a modular, independent way of supplying the purely linguistic information that the process must have and do so without imposing specific demands on its control structure. [Ritchie 1986] has shown, however, that this generality of the FUG notation may give undesirable computational complexity properties: *generating a structure from an arbitrary FUG appears to be NP-complete*. Thus, implementers of FUG generators must be especially careful in the construction of their algorithms, since the formalism itself is not efficient.

[Appelt 1985] and [McKeown 1985] have developed and employed a FUG grammar in their generators extending it just by adding terms like "subject", "premodifier", or "head" -- descriptions of the role a constituent plays within the category that dominates it. Moreover, [Lancel et al. 1986] have used the FUG formalism as a common grammar suited to both parsing and generation in their "multilingual document generation" project.

FUG grammar allows descriptions of linguistic structures to be expressed as **functional descriptions (FDs)**, which are set of *features* **(attribute-value pairs)**, and grammatical derivation is expressed in terms of these structures. *Attributes* specify categories (syntactic, functional, or semantic) and *values* specify the legal fillers of those categories which may be either symbols or sub-grammars (other attribute-value pairs).

Within a level of an FD, each feature-name can appear only once; i. e., no feature can appear with two different values. However, FDs may contain **alternatives** which specify that a particular feature may be formed in more than one way. **Patterns** are used within an FD to specify the surface order of the constituents in the resulting string. A pattern is a list of attribute names which specifies the left to right order of the constituents. Feature-values written in angle brackets (e.g., <DEFINITE>) are not simple data-values, but are pointers to other positions within the structure.

They are called **paths** and indicate a structural position that can be found by starting at the outmost level of nesting and tracing feature names inward along the path. The sample path <a_1 a_2 ... a_{n-1} a_n> points to the value of attribute a_n in the value of attribute a_{n-1}, and so forth.

An FUG grammar is used by the tactical component of McKeown which takes as input a deep structure representation of the "message" and uses the grammar to translate the "message" into English. FUG was selected because of the ability to encode directly in the grammar tests on focus and theme -- information provided by the strategic component, for selecting pronominalization, the passive consrtuction over the active, and there-insertion. The FUG formalism is thus perfectly suited to the generation task as it allows the input representation (in the strategic component) to leave many details of the actual surface structure unspecified. These can then be filled in by the grammar.

In the FUG system, both the input and the grammar are represented in the same formalism, each as a list of attribute-value pairs. Attribute-value pairs represent the constituents and features of the desired sentence or grammar. To generate a sentence, the input, which represents the deep structure of the sentence, is **unified** with the grammar. During this process, the actual surface structure of the sentence is determined. Syntactic structures that are missing in the input representation are filled in by the grammar. The final sentence is produced by linearizing the result of the unification process.

More specifically, during the unification process, variables (values of **any**) are replaced by values from the input and alternatives in the grammar are eliminated. The process involves unifying the value of each attribute in the *grammar FD* with the value of the attribute of the same name in the *input FD* . If the grammar value is an alternative, all options are unified and the first succesful result is taken. If either value is a symbol, then unification succeeds when the two values are equal, when one value is a wild card (**any**) and the other value is non-null, or when either of the values is nil. If both values are FDs then the two FDs are unified. If an attribute occuring in the grammar does not occur in the input, the attribute and its value are added to the result. In all other cases, unification results in failure. After the input FD is unified with the grammar, each constituent of the FD is unified with the appropriate sub-grammar. Unification is a fully recursive non-deterministic process.

The major **disadvantages** of FUG involve implementation issues. Since the unifier is non-deterministic, the production of a single sentence from the given deep structure is incredibly time-consuming. In addition to it, debugging the grammar during its design stages is made more difficult by the fact that an error does not show up until far from the place where it first caused the problem. This situation is common to many non-deterministic processes and could be improved through the incorporation of testing diagnostics in the unifier.

On the other hand, FUG has significant **advantages:**

(1) It allows for equal treatment of functional (protagonist,goal) and syntactic catagories: this means that the input to the tactical component can be formulated by a process (a text planner) having little information about syntax. For instance, the dictionary designer could easily specify the protagonist (agent) and goal (object) of a predicate, but more detailed syntactic information, such as syntactic category and surface position, can be deduced by the tactical component.

(2) Augmenting the grammar also turns out to be a fairly easy task in FUG. When a new type of syntactic category is needed, another alternative can be added to the grammar specifying the structure. Only the alternatives which encode the syntactic categories of which the new category is a constituent need to be modified. All other alternatives remain unchanged. The use of alternatives results in a clearly modular design for the grammar.

(3) Incorporation of the use of focus information into the grammar is another favorable aspect of it. The grammar allows for easy incorporation of tests on focus to select an appropriate syntactic construction (passive-active construction, pronominalization, etc.).

As a consequence, the advantages are more worth than the disadvantages which could, in fact, be minimized through an increased implementation effort. For this reason, the FUG formalism can be perfectly suited to the generation task.

Finally, Appelt considered the fact that a straightforward adoption of an FUG grammar as a representation of linguistic knowledge, encourages one to make a strong separation between the planning and grammatical processes -- the "*what-how* " distinction that he has been trying to avoid. Thus, he developed an FUG grammar, called TELEGRAM [Appelt 1983], that facilitated the sharing of control and information between the planning process and the unification process. The central idea is to associate a minimal FD with the surface speech act as soon as it is inserted into the plan. This minimal FD is then unified with the grammar by means of a specific unification algorithm that is capable of recognizing when it has insufficient information and invoking the planner to provide more complete information at the right time. The final result of the TELEGRAM unification is the same as if an ordinary unification algorithm had been used to unify the complete functional description with the grammar.

An additional important benefit of integrating planning and linguistic reasoning is the ability to coordinate physical intention communication actions with the utterance: When the planner is reasoning about how to best comminicate the speaker´s intention to refer to something, it is not constrained to plan only linguistic actions, such as DESCRIBING, but it can make any modification to the plan that it considers advisable, including the planning of physical actions, such as POINTING.

As a consequence, by postponing some of the planning and placing it under control of the unification process, the system maintains its capacity for efficient hierarchical planning while enhancing its ability to coordinate physical and linguistic actions.

11. NL GENERATION IN ROMANCE LANGUAGES

In closing the presentation of the various approaches of later research in NL generation, we discuss issues in generation of texts in Romance languages [Danlos 1989]. Thus, taking into account the rich inflectional system of Romance languages -- an issue which has not been seriously considered in English -- Danlos had to specify the way of integrating conceptual, linguistic, syntactic and morphological decisions. She did that by developing a "discourse grammar", and by using a "lexicon grammar" as well as syntactic and morphological operations in an interacting way.

According to Danlos, the generation process is divided into two main components: a **(non-linguistic) reasoning component** (planning or expert system) deals with the "*what to say*" question and returns a semantic representation of the information that is to be conveyed to the user; this semantic representation is translated into a text by a **linguistic generation component** which deals with the "*how to say it*" question. Danlos has been concerned only with the linguistic generation component. Thus, based on a specific domain, that of terrorist crimes as reported in newspapers, she intended to produce well-constructed texts -- syntactically coherent and of good legibility both from the point of view of clarity and style -- and not the kind of texts produced while speaking.

Decisions Taken by the Generation Process

The ordering of the decisions that have to be made in the linguistic component, is one of the major issues, because most of the decisions are dependent one upon each other. We shall examine these decisions and their interactions, as well as a further decomposition of the linguistic component into two, though not quite independent, components that are concerned with these decisions.

In particular, starting from a semantic representation, the generation process needs to take the following decisions:

conceptual decisions: Which is the information that should appear in the text? Which is the order it should appear? Which of this information should be expressed explicitly and which could be left implicit?

linguistic decisions: How to perform a linearization of the text into sentences (namely, the choice between juxtaposition, subordination, relativization or coordination as a sentence linearization procedure)? What syntactic constructions to choose? What words to choose?

stylistic decisions: When the chronological order is adopted to tell the events of a story, this order has to be respected all through the story. The chaining of subjects that have the same referent should not only take into account the subjects of the preceding sentences, but also those of the following ones. Moreover, these decisions are global and not local (they refer to the text as a whole).

In addition to these decisions, the linguistic component has to perform the following operations:

syntactic operations: Form sentences that obey to grammatucal rules of the object language (good formation of the sentence and its components, subject-verb agreement, placement of negation, formation of pronouns, etc.).

morphological operations: conjugation of verbs, computation of the morphological antecedent of a pronoun (e.g., the pronoun "he", its morphological antecedents are a masculine singular noun).

Using a "Discourse Grammar to Integrate Conceptual and Linguistic Decisions

It has been shown in [Danlos 1985, 1987a] that the conceptual and linguistic decisions are operations that depend on each other. In particular, by examining *direct causal relations* (phrases describing an action and the direct result of that action), we see that the criteria on the order of information, the linearization into sentences and the syntactic constructions offer a combination of 36 cases over which only 15 are realizable. This result is not only specific to direct causal relations: for every type of semantic relations (indirect causal relations, potential causal relations, logical implications, dictionary definitions, etc.) one should expect that these criteria offer a combination of structures over which only certain structures are formally realizable and semantically appropriate. As a consequence, the decisions with regard to these criteria should not be taken independently the ones from the others.

Furthermore, the 15 discourse structures, that enable to express a direct causal relation, constitute a linguistic data base of a new type that is called a **"discourse grammar"**. A discourse grammar establishes a mapping between a type of semantic relation (i.e., a causal relation) and the list of discourse structures that enable it to be expressed. Other discourse grammars could be also constructed for the other semantic relations. This notion of the discourse grammar presents the originality of integrating the decisions on (1) the ordering of information, (2) the linearization into sentences, and (3) the form of the sentences (namely, the choice between the active or passive syntactic construction). In other words, the discourse grammar integrates conceptual and linguistic decisions and it is essential in both generation and interpretation. Moreover, it seems to be the only way to establish a correspondance between meaning and form that guarantees the production of texts which are semantically and grammatically appropriate.

Deciding for an Appropriate Lexical Choice by Using a "Lexicon-Grammar"

We have seen that all conceptual and linguistic decisions, except the lexical choice, require the use of a discourse grammar. On the other hand, the selection of a term could be limited in the search of an element into a lexicon suitable for the domain under consideration. For instance, in a terrorist crime context, the semantic factors relevant for choosing a term to express the result of an attempt, i.e., a STATE which concerns an OBJECT because of an ACT done by an ACTOR, are the particular STATE (e.g., dead or wounded), OBJECT (e.g., famous or not), ACT (e.g., strangling or throat cutting) and ACTOR (e.g., specified or not) involved, in addition to the context in which this result occurs. This situation is not peculiar to the choice of a term to express a RESULT. It seems likely that, for each kind of concept, all the semantic roles concerned with that concept may be factors in choosing the best terms to express it.

The lexicon that is used to decide for an appropriate lexical choice is called **"lexicon-grammar"** [Gross 1986]. The lexicon-grammar contains terms describing the principal concepts of the domain (such as the act and the result of an attempt) with their conditions of use. These conditions involve factors dispersed in the semantic representation such as those described above. Moreover, the lexicon-grammar describes the syntactical properties of terms which indicate whether the terms can undergo such modifications as the formation of the passive or the subject distribution (e.g., the verb "to assassinate" can be constructed only with a human subject, unlike "to kill" that can take either an animate or non-animate subject). For each verb, a lexicon-grammar records all its syntactic properties, among them those concerning pronominalization.

Interdependencies Between Lexical, Conceptual, and Stylistic Decisions

We have seen that we can select an appropriate discourse structure from the discourse grammar based on the integrated conceptual and linguistic decisions described above. However, a discourse structure does not generate a correct text if it is not realized with terms that present the required syntactical properties. For example, for a direct causal relation, a structure where the act is in passive requires that the term that expresses the act permits the formation of passive. Thus, the operations of lexical choice and discourse structure selection are dependent one upon each other: If one selects first a discourse structure, one must choose terms in a way that they present the syntactical properties required by the choosen discourse structure; if one selects first the terms, the authorized discourse strucures depend on the syntactical properties of the choosen terms.

Moreover, certain stylistic factors (e.g., the consideration of the chronological order) could also affect the selection of a discourse structure, if we want the text to be of an appropriate style. Thus, we come to a first conclusion:

- the selection of terms
- the eventual contribution of a modification to these terms, for example, the passive transformation (syntactic choice)
- the determination of the order of the terms
- the linearization of the terms (juxtaposition or combination in a complete sentence)

are operations dependent one upon each other. This result is fundamental in generation because it has an immediate consequence: ordering these decisions results in giving them a priority order. However, taking into consideration the interdependencies between the lexical, conceptual and stylistic decisions, the discourse structure of the text to be generated should result (be selected) from the intersection of all the three.

Components of the Linguistic Generation Module (an Integrated Approach)

Taking into account the interactions between conceptual and linguistic decisions, Danlos has further divided the linguistic component into a "**strategic component**" and a "**syntactic component**".

The *strategic component* makes the conceptual and linguistic decisions simultaneously, and thus it relies upon two linguistic bases, the lexicon-grammar and the discourse grammar, as we presented above. Here, we note that the term "*strategic component* " is defined and used in a distinct way from the one of [McKeown 1985]. McKeown has defined and implemented a *strategic component* to deal with the content determination and text planning issues, that is the *"what to say "* question. In particular, she used a kind of "discourse grammar" (*discourse strategies*) to encode text organization (text planning); therefore, her notion of discourse grammar gives only quidelines on the order of information, thus is used only for making conceptual decisions. Instead, she considered that the options over syntactic constructions and linearization into sentences (linguistic decisions) should be based on focus rules and both these decisions are determined by the *tactical component* (which treats the *"how to say it"* question). In other words, she established a kind of modularity in which the ordering of information is determined independently of any linguistic consideration; on the contrary, in the *strategic component* of Danlos, the conceptual (text organization) and linguistic decisions depend on each other, while a discourse grammar, instead of focus, is used for considering all the three decisions (ordering, linearization and form of the sentence) together.

As a consequence, Danlos considers the organization(planning) of the text within the scope of the *"how to say it"* question by thus leaving to an external reasoning component only the task of content determination. Furthermore, she does not assume a separation between conceptual (*planning*) and linguistic (*realization*) activities. Both are integrated in a strategic component which in turn is defined and implemented during the realization phase of the generation process.

After selecting an appropriate discourse structure and the lexical terms that express the principal concepts, the stategic component combines the results of these two operations in order to construct a *"text schema"* (or *"text template"* [Danlos 1987b]). Text templates are synthesized into texts by the *syntactic component* [Danlos 1986]. On the one hand, the syntactic component performs *syntactic operations*, such as the coordination, the pronominalization, the subject-verb agreement, or the reduction of a sentential clause into an infinitive clause. On the other hand, it performs *morphological operations*, such as the conjugation of verbs, etc.

A Complete Interaction Among All Levels of the Generation Process

[Danlos and Namer 1988] have further shown that, within the syntactic component, pronominalization involves the morphological level for the generation systems that produce texts in Romance languages. For example, the inflected forms of adjectives or past participles, which is morphological information, has to be taken into account when determining whether a term can be synthesized as a personal pronoun. Thus, the morphological level (level supposedly very "low") must not be taken into account only at the very last stage of the generation process.

Furthermore, [Danlos 1989] shows that the initial modularization of the linguistic component into two components -- a strategic component and a syntactic component, the latter handling pronominalization -- is even too modular: some pronominalization questions should be taken into account while the conceptual or linguistic decisions are made.

To show this, we recall that the passive transformation (which is actually a syntactic transformation) is decided in the strategic component, since a discourse structure indicates whether a sentence must be in the active or in the passive. This is justified by the fact that the choice between active and passive is relevant to the discourse semantics [Danlos 1987a]. On the other hand, if we consider a sentence that has a predicate as a subject, for example, the French sentence: ***Ugo boit du vin. Cela*** plaît à Marie (***Ugo drinks wine. It*** pleases Mary). Then if the [subject] predicate is not synthesized as a personal pronoun, it is synthesized as a sentential clause (e.g., * ***That Ugo drinks wine*** pleases Mary). Since this form is stylistically unpleasant, we apply the subject extraposition transformation to improve it (i.e., It pleases Mary ***that Ugo drinks wine***). However, this decision must be made after the pronominalization decisions, because subject extraposition (SE) **cannot** take place when the subject is pronominalized (i.e., It pleases Mary -SE-> *It pleases Mary ***it***).

Moreover, subject extraposition does not apply to all French verbs; roughly it doed not apply to transitive verbs: *Qu' Ugo boive du vin* ravit Marie ---> * Il ravit Marie *qu' Ugo boive du vin* (*That Ugo drinks wine* delights Mary ---> It delights Mary *that Ugo drinks wine*). In such a case we can apply the passive transformation to improve this form, e.g., Marie est ravie de ce qu' Ugo boive du vin (Mary is delighted by Ugo drinking wine).

Hence, we have the following rule: if a [subject] predicate is not synthesized as a personal pronoun, synthesize it as a sentential clause and apply subject extraposition if possible, if not apply the passive transformation.

As we have seen above, such a rule (which includes the passive transformation) can take place only **after** the pronominalization decision, that is, in the syntactic component. However, we have seen that the passive transformation should be decided in the strategic component. Hence, a contradiction which shows that there is a total interaction between the conceptual, linguistic, syntactic and morphological levels of the generation process. An English syntactic component covering some of these phenomena has been implemented in a declarative formalism using functional descriptions [Lancel et al. 1988]. However, a generation algorithm which would handle the total interaction between the various levels has **not** yet been designed.

Use of Specific Knowledge (and not General Rules) to Guide the Generation Process

A final remark would be that the generation process does **not** operate by use of generation rules of a general nature. Of course, this claim is not that there exist no general principles of generation. For example, it is certain that one must not say something which has already been said, or which is easily inferable from the previous phrases. But first, the application of such abstract principles varies from one case to another under conditions which escape all formalization in the present state of knowledge. Secondly, the generation of connected text must rely upon discourse structures. Such structures are domain and task dependent. For instance, in order to determine whether we should indicate explicitly or implicitly the context in which the death of one or more people is situated, we should take into account at least two types of considerations: the first type is the *linguistic order* , for example, the verb "to kill" could be used for both a crime and accident context, unlike "to assassinate" that necessarily implies an attempt (crime). The second type of considerations is the *pragmatic order* , for example, the knowledge that one has on political conflicts that oppose the individuals of certain nationalities. This linguistic or pragmatic knowledge is very specific.

Thus one should not expect any general rule that would determine the inferable elements of a story . Instead, the generation process requires methods based on concrete descriptions of language facts, that is, on linguistic knowledge which is precise, detailed and elegant. The generator must be given the patterns (templates) of sentences [Danlos 1987b] where the verbs occur and the situations that correspond to the patterns. This use of templates gives far more information than those which are used only for the purpose of indicating the relation between the semantic roles (ACTOR, TARGET, etc.) and the syntactic complements (subject, direct object, etc.). An example of such a sentence template underlying the sentence Ugo shot Mary **and** she was killed is:

S1 (: COORD - S (: coord-conj *and*)
(: cl (: subject HUM1) (: verb *shoot* (:tense past) (:dir-object HUM2))
(: cl (: subject HUM2) (: verb *kill* (:tense past) (: voice passive))))

where HUM1 and HUM2 are tokens with the following definitions:

HUM1=: PERSON	HUM2=: PERSON
NAME=: Ugo	NAME=: Maria
SEX=: masc	SEX=: fem

Issues on Coordination

In general, the **coordination** operation enables joining formulations of different positive results and is usually followed by the deletion of repeated elements. For example, John has the book and John has the pen ---> John has the book and the pen.

However, this operation is appropriate only when it affects semantically comparable elements. For example, coordinating two sentences such as: * *Luc a de la haine* pour Max et *Luc a de la margarine* dans son frigidaire (Luc hates Max and Luc has sonme butter in his fridge) without any semantic relationship has a switching character.

Especially, this operation gives best results when it is applied to elements of the same form, e.g., John has some butter in his fridge and John has some cheece in his fridge ---> John has some butter and some cheece in his fridge.

Finally, when the coordination is applied to more than two elements, all occurrences of the conjuction of the coordination, except the last one, should be replaced by commas: John has butter and cheece and sugar and chocolate ---> John has butter, cheece, sugar and chocolate.

12. RELATED AREAS TO NL GENERATION

In completing this paper, we are going to present briefly some related areas of interest to NL generation such as: the area of explanation, the area of psycholinguistics as well as the translation and the database (question/answering) systems.

We first examine the **the area of explanation**. An early work of this kind was done by [Chester 1976] who developed a method to translate formal deduction proofs into English. His primary concern was to structure a linear sequence of proof statements into connected paragraph-length units (*text planning*). [Shortliffe 1984] has used the quite old method of *recursive template-instantiation* for translating MYCIN rules into English. However, this is a method that generates sentences in isolation: each rule is mapped independently to a single English sentence, without regard for how the others are translated. On the other hand, [Swartout 1983] presents work on enabling an expert system to justify its behavior rather than just describing the context for that behavior, as MYCIN does.[Miller 1984a,1984b] has developed an approach for "critiquing" both anesthesiology plans and plans for the management of hypertension. Its work involved both *content determination* and *text planning* .

An interesting work in explanation has also been done by [Weiner 1980] and [Goguen et al. 1982]. They formulated a theory of explanation structure on the basis of an analysis of naturally occurring explanations. They propose an *explanation grammar* which is similar to schemata because it considers what orderings of propositions are possible and captures the hierarchical structure of text. At the same time, they incorporate the notion of focus of attention by maintaining a pointer to the proposition in focus at each point in the explanation.[Stevens and Steinberg 1981] have looked at some of the *strategies* needed to provide explanations. Analyzing texts used by the Navy for instruction about propulsion plants, they identified nine types of explanations which included such strategies as : describing the flow of information through the process, describing the process components, etc. Finally, [Forbus 1981] has proposed a system that would use qualitative simulation of processes to provide explanations of this sort.

An area of interest to NL generation is also **the area of psycholinguistics** where approaches are developed to simulate a speaker and thus produce "psychologically plausible" generators. There has been work towards this purpose by [Kempen and Hoenkamp 1982] for restarting and lapse phenomena and by [McDonald 1984, 1985] for an account of people´s fluency and lack of grammatical errors and certain classes of speech errors (see also section 7).

A distinct work on lapse study done by [Garrett 1980] and other psychologists such as [Goldman and Eisler 1980] and [Butterworth 1980] shows that a speaker plans in advance what he/she is going to say, and what he/she is expressing, at a given moment, depends not only in the sequence of his/her preceding words but also on those that will follow. Thus the hypothesis that a discourse is based on left-to-right procedures should be refined so that it include a global appreciation and in particular the whole discourse structure.

It should be noted that **the translation area** also involves generation issues, once the word-for-word approach is abandoned. These are primarily issues of *realization* : thus while most translation systems restrict their activities to fairly local, and regular ordering (e.g., verb shifting by rule), language differences between source and target languages can nevertheless make an even adequate realization a tricky process. For instance, literal translation between the two languages may cause serious semantic problems, as it is the case where a personal pronoun is authorized in one of the languages but not in the other.

Finally, generation of natural language in **database** (or **question/answering) systems** has been used for providing *responses* to questions asked and for *paraphrasing* user´s questions as a means of verification.For the most part, this work has been concerned with issues of *realization* since the *content* of the response is usually determined by the database system to which the natural language front-end is interfaced and the *content* of the paraphrase is determined by the user´s question. However, it also bears on issues of *content determination* , where the paraphrase of multisentence text must show unequivocally how a definite pronoun or noun phrase has been resolved.

Response generation in database systems has been handled through the use of fairly simple techniques: presenting information in tabular form, or inverting the user´s question to produce a sentence which can be used to introduce the information retrieved from the database. An exception to this is [Grishman 1979] system which addressed the problem of unambiguously presenting answers containing conjuction or quantifiers.

Paraphrasing has also relied on fairly simplistic generation techniques such as "canned" text and the use of templates. Examples of such systems are the PLANES system of [Waltz 1978] and the Rendezvous system of [Codd 1978]. A exception to this type of generation can be found in the CO-OP paraphraser of [McKeown 1979] which used a transformational grammar to generate paraphrases, and a distinction between given and new information in the user´s question to generate a paraphrase that differed syntactically from the user´s. Later work on generation in database systems has also been done by [Mueckstein 1983] and [Boguraev and Sparck Jones 1984].

Furthermore, several experiments ([Malhotra 1975] and [Tennant 1979]) have shown that users need to ask questions to familiarize themselves with the database srtucture before proceeding to make requests about the database contents. Taking it into account, [McKeown 1985] developed a more sophisticated generation system, within the framework of a natural language interface to a database system, that addressed the specific problem of generating multisentence responses to questions (queries) about the database structure (see section 5).

Nevertheless, the type of the response generated by this system could be used not only for specific questions about the database structure, but also as supportive explanations for yes/no questions or as explanations for structural presumption failures [Mays 1980].

Furthermore, a system for generating textual responses to questions requiring descriptions or explanations could be useful in other application areas in addition to the database query system. **Computer assisted instruction systems** and **experts systems** are examples of areas where the provision of descriptions and explanations would be useful.

13. CONCLUSIONS

It has come the time when interest in the generation of natural language has began to grow and receive the attention it really deserves as more systems are developed which require the capability for sophisticated communication with their users. This resulted in the abandonment of the old pre-stored text and template methods to the profit of automatic generation techniques.

NL generation is the inverse of NL understanding and thus it raises a set of different problems: generation accepts a representation of some set of linguistic goals and has to produce a set of natural language sentences that realize those goals. For this purpose, a system must explicitly reason about how to organize the content of the sentences and must have much more knowledge about the effects of different ways of phrasing sentences. There are two main problem areas, *which are not wholly independent* :

1. selecting from the knowledge representation (knowledge base) the particular semantic content of the sentences (i.e., **planning the content of what to say**)
2. transforming the semantic content into actual natural language sentences (i.e., **deciding how to say it**).

The area of planning and plan-based representations of linguistic goals is relevant to the first problem. In particular, planning of an extended discourse has to be involved if the content cannot be realized in a single sentence. To build a system that can produce discourse effectively, determining content and textual shape, the development and application of *principles of discourse structure and discourse coherency, and criteria for determining information relevancy* are essential to its success. Moreover, structure derived from goal pursuit rather than from previous texts leads to larger flexibility and diversity of texts.

Finally, the generation system would ideally have a large knowledge base of basic concepts and commonsense knowledge -- possibly shared with its analyzer component -- so that it could converse about any situation that arose naturally in its domain. However, in real systems today this is not actually the case.

The mappings between the final representation, the logical form and the syntactic structure are relevant to the second problem. To build a system with a great deal of syntactic competence, it should be capable of using correctly and appropriately such syntactic devices as conjuction and ellipsis; it should be competent at fitting its utterances into a discourse, using pronominal references where appropriate, choosing syntactic structures consistent with the changing focus and lexical choices, and giving an overall feeling of coherence to the discourse.

The role of the grammar in generation is different depending on whether it is used to control the generation process or it is only one of many components that contribute to the design of the sentence. In either of these cases, the decisions about the actual structure of the sentence might be best made **last** after the more functional properties of the structure have been considered. Systemic grammars and functional unification grammars may be suitable frameworks for this type of processing.

Moreover, characterizing generation as a decision-making process, or "message" (description)-directed, where all decisions are indelible, the grammar is used to specify and represent the surface structure of the "message" as an intermediate level of representation which is then traversed and realized as a natural language text. An appropriate grammar for this purpose is the tree adjoining grammar.

Another type of grammar, called "discourse grammar", integrates conceptual., linguistic and stylistic decisions which in turn interact with syntactic and morphological operations resulting in the **non** - modularization of the linguistic component of the generation process into a strict ordering of modules that make such decisions.

Finally, the use of a phrasal lexicon makes the distinction between an ordinary lexicon and a grammar to disappear as all linguistic knowledge could be then contained in the lexicon. Thus the lexicon becomes the primary organizing tool of the generation process, where interactions among semantic and linguistic knowledge processes also appear to be responsible for fluency skills and defects in natural language texts.

To conclude, we make a final comment on one of the principal concerns of the generation research: the distinction between the *"what to say"* and *"how to say it"* questions.

We have seen that certain aspects of the traditional *"what to say"* problem cannot be separated from the *"how to say it"* component [Danlos 1984, 1989], [Appelt 1985]. In other words, **the *"how to say it"* component of generation seems to have a *"what to say"* element to it.** Furthermore, certain aspects of text planning and rhetorical structure have their influence at all levels of language production, but there seems to be a difference between the construction of text plans and the execution of the plans. Thus, the distinction between the *strategic (planning)* and *linguistic (tactical)* aspects of the generation can still hold, but strategy cannot be isolated from tactics.

14. FUTURE DIRECTIONS

A primary goal of natural language generation research is to design a system that is capable of thinking complex thoughts and communicate them to its users with the same fluency as that of a human speaker. Unfortunately, no natural-language generation system yet developed can pass the test for an extended dialog.

On the other hand, we have the problem of *"deliberate writing"* : writing that is "careful and considered" in contrast with what can happen with spontaneous writing or speech. This task may well be one of the ultimate problems of the generation research, since it requires the strongest imaginable capabilities in planning, knowledge representation, problem solving, world and person modeling, creativity, and judgement.

These are the long term goals of the generation research: *the process of creating a technology for building computer programs that can function as "authors" or "speakers", as well as the development of a computational theory of the human capacity for language (**versatility** and **creativity**) and the processes that engage it .*

Thus on the one part, in the speaking situation, generation involves **planning to communicate:** goals, intentions, actions to ask a listener´s aid, actions to indicate a reference, and so on. All this embodies an entire theory of communication whose objective is to account for how agents manage to intentionally affect the beliefs, desires and intentions of other agents. Developing such a theory requires examining utterances to determine the goals the speakers are attempting to achieve thereby, and during the process that exposes the knowledge about their environment, about their audience, and about their language that these speakers must have. For this purpose, language generation requires producing a coherent plan in the first place, and therefore requires uncovering the underlying principles that make such a plan coherent.

On the other part, in the writing situation, generation involves **text organization.** It is evident that natural text is organized and that its organization is essential to its function. Text has parts, arranged systematically. But what is the nature of text organization or structure? What are the parts, and what are the principles of arrangement? We must have answers to such questions if we are to create text generators. However, there are **no** widely accepted answers to these questions. Also, crucially for text generation, there are **no** accounts of text organization at a level of detail sufficient to support computer programming. Far more detail is needed. As a result, text generation has been inventing its own answers to these questions, having as examples [McKeown 1985] and later on

[Mann and Thompson 1987].

In **the area of planning**, physical and linguistic actions will arise from the need to perform comunicative acts. There are strong interactions between the physical and linguistic actions that comprise a coherent plan. The need to reason about and coordinate both actions argues strongly in favor of an integrated planning system. Intentions behind actions should be also recognized. A more vigorous formal description of action subsumption is required, as well as a larger repertory of linguistic actions.

Analysis of referring expressions is presently extremely limited [Appelt 1985]. A wider analysis of description identification and referring expressions is required: Identifying and characterizing the many different intentions that speakers may communicate by means of referring expressions, and exactly what it means for a speaker to intend a hearer to identify a referent, constitute important areas of research in utterance planning. Although utterance planning deals with some discourse phenomena, such as pronouns and other anaphora, it is silent on the important topic of *planning and organizing extended discourse* (i.e., to reason about discourse as a whole).

In the case of complex goals involving a large number of propositions that the hearer is supposed to believe, this process becomes inadequate, and some type of higher-level organization is needed, such as the discourse schemata of [McKeown 1985] or the rhetorical structure schemata of [Mann and Thompson 1987]. It seems intuitively plausible, although there is nothing yet to support the conjecture other than intuition, that these discourse schemata (such as, comparison and contrast, elaboration and explanation) can be treated by a planner as very high level actions.

Finally, there is a number of efficient issues concerned with how to make a planning theory useful in the design of practical computer systems: it is necessary to at least guarantee that the axioms and deduction system are as efficient as possible, e.g., avoid a large number of irrelevant inferences to be made during the course of a proof.

In **the area of text generation** [Mann 1984], [Mann and Thompson 1985, 1986, 1987] and [Matthiessen and Thompson 1986] have developed different discourse strategies in a manner more general than that of [McKeown 1985]. Because of their work, text generaration took another dimension since discourse structure can be derived directly from goal pursuit rather than from previous texts as considered by McKeown. The diversity of processes that make up the text structure offer the possibility to obtain a diversity of texts, while the system is of linguistic utility and is based on diverse and detailed appeals to the user´s model.

Other features that text generation systems can include to allow for improvement in both the quality and quantity of the text produced, are: an inferencing capability, the ability to vary the amount of detail according to the user´s beliefs and goals, and segmentation of the discourse (sentence and paragraph boundaries). Moreover, in order to incorporate information about the user´s model into the generation system, research needs to be done on exactly what information about the users can be determined from the discourse and how that information can be deduced. This effort could be based on formalisms and methods for reasoning about knowledge and belief such as in [Appelt 1985].

It is clear that these classes of methods have important places in the generation technology of the future. However, it is not clear what those places are. In these very early and formative days of generation research, as it is sorting out its problems , these can form separate but interacting themes in the development of the art.

15. BIBLIOGRAPHY

[Allen 1983] Allen, J.F., *Recognizing intentions from NL utterances* , in *Computational Models of Discourse* (M. Brady and R.C. Berwick, eds.) pp. 107-66, MIT Press, Cambridge, MA, 1983.

[Allen 1984] Allen, J.F., *Towards a general theory of action and time* , AI(23), 123-154, 1984.

[Allen and Perrault 1980] Allen, J.F. and Perrault, C.R., *Analyzing intention in utterances* , AI, 15, 143-78, 1980. Reprinted in *Readings of Natural Language Processing* (B.J. Grosz, K.S. Jones, B.L. Webber eds.), pp. 441-58, Morgan Kaufmann: Los Altos, 1986.

[Appelt 1983] Appelt, D.E., *TELEGRAM: A grammar formalism for language planning* , Proc. of the 8^{th} IJCAI, pp. 595-599, 1983.

[Appelt 1985] Appelt, D.E., *Studies in NL processing: Planning English sentences* , Cambridge, England: Cambridge, U. Press, 1985.

[Becker 1975] Becker, J., *The phrasal lexicon* , Proc. Theoretical issues in NL Processing (TINLAP-I), 60-64, 1975.

[Boguraev and Sparck Jones 1984] Boguraev, B.K. and Sparck Jones, K., *A NL front end to DBs with evaluative feedback* , in *New Applications of Data Bases* (Gandarin G. and Gelenbe E. eds.), pp. 159-182, Academic Press, London, England, 1984.

[Buchberger and Horacek 1984] Buchberger, E. and Horacek, H., *VIE-GEN: A generator for German texts* , in *Natural Language Generation Systems* (McDonald D. and Bolc L. eds.), Springer-Verlag, New York Inc, pp. 166-204, 1988.

[Busemann 1984] Busemann, S., *Surface transformations during the generation of written German sentences* , in *Natural Language Generation Systems* , (McDonald D. and Bolc L. eds.), Springer-Verlag, New York Inc, pp. 98-165, 1988.

[Butterworth 1980] Butterworth, B., *Evidence from Pauses in speech* , in *Language Production* (B. Butterworth eds.), Academic Press, London, 1980.

[Chester 1976] Chester, D.,*The translation of formal proofs into English* , AI(7), 261-75, 1976.
[Codd 1978] Codd, E.F., *Rendezvous version 1: An experimental English language query formulation system for casual users of relational databases* , IBM Research Laboratory, San Jose, Ca, Technical Report RJ2144(29407), 1978.

[Cohen 1978] Cohen, P., *On knowing what to say: Planning speech acts* , PhD thesis, University of Toronto, TR 118, 1978.

[Cohen 1981] Cohen, P., *The need for identification as a planned action* , Proc. of the 7th IJCAI, pp. 31-36, 1981.

[Cohen 1984] Cohen, P., *Refering as requesting* , COLING - 84, 207-11, 1984.

[Danlos 1984] Danlos, L., *Conceptual and linguistic decisions in generation* , Proc. of the COLING - 84, pp. 501-504, 1984.

[Danlos 1985] Danlos, L., *Génération automatique de textes en langues naturelles* , Masson, Paris, 1985.

[Danlos 1986] Danlos, L., *A robust syntactic component in generation* , in *Proccedings of the 3rd International Workshop on Text Generation* (Gerard Kempen eds.), Nijmegen, The Netherlands, August, 1986.

[Danlos 1987a] Danlos, L., *The linguistic basis of text generation* , Cambridge University Press, Cambridge, 1987.

[Danlos 1987b] Danlos, L., *A French and English syntactic component for generation* , in *Natural Language Generation: New Results in Artificial Intelligence, Psychology and Linguistics* (Kempen G. eds.). Dortrecht/Boston, Martinus Nijhoff Publishers, 1987.

[Danlos 1989] Danlos, L., *Some pronominalization issues in generation of texts in Romance languages* , in *Procesamiento del Lenguaje Natural* , Actas de la IV Reunión Annual de la SEPLN, Boletín nº 7, Enero 1989.

[Danlos and Namer 1988] Danlos, L., Namer, F., *Morphology and cross dependencies in the synthesis of personal pronouns in Romance languages* , Proc. of the COLING - 88, Budapest, 1988.
[Forbus 1981] Forbus, K., *Using qualitative simulation to generate explanations* , Cognitive Science nº 3, 1981.

[Garrett 1980] Garrett, M., *Levels of processing in sentence production* , in *Language Production* (B. Butterworth eds.), Academic Press, London, 1980.

[Goguen et al. 1982] Gogeun J., Linde C., and Weiner J., *Reasoning and natural explanation,* unpublished proceedings of ISI Workshop on Explanation, 1982.

[Goldman 1975] Goldman, N.M., *Conceptual generation* , in *Conceptual Information Processing* (R.C. Schank eds.), Amsterdam: North-Holland, 1975.

[Goldman and Eisler 1980] Goldman - Eisler F., *Psychological mechanisms of speech production as studied through the analysis of simultaneous production* , in *Language Production* (B. Butterworth eds.), Academic Press, London, 1980.

[Grishman 1979] Grishman, R., *Response generation in question-answering systems* , Proc. of the 17th Annual Meeting of the ACL, pp. 99-102, La Jolla, CA, August, 1979.

[Gross 1986] Gross, M., *Lexicon-Grammar: The representation of compound words* , in Proc. of the COLING - 86, 11th International Conference on Computational Linguistics, Bonn, 1986.

[Grosz et al. 1983] Grosz B., Joshi A., and Weinstein S., *Providing a unified account of definite noun phrases in discourse* , Proc. of the Association for Computational Linguistics, pp. 44-50, June, 1983.

[Halliday 1976] Halliday, M.A.K., *System and function in language* , Oxford University Press, 1976.

[Halliday 1985] Halliday, M.A.K., *A short introduction to functional grammar* , London: Arnold, 1985.

[Houghton 1989] Houghton, G., *A computationl model of discourse production* , Ablex, Norwood, N.J., 1989.

[Hovy 1985] Hovy, E., *Integrating text planning and production in generation* , Proc. of the 9th IJCAI, L.A., 848-851, 1985.

[Hovy 1986] Hovy, E., *Putting affect into text* , Proc. of the Cognitive Science Society Conference, 1986.

[Hovy 1987a] Hovy, E., *Interpretation in generation* , Proc. of the AAAI, 1987.

[Hovy 1987b] Hovy, E., *Generating natural language under pragmatic constraints* , in *Journal of Pragmatics* , vol XI, nº 6, 1987.

[Hovy 1988] Hovy, E., *Generating language with a phrasal lexicon* , in *Natural Language Generation Systems* , (McDonald D. and Bolc L. eds.), Springer-Verlag, New York Inc, pp. 353-384, 1988.

[Jacobs 1985a] Jacobs, P., *PHRED: A generator for N.L. interfaces* , Computational Linguistics, 11, 4, 219-242, 1985.

[Jacobs 1985b] Jacobs, P., *A knoweledge-based approach to language production* , PhD thesis, Berkeley Computer Science Department, TR 86/254, 1985.

[Jacobs 1986] Jacobs, P., *Knowledge structures for natural language generation* , Proc. of the COLING - 86, 1986.

[Jacobs 1987] Jacobs, P., *Knowledge-Intensive natural language generation* , AI, vol 33, nº 3, pp. 325-378, November 1987.

[Joshi 1985] Joshi, A., *TAG grammar* , in *NL Processing - Theoretical, Computational and Psychological Perspectives* (Dowty, Karttunen, Zwicky eds.), 1985.

[Kay 1979] Kay, M., *Functional grammar* , Proc. of the Berkeley Linguistic Society, 1979.

[Kay 1984] Kay, M., *FUG: A formalism for machine translation* , Proc. of the COLING - 84, Stanford, CA, pp. 75-78, July, 1984. Also in Computational Linguistics, 1984.

[Kay 1985] Kay, M., *Parsing in FUG* in *Natural Language Parsing* (Dowty D.R., Karttunen L., and Zwicky A., eds.), 251-278, NY: Cambridge, U. Press, 1985, (reprinted in RLNP).
[Kempen and Hoenkamp 1982] Kempen, G. and Hoenkamp, E., *Incremental sentence generation: Implications for the structure of a syntactic processor* , Proc. of the COLING - 82, Prague, August, 1982.

[Kroch and Joshi 1985] Kroch, T. and Joshi, A., *The linguistic relevance of Tree Adjoinjng Grammar* , Univ. of Pennsylvania, Dept. of Computer and Information Science, 1985.

[Kukich 1984] Kukich, K., *Fluency in natural language reports* , in *Natural Language Generation Systems* (McDonald D. and Bolc L. eds.), Springer-Verlag, New York Inc, pp. 280-311, 1988.

[Lancel et al. 1986] Lancel J.M., Rousselot F., Simonin N., *A grammar used for parsing and generation* , Proc. of the COLING - 86, Bonn, FRG, August, pp. 536-539, 1986.

[Lancel et al. 1988] Lancel J.M., Otani M., Simonin N., Danlos L., *Sentence Parsing and generation with a semantic dictionary and a lexicon-grammar* , in Proc. of the COLING - 88, Budapest, 1988.

[Malhotra 1975] Malhotra, A., *Design criteria for a knowledge-based English language system for management: an experimental analysis* , MAC TR-146, MIT, Cambridge, Mass., 1975.

[Mann 1982] Mann, W., *Text generation* , JACL, 8(2), 1982.

[Mann 1983a] Mann, W., *An overview of the Penman text generation system* , Proc. of the AAAI, 261-265, 1983. Also in the 21st meeting of the ACL, 74-78, 1983.

[Mann 1983b] Mann, W., *An overview of the Nigel text generation grammar* , in the 21st meeting of the ACL, 79-84, 1983.

[Mann 1984] Mann, W., *Discourse structures for text generation* , in Proc. of the COLING - 84, Stanford University, CA , 1984.

[Mann 1988] Mann, W., *Text generation: The problem of text structure* , in *Natural Language Generation Systems* (McDonald D. and Bolc L. eds.), Springer-Verlag, New York Inc, pp. 47-68, 1988.

[Mann et al. 1982] Mann W., Bates M., Grosz B., McDonald D., McKeown K., and Swartout W., *Text generation: The state of the art and literature* , JACL, 8(2), 1982.

[Mann and Mathiessen 1985] Mann W., and Mathiessen C., *Nigel: A systemic grammar for text generation* , in *Systemic Perspectives on Discourse* (Freedle eds.), Ablex, Norwood, NJ, 1985.

[Mann and Moore 1981] Mann W., and Moore J., *Computer generation of multi-paragraph English text* , JACL, 7(1), 1981.

[Mann and Thompson 1985] Mann W., and Thompson S., *Assertions from discourse structure* , in Proc. of the 11th Annual Meeting of the Berkeley Linguistic Society, Berkeley, 1985.

[Mann and Thompson 1986] Mann W., and Thompson S., *Relational propositions in discourse* , in *Discourse Processes* , 9(1), 57-90, Jan - March, 1986.

[Mann and Thompson 1987] Mann W., and Thompson S., *Rhetorical structure theory: A theory for text organization* , in *Discourse Structure* (Livia Polanyi eds.), Ablex, Norwood, NJ, 1987.

[Matthiessen and Thompson 1986] Matthiessen C., Thompson S., *The structure of discourse and subordination* , in *Clause Combining in Grammar and Discourse* (Haiman and Thompson eds.), Benjamins, Amsterdam, 1986.

[Mays 1980] Mays E., *Correcting misconceptions about database structure* , Proc. of the 3rd CSCSI Biennial Meeting, Victoria, B.C., May 1980.

[McDonald 1983] McDonald, D., *NLG as a computational problem: An introduction* , in *Computational Models of Discourse* (M. Brady and R. Berwick eds.), MIT Press, Cambridge, MA, pp. 209-266, 1983.

[McDonald 1984] McDonald, D., *Description directed control: Its implications for NLG* , in

Computational Linguistics (Cercone eds.), Plenum, NY, pp. 403-424, 1984.

[McDonald 1985] McDonald, D., *Description directed NLG* , Proc. of the 9th IJCAI, Los Angeles, CA, pp. 799-805, 1985.
[McDonald and Conklin 1982] McDonald D., and Conklin E.J., *Salience as a simplifying metaphor for NLG* , Proc. of the AAAI, 75-78, 1982.

[McDonald and Pustejovsky 1985] McDonald D., and Pustejovsky J., *TAGs as a grammatical formalism for generation* , Proc. of the ACL, Chicago, pp. 94-103, July 1985.

[McKeown 1979] McKeown K., *Paraphrasing using given and new information in a question-answering system* , in Proc. of the 17th Annual Meeting of the ACL, pp. 67-72, 1979. Also in JACL, 9(1), pp. 1-11, Jan - March 1983.

[McKeown 1985] McKeown K.,*Text generation* , Cambridge U. Press, Cambridge, UK, 1985.

[Mellish and Evans 1989] Mellish C., and Evans R., *Natural language generation from plans,* Computational Linguistics, Vol. 15, nº 4, December 1989.

[Miller 1984a] Miller, P., *A critiquing approach to expert computer advice: ATTENDING* , Pitman Publishing Co., London, England, 1984.

[Miller 1984b] Miller, P., *Medical plan-analysis by computer: Critiquing the pharmacologic management of essential hypertension* , Computers and Biomedical Research (17), 38-54, 1984.

[Mueckstein 1983] Mueckstein, E.M., *Q-Trans: Query translation into English* , in Proc. of the 8th IJCAI, pp. 660-662, Karlsruhe, W. Germany, August 1983.

[Paris and McKeown 1986] Paris C., and McKeown K., *Discourse strategies for descriptions of complex physical objects* , in *Proccedings of the 3rd International Workshop on Text Generation* (Gerard Kempen eds.), Nijmegen, The Netherlands, August, 1986.

[Patten and Ritchie 1986] Patten T., and Ritchie G., *A formal model of systemic grammar* , in *Proccedings of the 3rd International Workshop on Text Generation* (Gerard Kempen eds.), Nijmegen, The Netherlands, August, 1986.

[Perrault and Allen 1980] Perrault C.R., and Allen J.F., *A plan-based analysis of indirect speech act* , JACL, 6(3), 167-182, 1980.

[Ritchie 1986] Ritchie, G., *The computational complexity of sentence generation using FUG* , Proc. of the COLING - 86, Bonn, FRG, 25-29, August 1986.

[Schank 1975] Schank, R.C., *Conceptual information processing* , North-Holland Publishing Company, 1975.

[Shapiro 1982] Shapiro, S.C., *Generalized augmented transition network grammars for generation from semantic networks* , JACL, 8(1), 12-25, 1982.

[Shapiro 1987] Shapiro, S.C. (ed), *Encyclopedia of A.I.* , New York, Wiley, 1987.

[Shortliffe 1984] Shortliffe, E., *Details of the consultation system* , in *Rule-Based Expert Systems* (Buchanan B., and Shortliffe E., eds.), Addison-Wesley, 1984.

[Sidner 1982] Sidner, C., *Focusing in the comprehension of definite anaphora* , in *Reading in natural language processing* (Grosz B.J., Jones K.S., Webber B.L., eds.), Morgan Kaufmann Publishers Inc., Los Altos, CA, pp. 363-394, 1986.

[Simmons and Slocum 1972] Simmons, R. and Slocum, J., *Generating English discourse from semantic networks* , Commun. of the ACM, 15, 10, 891-905, 1972.

[Stevens and Steinberg 1981] Stevens A. and Steinberg C., *A typology of explanations and its application to intelligent computer aided instruction* , Bolt Beranek and Newman Inc., TR 4626, March 1981.

[Swartout 1983] Swartout, W., *XPLAIN: A system for creating and explaining expert consulting programs* , AI (21), 285-325, 1983.

[Tennant 1979] Tennant, H., *Experrience with the evaluation of NL question answerers* , Working paper 18, Univ. of Illinois, Urbana-Champain, ILL, 1979.

[Vijay-Shankar and Joshi 1985] Vijay-Shankar, K., and Joshi, A., *Some computational* *properties of Tree Adjoining Grammars*, 23rd Annual Meeting of the ACL, pp. 82-93, 1985.

[Wahlster et al. 1983] Wahlster W., Marburger H., Jameson A., Busemann S., *Over-Answering yes-no-questions: Extended responses in a NL interface to a vision system*, Proc. of the 8th IJCAI, Karlsruhe, pp. 643-646, 1983.

[Waltz 1978] Waltz D.L., *An English language question-answering system for a large relational database*, Communications of the ACM, Vol. 21, nº 7, July 1978.

[Weiner 1980] Weiner, J.L., *BLAH, a system which explains its reasoning*, AI (15), pp. 19-48, 1980.

[Wilensky et al. 1984] Wilensky R., Arens Y., Chin D., *Talking to UNIX in English: An overview of UC*, Communications of the ACM, 27 (6), 1984.