Discourse Strategies for Generating Natural-Language Text*

Kathleen R. McKeown
Department of Computer Science, Columbia University, New York, NY 10027, U.S.A.

ABSTRACT

If a generation system is to produce text in response to a given communicative goal, it must be able to determine what to include in its text and how to organize this information so that it can be easily understood. In this paper, a computational model of discourse strategies *is presented that can be used to guide the generation process in its decisions about what to say next. The model is based on an analysis of naturally occurring texts and represents strategies that can be used for three communicative goals*: define, compare, *and* describe. *We show how this model has been implemented in* TEXT, *a system which generates paragraph-length responses to questions about database structure.*

1. Introduction

In order to appropriately generate natural-language text, a system must be able to determine what information to include and how to organize this information to achieve its communicative goal most effectively. While researchers in natural-language processing have investigated issues involved in determining the surface structure of a pre-determined message in natural language, problems involving the content and textual shape of the message have gone largely unanswered. In this paper, a generation theory is presented that identifies how the content and organization of a text can be determined given a communicative goal. The theory has been implemented in the TEXT system, which generates paragraph-length responses to questions about database structure.

Our approach is based on the fundamental hypothesis that people have preconceived ideas about the means with which particular communicative goals

* This work was done while the author was a graduate student at University of Pennsylvania and was partially supported by NSF grant #MCS81-07290. Research in artificial intelligence at Columbia University is supported by ARPA grant N00039-84-C-0165 and ONR grant N00014-82-K-0256.

0004-3702/85/$3.30 © 1985, Elsevier Science Publishers B.V. (North-Holland)

can be achieved as well as about the ways in which these means can be integrated to form a text. That is, texts reflect one or more principles of organization. The structure of a narrative, for example, follows certain standard patterns, one of which dictates that it begin with a description of setting (scene, characters, or time-frame). A computational model of *discourse strategies* encoding text organization has been developed that is used to guide the generation process in deciding what to say next. The model was developed for three communicative goals: *define*, *describe*, and *compare*. It is based on an analysis of discourse strategies that are commonly used in naturally occurring texts for these purposes.

Discourse strategies are only part of the generation method developed. Interaction with semantic knowledge about information relevant to the communicative goal and how it relates to what has already been said is used to determine the final content and structure of the text. These constraints are captured in a representation of *focus of attention*. Focus of attention constrains the information that needs to be considered when deciding what to say next. It also provides constraints when the discourse strategy allows for several possible choices for what to say next by indicating which information ties in best with the preceding discourse. In this paper, focus of attention will be discussed only as it relates to the discourse strategies (for further details see [23, 24]).

The use of a formal model of discourse strategies which interacts with focus of attention constitutes a departure from earlier language-generation systems. First, it specifies a mechanism for generating coherent *text*. This is in contrast to the majority of earlier systems which focused on the generation of single sentences. Of those systems that could generate connected text (see, e.g. [17, 25, 35, 39]), few have used a formal representation of strategy to determine the content and organization of the text.[1] Furthermore, the use of interacting influences on the content and structure is another feature of our approach that is lacking in other systems.

2. Problems in Generating Text

What must a generation system take into account to generate a text, given a specific communicative goal? The following questions at least must be considered:

- How do problems in language generation differ from those of language interpretation?
- What is the range of choices a generation system must consider?
- How does generation of text differ from generation of single sentences?

Since less research has been done in language generation than interpretation, people are less familiar with its problems. Although there is research that

[1] These systems and their differences from TEXT will be discussed in detail in Section 8.

suggests that the same information can be used both for interpretation and generation (e.g. [15, 40, 42]), there are some important distinctions that can be made about the processes required for each task.

Interpretation of natural language requires examination of a text in order to determine its meaning and the intentions of the writer who produced it. It necessitates using the evidence available and examining the limited set of options the system knows to be available to the writer to determine the option actually taken. For example, in interpreting Example 0 below, a system would use the evidence that 'give' occurs in the passive form to determine that 'book' is the object being given and 'Mary' the agent that does the giving.

Example 0. Mary was given a book.

While developing an interpretation system involves specification of how a speaker's options are limited at any given point (for example, by writing grammars), it does not require a formulation of reasons for selecting between those options.[2] Thus, in interpreting Example 0, a system does not consider *why* the writer used the passive form as opposed to any of the other options available at that point.

In generation of natural language, however, this is exactly what is required. To produce Example 0 in an appropriate discourse sequence, a generator must decide that although both the active and passive forms are possible, the passive is better than the active. Furthermore, the generator must have a principled reason for making that decision, which it can use in all similar cases. Where research on interpretation may describe limitations on options in order to more efficiently determine the option taken, research in generation must specify why one option is better than others in various situations.[3]

The options which a generation system must address range across a variety of knowledge sources. 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 intent. In the last of these stages, local decisions such as the syntactic choice shown for Example 0 are made, often using a grammar and dictionary to do so. Until recently, this has been the focus of language-generation research. But determining what to say and how to structure text above the sentence level also introduce language issues that must be addressed by any speaker or writer of extended discourse. These three classes of decisions are all part of the language-generation problem.

[2] Note that as interpretation systems become more sophisticated, the analysis of reasoning behind the selection of a choice may be helpful in determining the goals and intentions of the speaker.

[3] Of course, a robust understanding system must be prepared to handle any input, while a generation system may be limited in the type of language it can produce without causing the system to actually fail (although it may produce inappropriate output).

If connected text is to be generated, issues of discourse structure and discourse coherency and their influence on content are particularly important. For some tasks, deciding what to say may be fairly straightforward (e.g., a search of a database), while for others it may require more complicated reasoning processes (e.g., selecting information appropriate to the level of the learner for computer aided instruction systems). At any rate, it is clear that one of the first steps in speaking or writing is the narrowing of attention to knowledge relevant to the purpose at hand. For example, when asked for my opinion on punk rock, it would be inappropriate for me to start telling you about my favorite Greek classic, even if I knew much more about ancient Greek literature than about punk rock. Unless I wanted to compare my knowledge about Greek classics to some aspect of punk rock, I would be unlikely to even consider it in formulating my answer.

After determining what information is likely to be relevant to its current discourse goal, a generation system must be able to decide what to say first, what next, and how to close a discourse. Order of information in a text can be crucial to a reader's understanding of it. For example, the sequence of sentences shown in Example 1 is easily understood, but if examples of a concept are presented before the concept is introduced as in Example 2, the meaning is unclear.

Example 1.
(A) Many sports are just a rich man's domain.
(B) Skiing, golf, and tennis are cases in point.

Example 2.
(A) Skiing, golf, and tennis are cases in point.
(B) Many sports are just a rich man's domain.

Given that the generator is producing text and not simply single sentences, certain choices at the surface level are critical in order to produce a coherent text. The generator must be able to make reasoned decisions about when to use pronominal reference and about the syntactic construction that should be used. Examples illustrating these choices are shown in Examples 3–5 below. While a generator could arbitrarily decide which of these choices to select in any given situation, an inappropriate decision could easily be made without additional guidance. If the three propositions shown in Examples 3–5 are to be expressed as part of a textual sequence (Examples 6–8 below), then one choice in each pair is clearly inappropriate.

Example 3. Lexical choice: *bought* vs. *sold*
(A) Jane bought $3.00 worth of bobby socks from Michael.
(B) Michael sold $3.00 worth of bobby socks to Jane.

Example 4. Pronominal choice: *Linda* vs. *she*
(A) Linda flew to Washington.
(B) She flew to Washington.

Example 5. Syntactic choice: *passive* vs. *active*
(A) John gave the book to Mary.
(B) Mary was given the book by John.

Example 6.
Jane was in a hurry to finish her shopping.
It was a chore she particularly despised.
First, Jane bought $3.00 worth of bobby socks from Michael.
*Michael sold $3.00 worth of bobby socks to Jane.[4]

Example 7.
We knew that Mary took the train to New York with Linda, but didn't realize that
Linda flew to Washington from there.
*she flew to Washington from there.

Example 8.
John bought that great new book on data structures.
He read the first three chapters and then
John gave the book to Mary.
*Mary was given the book by John.

3. The TEXT Generation Model

Our approach relies on a model of language production which divides processing into two stages. The first stage determines the content and structure of the discourse and is termed the 'strategic' component, following Thompson [38]. The second stage, the 'tactical' component, uses a grammar and dictionary to realize in English the message produced by the strategic component. This division allows for focus on the problems of determining content and structure as part of the strategic component.[5] The TEXT implementation includes both

[4] Note that even if we bring Jane into subject position still using the verb sell, the sentence is inappropriate in this sequence: *Jane was sold $3.00 worth of paper goods by Michael.

[5] A control structure which allows for backtracking between the tactical and strategic components (e.g. Appelt [2]) would also be possible. The approach we have taken clearly specifies how the planning of the text influences the realization of a message in natural language. Backtracking would allow for processes that produce the surface expression to influence the planning of the discourse. Our division of the processes, however, does allow us to focus on textual organization, an issue which Appelt has not addressed.

tactical and strategic components as a second emphasis of the work is on the kind of information the strategic component must produce to allow surface choices to be made appropriately.

The main features of the generation method developed for the TEXT strategic component are (1) the selection of relevant information for the answer; (2) the pairing of rhetorical techniques for communication (such as analogy) with discourse purposes (for example, providing definitions); and (3) a focusing mechanism. Questions are answered by first partitioning off a subset of the knowledge base determined to be relevant to the given question (Determine Relevancy, Fig. 1). This partition is termed the *relevant knowledge pool.* Then, based on the discourse purpose and a characterization of information in the relevant knowledge pool, a discourse strategy encoding partially ordered *rhetorical techniques* is chosen (Select Strategy, Fig. 1; Section 5 describes this process). These techniques guide the selection of propositions from the relevant knowledge pool. A focusing mechanism representing *immediate focus* (Section 6) interacts with the use of rhetorical strategies to fully determine the content and order of the answer. It helps maintain discourse coherency by filtering the next possible propositions indicated by the discourse strategy to that proposition which ties in most appropriately with the previous discourse. The message thus determined is passed to the tactical component which uses a functional grammar [15] to transform the message into English (for more

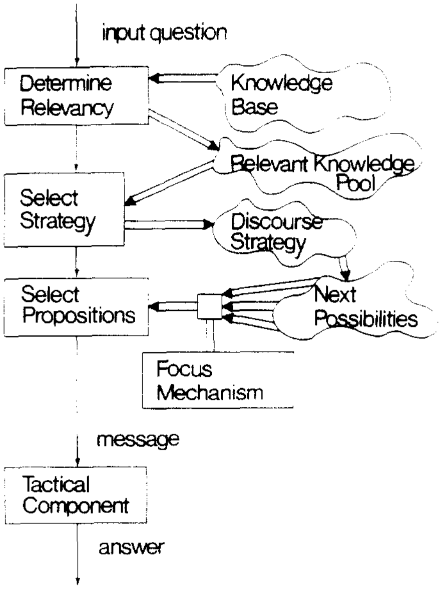


FIG. 1. System overview.

details on the tactical component see [3, 23]. This flow of control is illustrated in Fig. 1.

This method provides the means for the system to effectively (and efficiently, since it first narrows the information to be considered to a small subset of the knowledge base) determine what to include in a text and how to order it. Moreover, in the process of determining what to say next, the strategic component produces information that can be used by the tactical component to select between the surface-level choices outlined above. That is, the tactical component can use the tracking of focus of attention to select, for example, the passive construction over the active.

4. The Application Domain

Most natural-language database systems have concentrated on answering factual questions, providing answers in the form of lists or tables of objects in the database.[6] To ask such questions, a user must already be familiar with the database. Several experiments [16, 36] have shown that users often need to ask questions about database structure to familiarize themselves with it before making requests about its contents.

The task of the TEXT system is to generate responses to such meta-level questions. Three classes of questions corresponding to the three communicative goals have been considered: questions about information available in the database, requests for definitions, and questions about the differences between database entities. In this context, input questions provide the initial motivation for generating text.

Implementation of TEXT used a portion of an Office of Naval Research (ONR) database containing information about vehicles and destructive devices. Some examples of questions that can be asked of TEXT include[7]:

- What is a frigate?
- What do you know about submarines?
- What is the difference between a whisky and a kitty hawk?

The kind of generation of which the system is capable is illustrated by the response it generates to the question in Example 9 below.

[6] Note that in some systems, the list (especially in cases where it consists of only one object) may be embedded in a sentence, or a table may be introduced by a sentence which has been generated by the system [12]. In a few systems (e.g. [6, 16]), a one- or two-sentence reply about the information in the database may be generated, but this reply is usually stored as a whole in the knowledge structure.

[7] Note that the system is not able to parse input natural-language questions. Instead questions must be asked via simple function notation corresponding to the three classes of questions: (1) (definition ⟨*e*⟩), (2) (information ⟨*e*⟩), (3) (difference ⟨*e*1⟩ ⟨*e*2⟩), where ⟨*e*⟩, ⟨*e*1⟩, ⟨*e*2⟩ are entity classes in the database.

Example 9. What kind of data do you have?

All entities in the ONR database have DB attributes REMARKS. There are 2 types of entities in the ONR database: destructive devices and vehicles. The vehicle has DB attributes that provide information on SPEED INDICES and TRAVEL MEANS. The destructive device has DB attributes that provide information on LETHAL INDICES.

The knowledge base developed for TEXT is an enhanced database schema based on the Chen entity-relationship model [5]. It includes a hierarchy on both entities and attributes (part of this hierarchy was generated automatically [20]). TEXT uses this meta-level representation to answer questions, although in some cases the ONR database itself is accessed.

5. Discourse Structure

The use of discourse strategies implies that text is generated by selecting information out of the underlying knowledge base and ordering it. Textural order is not pre-determined by the underlying knowledge base, but is determined at the time of generation by the discourse strategy used. This is based on an assumption that the structure of information in memory and the structure of a description of that information need not be the same. Note that although a person may describe the same event on several different occasions, exact repetition is unlikely. If the event is related for a different purpose moreover, it is even more likely that different information will be included. Experiments done by Chafe [4] support the assumption that a speaker decides as he is talking what material should go into a sentence. He showed that the distribution of semantic constituents among sentences often varies significantly from one version of a narrative to another.

The TEXT approach is also based on the observation that people follow certain standard patterns of discourse organization for different discourse goals. The production of narratives is an obvious example. A second example is the writing of short technical papers where people use knowledge about what normally goes into an introduction and which points a conclusion should emphasize. The discourse strategies used for TEXT were formalized on the basis of an analysis of naturally occurring texts which revealed a set of standard patterns.

Earlier we showed that the order of a text can influence both its meaning and clarity: if two sentences in a text are interchanged, then its meaning may be obscured. The discourse strategies we identify in the analysis specify order and content so that a writer's purpose is clearly communicated. That is, a writer uses such strategies because it makes it easier for a reader to understand a text. For a generation system, therefore, the strategies serve two purposes: they constitute a tractable mechanism which a system can use to generate text and they specify an appropriate ordering that ensures effective communication of specific discourse goals.

In addition to ensuring understandable text, the use of discourse strategies also implies that different descriptions can be generated from the same knowledge representation. Since the discourse strategies control what is said and how that is structured, different strategies can be mapped onto the same piece of knowledge base to produce different texts. This means that the knowledge representation does not have to be appropriately structured for the generation task in addition to meeting all the other demands which are placed on it. This is not to say that representation isn't important for generation. The organization of the text, however, will not be dependent upon a particular organization of the knowledge base.

5.1. Rhetorical predicates

The basic units of discourse strategies are *rhetorical predicates*. They characterize the predicating acts a speaker may use and delineate the structural relation between propositions in a text. Some examples are 'analogy', 'constituency' (description of sub-parts or sub-types), and 'attributive' (providing detail about an entity or event). Linguistic discussion of such predicates indicates that some combinations of rhetorical predicates are preferable to others.

The notion of predicates goes back to Aristotle [22], who describes *enthymemes* and *examples*, predicates which a speaker can use for persuasive argument. Both Williams [41] and Shipherd [31], old-style grammarians, categorize sentences by their function in order to illustrate to the beginning writer how to construct paragraphs although neither says anything about combining sentence functions to form paragraphs. More recently, Grimes [11] has described rhetorical predicates as explicit organizing relations used in discourse. Grimes claims that the predicates are recursive and can be used to identify the organization of text at any level (i.e., proposition, sentence, paragraph, or longer sequence of text), but does not show how.

Our examination of texts and transcripts[8] has shown that not only are certain combinations of rhetorical techniques more likely than others, some are more appropriate for one communicative goal than another. For example, *definitions* of objects were frequently provided by a principled combination of certain techniques, while a *comparison* of two objects used others. For the analysis, a variety of texts was examined – ten different authors, in varying styles, from very literate written to transcribed spoken texts form the basis of the study. Short samples of *expository* texts were used because of their relevance to the

[8] Transcripts of mother-child dialogues [32] providing definitions of unfamiliar objects were used.

system being developed. This also avoided problems involved in narrative writing (e.g., scene, temporal description, personality).

To do the analysis, each proposition in self-contained samples from the texts was classified as one of a set of predicates. The set of predicates was drawn as much as possible from previous linguistic work. Both Grimes' and Williams' predicates were used (Figs. 2 and 3), but these did not capture all the structural relations in the examined texts, so an additional three predicates were adopted (Fig. 4).

The definitions of the predicates put forth by the authors were used to determine how to classify a proposition. These definitions were usually stated in English. For example, Grimes defines explanation as a proposition which provides the reason for which an inference (which can be implicit or explicit in the text) was drawn. Evidence, on the other hand, characterizes a proposition which provides support for a stated fact. Examples illustrating the use of other predicates are given in Figs. 2–4. While these definitions are not precise (see

1. **Attributive:**
   Mary has a pink coat.
2. **Equivalent:**
   Wines described as 'great' are fine wines from an especially good village.
3. **Specification (of general fact):**
   Mary is quite heavy. *She weighs 200 pounds.*
4. **Explanation (reasoning behind an inference drawn):**
   So people form a low self-image of themselves, *because their lives can never match the way Americans live on the screen.*
5. **Evidence (for a given fact):**
   The audience recognized the difference. *They started laughing right from the very first frames of that film.*
6. **Analogy:**
   You make it in exactly the same way as red-wine sangria, except that you use any of your inexpensive white wines instead of one of your inexpensive reds.
7. **Representative (item representative of a set):**
   What does a giraffe have that's special? . . . a long neck.
8. **Constituency (presentation of sub-parts or sub-classes):**
   This is an octopus . . . *There is his eye, these are his legs, and he has these suction cups.*
9. **Covariance (antecedent, consequent statement):**
   If John went to the movies, then he can tell us what happened.
10. **Alternatives:**
    We can visit the Empire State Building or call it a day.
11. **Cause-effect:**
    The addition of spirit during the period of fermentation arrests the fermentation development . . .
12. **Adversative:**
    It was a case of sink or swim.
13. **Inference:**
    So people form a low self-image of themselves.

FIG. 2. Grimes' predicates.

Williams' predicates are illustrated by providing an example paragraph from his text in which each sentence is classified as one of his predicates. The classifying predicate follows the sentence.

Comparison
General-illustration
Amplification
Conclusion
Topic
Particular-illustration
Contrasting

"What, then, are the proper encouragements of genius? (topic). I answer, subsistence and respect, for these are rewards congenial to nature. (amplification). Every animal has an aliment suited to its constitution. (general-illustration). The heavy ox seeks nourishment from earth; the light chameleon has been supposed to exist on air. (particular-illustration). A sparer diet than even this satisfies the man of true genius, for he makes a luxurious banquet upon empty applause. (comparison). It is this alone which has inspired all that ever was truly great and noble among us. It is as Cicero finely calls it, the echo of virtue. (amplification). Avarice is the pain of inferior natures; money the pay of the common herd. (contrasting sentences). The author who draws his quill merely to take a purse no more deserves success than he who presents a pistol. (conclusion)."

FIG. 3. Williams' predicates.

[13] for more formal definitions), in general they provide characterizations of predicates which are sufficiently distinct to allow for classification.

To classify a proposition, the text was segmented into clauses that could be characterized by one of the predicates. For the most part, a proposition corresponds to a single clause in the text, but in some cases several clauses together better capture a predicate (for instance, see the text example for predicate 5, evidence in Fig. 2). In addition, the classification of a proposition was sometimes ambiguous between several predicates and for such cases, the single proposition was classified by all applicable predicates.

Our analysis has shown that, with slight variations, similar patterns of predicate usage occur across the various expository texts. Four different predicate patterns were noted and have been represented as *schemata*. They are the *identification*, *constituency*, *attributive*, and *contrastive* schemata and are shown in Figs. 5–8. A grammar notation was used to represent the schemata: '{ }' indicates optionality, '/' indicates alternatives, '+' indicates that the item may appear 1 to *n* times, and '*' indicates that the item is optional and may

1. **Identification:**
   ELTVILLE (Germany) An important wine village of the Rheingau region.
2. **Renaming:**
   Also known as the Red Baron.
3. **Positing:**
   Just think of Marcus Welby.

FIG. 4. Additional predicates needed for the analysis.

appear 0 to *n* times. Each schema is followed by a sample paragraph and a classification of the propositions contained in the paragraph. ‘;’ is used to represent classification of ambiguous propositions in the paragraph. These were translated into the schemata as alternatives.

The *identification* schema (Fig. 5) captures a strategy used for providing definitions. Its characteristic techniques include identification of an item as a member of a generic class (identification), description of an object's constituency or attributes (constituency/attributive), analogies (analogy), and examples (particular-illustration/evidence). It should be noted that the *identification* schema was only found in texts whose primary function was to provide definitions (e.g., dictionaries and encyclopedias). The other texts examined simply did not have occasion to provide definitions.

The *constituency* schema (Fig. 6) describes an entity or event in terms of its sub-parts or sub-types. After identifying the sub-types, the schema dictates either a switch to each of its sub-types in turn (following the depth-identification or depth-attributive path) or can continue focusing on the entity itself, describing either its attributes (attributive path) or its functions (cause-effect path). The schema may end by optionally returning to discussion of the original object by using the amplification, explanation, attributive or analogy predicate. In the sample paragraph, taken from the *American Encyclopedia*, two types of torpedoes are first introduced. Then, the steam-propelled model is identified by citing facts about it and the electric-powered model is compared against it, with a significant difference cited.

Identification schema

Identification (class & attribute/function)
{Analogy/Constituency/Attributive/Renaming/Amplification}*
Particular-illustration/Evidence+
{Amplification/Analogy/Attributive}
{Particular-illustration/Evidence}

Example

"**Eltville** (Germany) (1) An important wine village of the Rheingau region. (2) The vineyards make wines that are emphatically of the Rheingau style, (3) with a considerable weight for a white wine. (4) Taubenberg, Sonnenberg and Langenstuck are among vineyards of note." (Paterson [28]).

Classification of example

1. Identification (class & attribute)
2. Attributive
3. Amplification
4. Particular-illustration

FIG. 5. The identification schema.

Constituency schema

Constituency
Cause-effect*/Attributive*/
{ Depth-identification/Depth-attributive
{Particular-illustration/evidence}
{Comparison/analogy} }+
{Amplification/Explanation/Attributive/Analogy}

Example

"Steam and electric torpedoes. (1) Modern torpedoes are of 2 general types. (2) Steam-propelled models have speeds of 27 to 45 knots and ranges of 4000 to 25,000 yds. (4,367–27,350 meters). (3) The electric powered models are similar (4) but do not leave the telltale wake created by the exhaust of a steam torpedo" [9].

Classification of example

1. Constituency
2. Depth-identification; (Depth-attributive)
3. Comparison
4. Depth-identification; (Depth-attributive)

FIG. 6. The constituency schema.

The *attributive* schema (Fig. 7) can be used to illustrate a particular point about a concept or object. The sample paragraph, taken from the Introduction to *Working*, attributes the topic (working and violence) to the book, amplifies on that in the second proposition ("spiritual as well as physical"), and in the third sentence, provides a series of illustrations. The fourth proposition selects out one instance as representative of the problem and the fifth amplifies on that instance.

The *contrastive* schema (Fig. 8) is used to describe something by contrasting a major point against a negative point (something the speaker wishes to show isn't true). The negative point is introduced first. The major concept is then described in more detail using one or more of the predicates shown in the second option of the schema. The closing sequence makes a direct comparison between the two. This schema dictates the structural relation between the two concepts – the use of A and $\sim A$ (not A) in the schema represent the major and negative points – but is less restrictive about which predicates are used than the other schemata.

In the sample paragraph, the contrastive schema is used to show how people form a bad self-image by comparing themselves against those in the movies. In the first sentence, the movie standard is introduced (the negative point or $\sim A$). In the second and third sentences, real-life occupations and the feelings associated with them are described (the major point or A). This is done by first attributing the property of 'feeling degraded' to people such as waitresses and then providing evidence for that staτement. Finally, a direct comparison is made between the glamorized occupations in movies and professions that people

Attributive schema

Attributive
{Amplification; Restriction}
Particular-illustration*
{Representative}
{Question; Problem
Answer}
{Comparison; Contrast
Adversative}
Amplification/Explanation/Inference/Comparison

Example

"(1) This book, being about work, is, by its very nature, about violence – (2) to the spirit as well as to the body. (3) It is about ulcers as well as accidents, about shouting matches as well as fistfights, about nervous breakdowns as well as kicking the dog around. (4) It is, above all (or beneath all), about daily humiliations. (5) To survive the day is triumph enough for the walking wounded among the great many of us" (Terkel [38]).

Classification of example

1. Attributive
2. Amplification
3. Particular-illustration
4. Representative
5. Amplification; Explanation

FIG. 7. The attributive schema.

have in real-life[9] and an inference drawn: "people form a low self-image of themselves".

It should be noted that the patterns found are fairly unrestrictive. Each contains a number of alternatives, indicating that a speaker has a wide variety of options within each type of structure. Moreover, since it is difficult to precisely define a predicate, the interpretation of each predicate in the pattern allows for additional speaker variation. In other words, text structure is not very rigidly defined. It allows for more individual variation than does the structure of sentences, for example.

Furthermore, the patterns are descriptive and not prescriptive. They identify commonly used means for achieving discourse goals, but do not dictate that these are the *only* means for achieving those goals. For example, the *identification* schema specifies one strategy for providing a definition, but it is likely that there are others. A mechanical device, for instance, might be defined by describing its function.

[9] Note that this statement could also be interpreted as an explanation for why people feel degraded and thus an ambiguous classification was made.

Compare and contrast schema

Positing/Attributive (~*A*)
{Attributive (*A*)/
  Particular-illustration/Evidence (*A*)/
  Amplification (*A*)/
  Inference (*A*)/
  Explanation (*A*)}+
{Comparison (*A* and ~*A*)/
  Explanation (*A* and ~*A*)/
  Generalization (*A* and ~*A*)/
  Inference (*A* and ~*A*)}+

"(1) Movies set up these glamorized occupations. (2) When people find they are waitresses, they feel degraded. (3) No kid says I want to be a waiter, I want to run a cleaning establishment. (4) There is a tendency in movies to degrade people if they don't have white-collar professions. (5) So, people form a low self-image of themselves, (6) because their lives can never match the way Americans live – on the screen." (Terkel [38])

Classification of example

1. Positing (~*A*)
2. Attributive (*A*)
3. Evidence (*A*)
4. Comparison, explanation (*A* and ~*A*)
5. Inference (*A* and ~*A*)
6. Comparison, explanation (*A* and ~*A*)

FIG. 8. The compare and contrast schema.

Moreover, the schemata do not have the same binding action on the writer as does a sentence grammar. A text that breaks the rules specified by a schema is not perceived as 'illegal' or outside the English language, while a sentence that does not conform to a grammar is often recognized as ill-formed. Earlier on we noted that writers of short technical papers are aware of conventions for what constitutes a reasonable introduction. A talented writer, however, may purposely ignore such conventions to create a particularly captivating introduction. While such a text may be considered unusual, it would not be considered wrong.

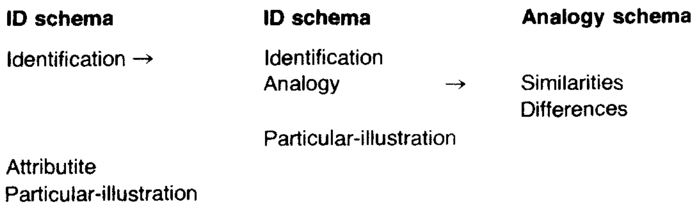
All this points to the fact that the schemata do *not* function as grammars of text. They do, however, identify common means for effectively achieving certain discourse goals. They capture patterns of textual structure that are frequently used by a variety of people (and therefore do not reflect individual variation in style) to successfully communicate information for a particular purpose. Thus, they describe the norm for achieving given discourse goals, although they do not capture all the means for achieving these goals. Since they formally capture means that are used by people to create understandable texts, they can be used by a generation system to produce effective text.[10]

5.1.1. *Schema recursion*

As Grimes suggested, the rhetorical predicates do function recursively, describing the structure of text at all levels. For example, a single sentence may be used to attribute information to an entity or a longer sequence of text may be used for the same purpose. Our analysis of texts was made in order to discover just how predicates are combined to form longer sequences of text having specific functions. Thus, the resulting schemata describe combinations of predicates which serve the function of a single predicate. For this reason, each schema is associated with the predicate whose function it serves.

Schema recursion is achieved by allowing each predicate in a schema to expand to either a single proposition (e.g., a sentence) or to a schema (e.g., a text sequence). A text generated by applying schemata recursively will be tree-structured, with a sub-tree occurring at each point where a predicate has been expanded into a schema. Propositions occur at the leaves of the tree.

Fig. 9 shows a hypothetical example of how schema recursion works. The *identification* schema is used in response to the question "What is a Hobie Cat?". The first step is to identify the Hobie Cat as a class of catamarans (1). To do so, a definition of a catamaran is also provided, assuming that the listener knows little about sailing. The identification predicate expands to the *identification* schema, where the speaker identifies the catamaran as a sailboat (2) and provides an analogy between the two, which consists of their similarities (3) and differences (4). These two steps are dictated by an *analogy* schema.



(1) A Hobbie Cat is a brand of catamaran, (2) which is a kind of sailboat. (3) Catamarans have sails and a mast like other sailboats, (4) but they have two hulls instead of one. (5) That thing over there is a catamaran. (6) Hobie Cats have a canvas cockpit connecting the two pontoons and one or two sails. (7) The 16-ft. Hobie Cat has a main and a jib and the 14-ft. Hobie Cat has only a main.

FIG. 9. Schema recursion.

[10] Note that while schemata are not grammars of text in general, they do serve as a text grammar for the system since they describe all possible text structure that TEXT can generate.

After pointing out a catamaran to the listener (5), the text returns to the original *identification* schema to provide additional information about the Hobie Cat (6) and finally, cites two types of Hobie Cats, the 16-ft. and the 14-ft. (7).

Although TEXT is capable of performing recursion in some instances, full recursion, such as is illustrated in the above example, is not currently implemented. In order for the system to be fully recursive, a schema must be written for each rhetorical predicate. Right now, schemata for only four of the predicates (out of a total of ten predicates) are written. (In the above example, the *analogy* schema shown is assumed to correspond to the compare and contrast schema, but this would require more analysis to verify).

Another issue in the recursive use of schemata is the question of when recursion is necessary. Clearly there are situations where a simple sentence is sufficient for fulfilling a communicative goal, while in other cases, it may be necessary to provide a more detailed explanation. One test for recursion hinges on an assessment of the user's knowledge. In the above example, a speaker might provide a detailed identification of the Hobie Cat if the listener knew very little about sailing. Tests on when detail is necessary are currently being explored by Paris [27]. In order to develop a comprehensive theory on recursion a full user-model [1, 26, 29] must be developed.

5.2. Using schemata in the TEXT system

In TEXT, schemata are used to guide the generation process in its decisions about what to say next at each step in constructing the text. They serve as a text plan. The four schemata which were developed from our analysis of texts (Figs. 5–8) are used as the basis for TEXT schemata.

The schemata were implemented using a formalism based on an augmented transition network (ATN) [43]. An ATN is a graph representation of a grammar and allows for actions on its arcs which may set or test various registers. The ATN formalism was originally developed to parse sentences. When parsing a sentence, taking an arc involves consuming a word from the input string and augmenting a syntactic parse tree to include the new word and its category.

For TEXT, an ATN is used to build discourse instead of a parse tree. Taking an arc corresponds to the selection of a proposition for the answer and the states correspond to filled stages of the schema. No input string is consumed; instead the relevant knowledge pool is consumed, although it is not consumed in any order and it need not necessarily be completely exhausted when the graph is exited.

One major difference between the TEXT ATN implementation and a usual ATN, however, is in the control of alternatives. In the TEXT system, at each state all possible next states are computed and a function that performs the focus constraints is used to select one arc from the set of possibilities. Thus, although all possible next states are explored, only one is actually taken. This differs from the normal ATN where unrestricted backtracking can occur.

Once a schema has been selected for a given response, the answer is constructed by traversing the schema, beginning at the start state. An arc's type determines how the system decides whether or not it can be traversed. There are five types of arcs in the TEXT ATN graphs: *fill, jump, push, subr*, and *pop* arcs. *Fill* arcs are used to represent the predicates of the schema. Each predicate has a function associated with it which 'matches' the predicate against the relevant knowledge pool and returns all propositions in the pool which are classified by the predicate. A *fill* arc is traversed if its predicate matches at least one proposition in the pool. On traversal, the matched proposition is consumed.

Jump arcs function as they do in the original ATNs and are used to capture optional predicates. *Subr* arcs are used to allow for simplicity in the graph. They name a sub-graph and can be traversed if the sub-graph named can be traversed. *Pop* arcs indicate where a graph is exited and *push* arcs are used for recursion.

Figs. 10–14 show the graphs that implement the schemata used in TEXT. TEXT schemata do not contain predicates of the original schemata for which no information exists in the database domain and are thus each subsets of the corresponding schemata which emerged from an analysis of the texts (see Figs. 5–8).

The TEXT *identification* graph (Figs. 10 and 11) has as its first arc a *fill* arc, emanating from the start state ID/[11]. It represents the first predicate of the schema, the identification predicate. Following the first arc is an optional arc, (*subr* Description/), which can be skipped by taking the *jump* arc also emanating from state ID/ID. The sub-graph labeled by Description/ (Fig. 11) represents the second line of the original schema, capturing three predicates that were present in the original. Following this comes at least one predicate from the Example/ sub-graph (that is, either particular-illustration or evidence). The Example/ arc leads to state ID/EX, from which the schema can be exited via the *pop* arc. Alternatively, additional Example/arcs can be taken by cycling back through state ID/EX or optional predicates from the End-seq/ sub-graph followed by another optional Example/. These last two arcs correspond to the last two lines of the original schema.

Both the TEXT *constituency* (Fig. 12) and *attributive* (Fig. 13) schema are modified versions of the schemata resulting from the text analysis. The *con-*

[11] The states in the schema are named following the normal ATN convention. The name of the schema (here ID abbreviates *identification*) precedes the '/' and the predicate arc most recently traversed appears after the slash (thus, ID/ names the initial state when no predicate arcs have been traversed and ID/ID names the second state when the identification arc of the *identification* schema has been traversed).

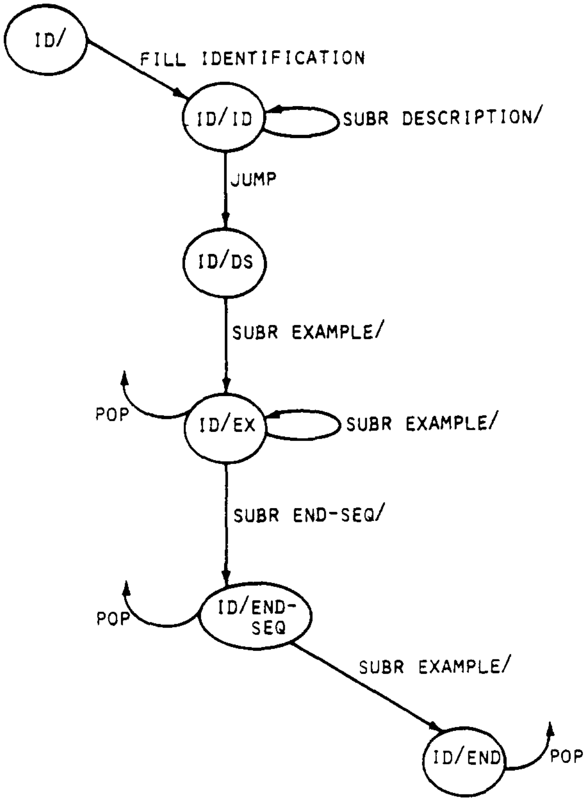


FIG. 10. The identification-schema graph.

stituency graph is optionally headed by an attributive or identification predicate so that it can be used for providing answers to requests about available information or definitions. Following state CONST/INTRO, the graph mirrors the schema (Fig. 6). Note that two of the alternatives in the schema (cause-effect* and attributive*) were eliminated from the TEXT version and only the alternative beginning with depth-identification/depth-attributive remains. Two of the predicates from the last line of the schema (attributive and analogy) were included in the graph.

The TEXT *attributive* graph does not include the restriction, question; problem, answer, or adversative predicates from the original schema as there is no translation for these in the database domain. In addition, the predicate representative in the original schema was translated as classification in the TEXT version and comparison; contrast as analogy in order to make use of predicates already used. Only the explanation predicate from the last line of the original schema was included in the TEXT graph.

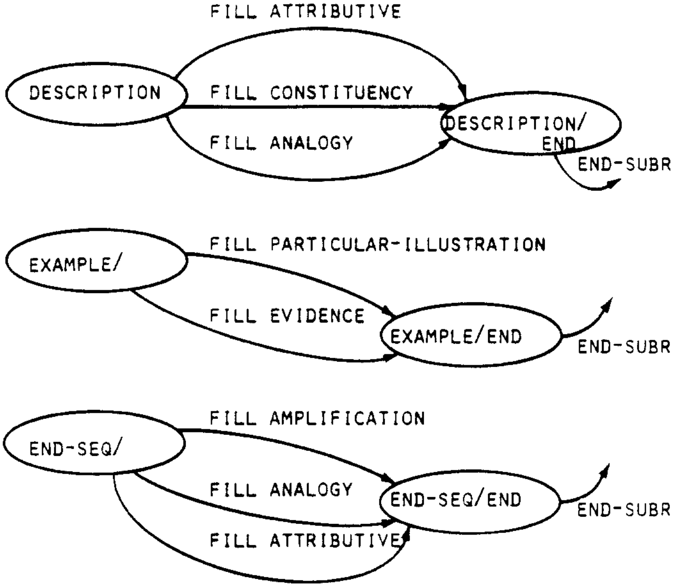


FIG. 11. Identification-schema sub-graphs.

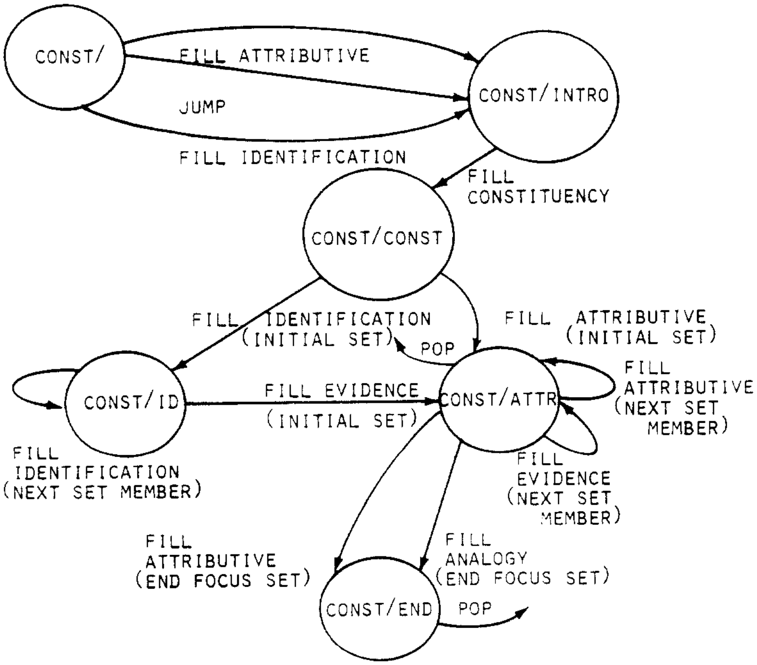

CONST/
FILL ATTRIBUTIVE
JUMP
FILL IDENTIFICATION
CONST/INTRO
FILL CONSTITUENCY
CONST/CONST
FILL IDENTIFICATION (INITIAL SET)
POP
FILL ATTRIBUTIVE (INITIAL SET)
CONST/ID
FILL EVIDENCE (INITIAL SET)
CONST/ATTR
FILL ATTRIBUTIVE (NEXT SET MEMBER)
FILL IDENTIFICATION (NEXT SET MEMBER)
FILL EVIDENCE (NEXT SET MEMBER)
FILL ATTRIBUTIVE (END FOCUS SET)
FILL ANALOGY (END FOCUS SET)
CONST/END
POP

FIG. 12. The constituency graph.

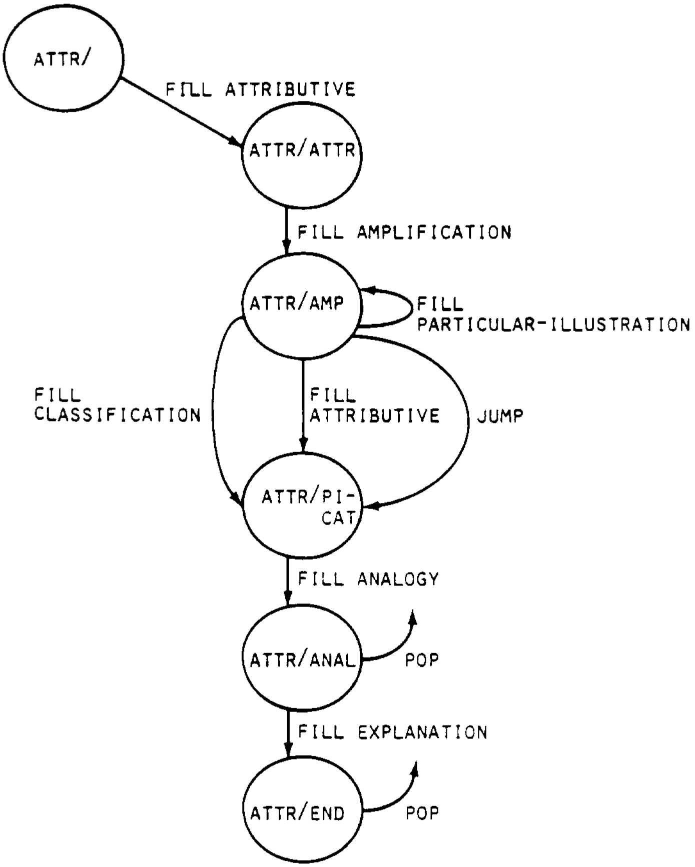


FIG. 13. The attributive graph.

The *compare and contrast* schema (Fig. 14) was modified to allow for equal discussion of the two items in question. The *contrastive* schema which emerged from the text analysis called for contrasting a major concept against a minor one. The minor concept, had, in most cases, either been discussed in the preceding text, or was assumed by the writer to be familiar to the reader. No history of discourse is currently maintained in TEXT and no user model other than a static one is constructed. Thus, the system does not know whether the user has more knowledge about one concept than another and the comparison must be equally balanced. Equal balance is achieved by first providing the similarities between the two objects and then presenting their differences.

The *compare and contrast* schema dictates a contrastive structure without specifying which predicates are to be used. To achieve this variation, the TEXT schema makes use of the three other schemata through recursion. A recursive call to a schema is indicated in the graph by a *push* arc. When another schema

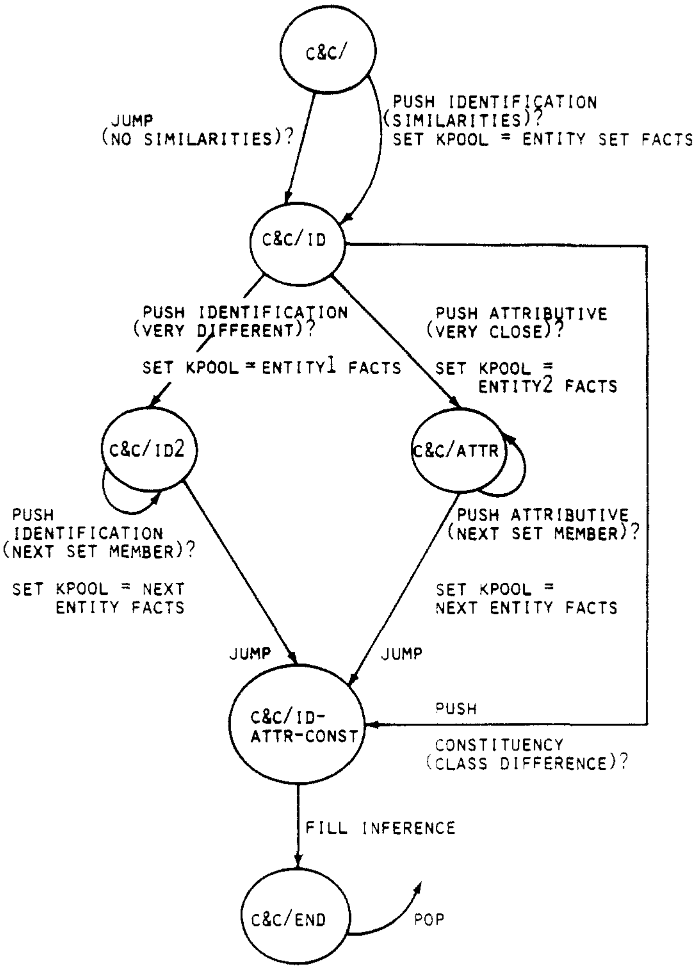


FIG. 14. The compare and contrast graph.

is invoked, the knowledge pool is reset to the information relevant for that portion of the response (this is indicated in Fig. 14 by the *set* action on each *push* arc).

The TEXT *compare and contrast* graph uses the *identification* schema to identify the similarities between the two objects. The schema that is used for the contrastive portion of the response depends on the semantic information available about the two entities. Either the *identification, constituency* or *attributive* schema can be used and this depends on whether the two entities are very different in concept, very similar, or in between (as determined by the tests on these three *push* arcs). This point is discussed further in the next section which describes the selection of a schema. The *compare and contrast* schema concludes with a direct comparison between the two entities via the inference predicate.

5.2.1. *Answering a question*

To answer a question, TEXT first selects a schema to guide the construction of the answer. An answer is then constructed by *filling* the schema.

5.2.2. *Selecting a schema*

In the TEXT system, association of strategy with discourse goal is achieved by associating the different schemata with different question-types. For example, if the question involves defining a term, different schemata are possible than if the question involves describing information. A summary of the assignment of schemata to question-types is shown in Fig. 15.

On the basis of the given question-type, the associated schemata are selected as possible structures for the response. A single schema is selected out of this set on the basis of the information available to answer the question. This is one case where semantic information interacts with information about discourse structure to determine the structure of the generated text.

In response to requests for definitions and information, the *constituency* schema is selected when the relevant knowledge pool contains a 'rich' description of the questioned object's sub-classes and less information about the object itself. This is true when the questioned object occurs at a higher level in the knowledge base hierarchy than a pre-determined level. Note that the higher an entity occurs in the hierarchy, the less descriptive information is available to describe the set of instances it represents since the larger the class, the less common features occur across it. When this is not the case, the *identification* schema is used for definition questions and the *attributive* schema is used for information questions. For the question "What is a guided projectile?" (Example 10) the *constituency* schema is selected since more information is available about the guided projectile's sub-classes than about the guided projectile itself, while the *identification* schema is selected for the question "What is a ship?" (Example 11).

The *compare and contrast* schema is always selected in response to a request

Requests for definitions
Identification
Constituency

Requests for available information
Attributive
Constituency

Requests about the difference between objects
Compare and contrast

FIG. 15. Schemata used for TEXT question-types.

about the difference between objects but the type of information in the relevant knowledge pool is used to determine which sub-schema path should be taken for the contrastive portion of the answer. When two objects are very close in concept (see [23]) the *attributive* sub-schema path is taken in order to describe detailed differences (Example 12).[12] When the two objects are very different, the *identification* sub-schema path is taken in order to describe generic differences (Example 13). In other cases, the *constituency* sub-schema path is taken.

Example 10.

```
(definition GUIDED)
;
; What is a guided projectile?
;
Schema selected: constituency
Message through dictionary. Entering tactical component
```

A guided projectile is a projectile that is self-propelled. There are 2 types of guided projectiles in the ONR database: torpedoes and missiles. The missile has a target location in the air or on the earth's surface. The torpedo has an underwater target location. The missile's target location is indicated by the DB attribute DESCRIPTION and the missile's flight capabilities are provided by the DB attribute ALTITUDE. The torpedo's underwater capabilities are provided by the DB attributes under DEPTH (for example, MAXIMUM OPERATING DEPTH). The guided projectile has DB attributes TIME TO TARGET & UNITS, HORZ RANGE & UNITS and NAME.

Example 11.

```
(definition SHIP)
;
; What is a ship?
;
Schema selected: identification
Message through dictionary. Entering tactical component
```

A ship is a water-going vehicle that travels on the surface. Its surface-going capabilities are provided by the DB attributes DISPLACEMENT and

[12] Note that, as currently implemented, the sub-schema paths do not have access to a discourse history of what has been included already in the text. This means that in selecting the information to contrast (for example, an entity's attributes), TEXT may select information for one entity that has not been included for another. This can lead to anomalies such as the last sentence of Example 12 where the DISPLACEMENT of the cruiser and ocean escort are compared although the cruiser's DISPLACEMENT was never stated. The inclusion of a discourse history is a topic for future work.

DRAFT. Other DB attributes of the ship include MAXIMUM SPEED, PROPULSION, FUEL (FUEL CAPACITY and FUEL TYPE), DIMENSIONS, SPEED DEPENDENT RANGE and OFFICIAL NAME. The DOWNES, for example, has MAXIMUM SPEED of 29, PROPULSION of STMTURGRD, FUEL of 810 (FUEL CAPACITY) and BNKR (FUEL TYPE), DIMENSIONS of 25 (DRAFT), 46 (BEAM), and 438 (LENGTH) and SPEED DEPENDENT RANGE of 4200 (ECONOMIC RANGE) and 2200 (ENDURANCE RANGE).

Example 12.

```
(difference OCEAN-ESCORT CRUISER)
;
; What is the difference between an ocean escort and a cruiser?
;

Schema selected: C & C-attributive

Message through dictionary. Entering tactical component
```

The cruiser and the ocean escort are surface ships. The ship's surface-going capabilities are provided by the DB attributes DISPLACEMENT and DRAFT. The ship has DB attributes MAXIMUM SPEED, PROPULSION, FUEL (FUEL CAPACITY and FUEL TYPE), DIMENSIONS, SPEED DEPENDENT RANGE and OFFICIAL NAME. Ocean escorts have a DISPLACEMENT between 3400 and 4100. All ocean escorts in the ONR database have REMARKS of 0, FUEL TYPE of BNKR, FLAG of BLBL, MAST HEIGHT of 85 and PROPULSION of STMTURGRD. Ocean escorts carry between 2 and 22 torpedoes, 16 missiles, and between 1 and 2 guns. A ship is classified as an ocean escort if the characters 1 through 2 of its HULL NO are DE. Cruisers have a PROPULSION of STMTURGRD and a LENGTH between 510 and 673. All cruisers in the ONR database have REMARKS of 0 and FUEL TYPE of BNKR. Cruisers carry between 8 and 42 torpedoes, between 4 and 98 missiles, and between 1 and 4 guns. A ship is classified as a cruiser if the characters 1 through 2 of its HULL NO are CL or the characters 1 through 2 of its HULL NO are CG. The ocean escort, therefore, has a smaller LENGTH and a smaller DISPLACEMENT than the cruiser.

Example 13.

```
(difference DESTROYER BOMB)
;
; What is the difference between a destroyer and a bomb?
;

Schema selected: C & C-identification

Message through dictionary. Entering tactical component
```

A destroyer is a surface ship with a DRAFT between 15 and 222. A ship is a vehicle. A bomb is a free falling projectile that has a surface target location. A free falling projectile is a lethal destructive device. The bomb and the destroyer, therefore, are very different kinds of entities.

5.2.3. *Filling the schema*

Each schema predicate has functions associated with it to define the type of information it can match in the knowledge pool. To construct a single proposition, the functions retrieve information from the knowledge base and format it in an internal representation. For example, in the database domain, one way to provide attributive information about an entity is through the use of database attributes. The attributive function, therefore, when passed an entity, retrieves the database attributes for that entity in the knowledge base and constructs a list containing the predicate, the entity, and the database attributes. This list is an internal representation of the proposition. The attributive proposition for the entity *ship* is shown below along with an eventual English translation. The first element in the list specifies that this is an attributive proposition, the second that this proposition identifies database attributes, the third identifies the entity to which the properties are attributed, and the remaining elements itemize the actual attributes.

> (attributive db SHIP (name OFFICIAL_NAME) (topics SPEED_DEPENDENT_RANGE DIMENSIONS) (duplicates (FUEL FUEL_TYPE FUEL_CAPACITY)) (attrs PROPULSION MAXIMUM_SPEED))
>
> Other DB attributes of the ship include MAXIMUM SPEED, PROPULSION, FUEL (FUEL CAPACITY and FUEL TYPE), DIMENSIONS, SPEED DEPENDENT RANGE and OFFICIAL NAME.

The predicate semantics thus defined for TEXT are particular to a database system and would have to be redefined if the schemata were to be used in another type of system (such as a tutorial system). The semantics are not particular, however, to the *domain* of the database. When transferring the system from one database to another, the predicate semantics would not have to be altered.

The schema is filled by traversing the graph, using the predicate semantics to select propositions from the relevant knowledge pool. Where several arcs emanate from a single state in the graph representation of the schema or where a single predicate matches more than one proposition in the knowledge pool, all propositions are retrieved and the focus constraints are used to select the most appropriate proposition, thereby specifying which arc is taken. When the arc is taken, the proposition is removed from the knowledge pool.

6. Focus of Attention

As noted earlier, schemata are only part of the mechanism that TEXT uses to determine the content and order of its generated text. In this section, we show how constraints on how focus of attention can shift from one sentence to the next are used to determine what to say next in cases where the schemata do not totally constrain the system's choice.

When producing a single utterance (as dictated by a schema), TEXT narrows its focus of attention to a single object in its pool of relevant information. Having made a decision about what to talk about first, TEXT must support that decision in succeeding utterances if it wants its text to be easily understood. That is, having decided to focus on a particular object(s), its utterances constrain the set of possibilities for what can be said next if the system is to avoid jumping around from one topic to another. These are termed *immediate focus* constraints since they apply locally between utterances.

TEXT uses constraints developed by Sidner [33] on how immediate focus can shift or be maintained. Sidner showed that speakers can either maintain their current focus, shift focus to an item just introduced, return to a previous focus, or focus on an item implicitly related to the current focus. These constraints are used to limit the information TEXT considers when deciding what to say next. If its discourse plan allows for several utterances, the system only considers propositions that can be focused in one of these ways.

Several problems arose in adapting Sidner's work to generation. Since it considered *interpretation*, there was no need to discriminate between members of the set of legal foci; when more than one possibility for immediate focus existed after a given sentence, the next incoming sentence would determine which of the choices was taken. While Sidner's constraints are sufficient for interpreting natural language, for generation a system must be able to decide which of the constraints is better than any other at any point.

A preference ordering on Sidner's constraints was developed for generation (see Fig. 16). The ordering suggests that a speaker should shift to focus on an item just introduced into conversation if possible. If the speaker chooses not to do so, that item will have to be re-introduced into conversation at a later point before the additional information can be conveyed. If, on the other hand, the speaker does shift to the item just mentioned, there will be no trouble in continuing with the old conversation by returning to a previous focus.

Several consecutive moves to items just introduced are not a problem. In fact, consecutive focus shifts over a sequence of sentences occurs frequently in

1. Shift focus to item mentioned in previous proposition
2. Maintain focus
3. Return to topic of previous discussion
4. Select proposition with greatest number of implicit links to previous proposition

FIG. 16. Ordering of focus constraints.

written text. If this rule were applied indefinitely though, it would result in never-ending side-tracking onto different topics of conversation. However, the model of generation assumes that information is being presented in order to achieve a particular goal (e.g., answer a question). Only a limited amount of information is within the speaker's scope of attention because of its relevance to that goal (as defined by the relevant knowledge pool). Hence only a limited amount of side-tracking can occur.

The second preference indicates that a speaker should continue talking about the same thing rather than returning to an earlier topic of conversation where possible. By returning to a previous discussion, a speaker closes the current topic. Therefore, having introduced a topic (which may entail the introduction of other topics), one should say all that needs to be said before returning to an earlier topic. That is, one avoids implying that the current subject has been completed when, in fact, there is more to be said. If neither of the first two preferences apply then the speaker must return to an earlier topic of discussion (Preference 3).

In cases where a speaker must choose between two propositions with the same focus, the preferences described so far proscribe no course of action. Rather than making an arbitrary choice, a speaker tends to group together in discourse information that is in some way related. When the system has a choice between two propositions with the same focus, it chooses that proposition with the most mentions to previously mentioned items (Preference 4).

This ordering doesn't dictate *absolute* constraints on the system. Just as a speaker may choose to suddenly switch topics, the system may choose to do so also. The ordered focus constraints are preferences which indicate the system's best move when faced with a choice. If the system's discourse plan indicates that no next choice meets these constraints, it will follow its plan making note of the abrupt switch in focus. This switch can then be syntactically marked by the tactical component to ease the transition for the user.

7. An Example of TEXT in Operation

To illustrate how schemata and focus constraints determine the content and organization of TEXT's output, consider the question "What is a ship?". The first step in generating a response is the construction of the relevant knowledge pool, immediately followed by the selection of a schema. For details on how the relevant knowledge pool is constructed (see [23]). Here, simply note that the area in the knowledge base immediately surrounding the questioned object (the *ship*) is selected. The resulting knowledge pool contains all associated information about the concept ship (including its database attributes, relations, and superordinates and subordinates in the knowledge base hierarchy). A diagram of the resulting pool is shown in Fig. 17. While the details are not important, the reader should note that only meta-level information is included in the pool.

To select a schema, TEXT first retrieves the set of schemata associated with the current question-type, a request for a *definition*, which includes both the *identification* and *constituency* schemata. The *identification* schema is selected since the ship occurs below a pre-determined level in the hierarchy.

After selecting the *identification* schema, TEXT begins traversing the schema graph. The first arc dictates that a proposition matching the identification

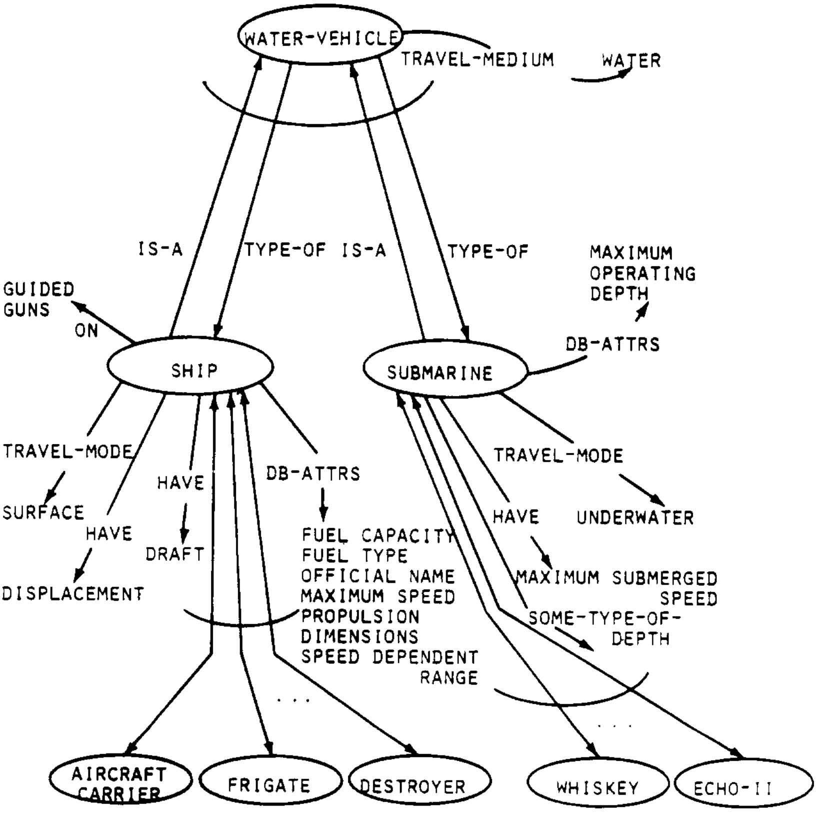


FIG. 17. Relevant knowledge pool.

predicate must begin the response. Accordingly, a proposition identifying the ship as a water going vehicle is selected from the relevant knowledge pool and is eventually translated as the first sentence of the generated response (shown in Fig. 18).

```
(definition SHIP)
;
; What is a ship?
;
Schema selected: identification
  1. A ship is a water-going vehicle that travels on the surface. 2. Its surface-going capabilities are
provided by the DB attributes DISPLACEMENT and DRAFT. 3. Other DB attributes of the ship
include MAXIMUM SPEED, PROPULSION, FUEL (FUEL CAPACITY and FUEL TYPE), DIMEN-
SIONS, SPEED DEPENDENT RANGE and OFFICIAL NAME. 4. The DOWNES, for example, has
MAXIMUM SPEED of 29, PROPULSION of STMTURGRD, FUEL of 810 (FUEL CAPACITY) and
BNKR (FUEL TYPE), DIMENSIONS of 25 (DRAFT), 46 (BEAM), and 438 (LENGTH) and SPEED
DEPENDENT RANGE of 4200 (ECONOMIC RANGE) and 2200 (ENDURANCE RANGE).
```

FIG. 18. "What is a ship?"

Having traversed the first arc, the system is now at state ID/ID. The schema dictates that the system now has two choices: it can either provide a Descriptive proposition (by following arc *subr* Description/) or it can jump to state ID/DS and provide an Example (by following arc *subr* Example/). Since the Description/ sub-graph has three arcs emanating from its initial state and the Example/ sub-graph has two arcs emanating from its initial state, TEXT has five arcs to choose from (representing the predicates analogy, constituency, attributive, evidence, and particular-illustration). One of these, the particular-illustration arc, is ruled out on the basis of information in the relevant knowledge pool compared with the semantics of the predicate. At this point, the relevant knowledge pool contains no information matching this predicate[13].

Since the schema does not constrain the system's choice for what to say next to a single proposition, the focus constraints are invoked to choose among the remaining arcs which match four propositions. (The four predicates, the matching propositions, the eventual translations of the propositions, and their foci are shown in Fig. 19.) Following the preferential ordering on how focus of attention should shift, TEXT first attempts to choose a proposition which allows it to shift focus to an element that was just introduced and is a potential

[13] The semantics for particular-illustration dictate that an example from the database can be extracted by instantiating database attributes already presented in the answer with values from the database. Since no database attributes have yet been mentioned, it is not possible to give an example at this point. Note that the schemata say nothing about what constitutes an appropriate example and this is one area where future work is required.

1. **Analogy**
   (analogy rels SHIP ON GUIDED GUNS)

   The ship carries guided projectiles and guns

   focus = *ship*

2. **Constituency**
   (constituency SHIP (AIRCRAFT-CARRIER FRIGATE OCEAN-ESCORT, CRUISER, DESTROYER)

   There are 5 types of ships in the ONR database: aircraft carriers, frigates, ocean escorts, cruisers, and destroyers.

   focus = *ship*

3. **Attributive**
   (attributive db SHIP (name OFFICIAL_NAME) (topics SPEED_DEPENDENT_RANGE DIMENSIONS) (duplicates (FUEL FUEL_TYPE FUEL_CAPACITY)) (attrs PROPULSION MAXIMUM_SPEED))

   The ship has DB attributes MAXIMUM SPEED, PROPULSION, FUEL (FUEL CAPACITY and FUEL TYPE), DIMENSIONS, SPEED DEPENDENT RANGE and OFFICIAL NAME.

   focus = *ship*

4. **Evidence**
   (evidence based-db SHIP (TRAVEL_MODE SURFACE) (HAVE DRAFT) (HAVE DISPLACEMENT))

   Its surface-going capabilities are provided by the DB attributes DISPLACEMENT and DRAFT.

   focus = *surface-going capabilities*

FIG. 19. Possible predicates, translations, and foci.

candidate for a shift in focus. The default focus[14] of all but the evidence proposition is on the concept 'ship', which is the focus of the previous proposition. The evidence proposition however, focuses on 'surface-going capabilities', a potential candidate for a shift, and thus it is selected as the next proposition for the text.

Note that the choice of the evidence predicate based on the focus constraints confirms the motivation for the rule that the system should shift focus if possible. It is quite natural for the system to return focus to the concept 'ship',

[14] Each predicate has a default focus associated with it which indicates the predicate argument that is most likely to be focused on. The default focus corresponds to the unmarked syntax associated with the predicating act. For example, the attributive predicate in its usual use attributes features to an entity or event. The unmarked use assumes an entity has been focused on: the entity is being talked about and some of its features are being described (see Sentence 1 below). The opposite case, of associating talked about features with a different entity is less usual (see Sentence 2 below):

1. The chimpanzee has fine control over finger use.
2. Fine control over finger use is also common to the chimpanzee.

which it does in sentence (3) of the text (Fig. 18) by including the attributive proposition it considered as a possibility for sentence (2). On the other hand, it would be awkward to shift at a later point in the text to the surface-going capabilities of the ship after continuing to focus on the concept 'ship'. This is a case where an opportunity to easily present information would be lost if it were not included at this point.

The process continues from state ID/EX until one of the *pop* arcs is taken and the *identification* schema is exited. At that point the full message has been constructed and is represented as a list of propositions. The tactical component is then invoked to transform the message into natural language and produce the final text as shown in Fig. 18.

To give a flavor of how this is done, the first two sentences of the answer are shown in proposition representation in Fig. 20 (further details on surface choice can be found in [23, 24]). Propositions are passed in this internal representation as input to the tactical component along with focus information, which includes the current focus of the proposition and its potential focus list (an ordered list of potential candidates for a shift in focus). The tactical component uses a dictionary to choose vocabulary for each argument of the proposition and to assign case roles. The predicate is always translated as the verb of the sentence and the choice of verb also determines which arguments of the proposition will fill the case roles of *protagonist*[15] and *goal*[16]. The representation of Proposition 2 at an intermediate stage of translation is shown in Fig. 21. Here, the verb 'provide' has been selected and the arguments '(HAVE DRAFT)(HAVE DISPLACEMENT)' have been assigned as the *protagonist* of the sentence and the arguments 'SHIP (TRAVEL_MODE SURFACE)' as the *goal*. The function 'entry-for' indicates that vocabulary for these arguments will also be chosen by accessing their entries in the dictionary.

```
Proposition 1:
(identification SHIP WATER-VEHICLE (restrictive TRAVEL-MODE SURFACE)
  (non-restrictive TRAVEL-MEDIUM WATER)
  (non-restrictive FUNCTION TRANSPORTATION))

Focus = SHIP

Proposition 2:
(evidence based-db SHIP (TRAVEL-MODE SURFACE)
  (HAVE DRAFT) (HAVE DISPLACEMENT))

focus = (TRAVEL-MODE SURFACE)
```

FIG. 20. Propositions 1 and 2.

[15] Often referred to as *agent*. We are following Kay's [15] terminology here.

[16] Often referred to as *object* of the sentence.

```
verb = = = provide

protagonist = (entry-for (HAVE DRAFT)(HAVE DISPLACEMENT)))

goal = possessive = (entry-for SHIP)
       nnp = (entry-for (TRAVEL-MODE SURFACE))
```

FIG. 21. Proposition 2 at an intermediate stage.

After vocabulary has been chosen for the remaining untranslated arguments, the grammar is invoked to fill in the syntactic details of the sentence and to order the constituents, producing the actual linear sentence. Here focus information is used to make appropriate surface choices. For example, since the concept 'ship' was focused in sentence (1), reference to it in the second sentence can be pronominalized, resulting in the choice of 'its'. Tests are also made on focus information to select the active or passive construction, or a construction known as *three-insertion.*[17] The rhetorical predicates provide information about the type of sentential connective that can be used. In this answer, the connective 'for example' in sentence (4) is selected on the basis of the particular-illustration predicate.

8. Related Research

By far, the majority of work done to date in language generation has addressed problems in the tactical component. This has included a system to directly translate a given semantic network into English sentences [34], Goldman's [10] work on lexical choice as part of the MARGIE system, Davey's [8] use of systemic grammar to generate commentaries on tic-tac-toe games, McDonald's MUMBLE [21] which includes a broad coverage of English syntax and employs a decision making process that takes into account many syntactic constraints on realization, and very recently, the work on NIGEL [18, 19] to develop a large linguistically justified grammar within a systemic framework. Without this earlier work in language generation, development of TEXT would have been nearly impossible as it draws on the results of this work for its own tactical component (in particular, the concept of dictionary developed by both Goldman and McDonald). These works have very little to say, however, about the issues that were of particular concern in TEXT: determining the content and organization of a text.

Work on planning and generation bears more closely on the problem of content. Cohen [7] addressed the problem of planning speech acts in response to a user's question. His system, OSCAR, could select a speech act and specify the propositional content of the act. Appelt [2] continued in this vein by showing that the planning formalism could be used for determining the lexical and syntactic structure of the text as well as the content. One of the major departures of

[17] E.g., There are two types of guided projectives in the ONR database: torpedoes and missiles.

Appelt's work is its refutation of the 'conduit metaphor'. While other generation systems have assumed a separation between the process of deciding what to say and how to say it, Appelt's work is based on the hypothesis that decisions made in the lowest level of the language generation process can influence decisions about what to say.[18] Both Appelt's and Cohen's work, however, deals with single-sentence generation for the most part. Their work does not address the problem of organizing it appropriately for text.

Two of the earlier systems that were capable of producing text emphasized not the problem of text generation, but the type of knowledge needed in order to produce appropriate text. Swartout [35] examined the problem of knowledge needed for generation in the context of a medical consultation system. He showed that knowledge conveniently represented in order to efficiently arrive at a medical diagnosis, may not allow for the generation of understandable explanations about the system's reasoning. He developed a representation appropriate for explaining the expert system's reasoning which was used for the generation of explanations. His main concern, however, was with the knowledge representation and not with the generation process.

Meehan was also interested in the problem of knowledge needed for generation as part of his work on the story-generation system TALESPIN [25]. TALESPIN was capable of producing simple short stories about persons (or anthropomorphic animals) making plans to achieve goals and their frustrations while achieving those goals. Meehan was most concerned with the planning aspects of the program and the knowledge needed to select plans for the characters, although his system could produce multi-sentence descriptions of the characters and their actions.

Mann and Moore [17] were interested in the specific problems that arise in the generation of multi-sentential strings. They developed the *Knowledge Delivery System* (KDS) which could produce a paragraph providing instructions about what to do in case of a fire alarm. Their system relies on hill-climbing techniques to produce optimal text and does not use knowledge about discourse structure. Another drawback to their system is the fact that it operates in the very limited domain of the fire-alarm system.

One advantage to KDS is its ability to do continual re-editing of the text to produce the final version. TEXT cannot evaluate its own text and clearly this is an important facility which must eventually be developed. KDS uses heuristics to do its re-evaluation, however, and has not been used to produce a wide range of texts. In contrast, TEXT makes its decisions about ordering on the basis of rhetorical strategies that are commonly used for particular discourse goals.

[18] TEXT is based on a generation model which does assume separation, a model that was adopted to allow focus on the problems of the strategic component. Some of the integration that Appelt proposes could be achieved by introducing backtracking between the two components. It should be noted, however, that while most researchers agree that there must be interaction between the two processes, exactly how that interaction should be achieved is still an open question.

Of previous work on text generation, Weiner's work [39] is most similar to TEXT. He is also interested in the structure of text, although he focuses on explanations in particular. He proposes an *explanation grammar* which is similar to our use of schemata in that it dictates what orderings of propositions are possible, it captures the hierarchical structure of text, and the kernels of the grammar (e.g., statement, reason) are at the same level of granularity as the predicates used for TEXT. Furthermore, he also incorporates the notion of focus of attention by maintaining a pointer to the proposition in focus at each point in the explanation.

Weiner proposes that a person may justify a statement in one of three ways:

(1) by providing a reason;

(2) by providing supporting examples;

(3) by providing alternatives, all of which are shown as inadequate except the alternative which supports the statement.

Thus, he uses basically four 'predicates' (*statement, reason, example, and alternative*), along with a number of subordinators such as *and/or* and *if/then.* Since explanations are frequently embedded, a statement followed by a reason may in turn function as the reason for another statement. To account for this, his grammar rules generate tree structures, which may be transformed by transformational rules, to generate the hierarchical structure representing the surface explanation. At each point in the explanation, one node of the tree is singled out as the focused node.

While the approach taken in TEXT is compatible with Weiner's, the theory of textural structure as captured in TEXT goes considerably beyond their formulation in the following ways: in TEXT, strategies are associated with different discourse goals, while in Weiner's system the grammars were developed only for justification; TEXT uses a greater number of predicates in schemata than does Weiner; the schemata used for TEXT capture a notion of variability which is resolved by other influences on the text, such as focus of attention and underlying semantic information; and finally, the specification of focus of attention and its interaction with the schemata is much more detailed in TEXT, as an argument of a proposition is identified as its focus, while in Weiner's work, an entire proposition is focused upon.

One other piece of work, from psychology, should also be mentioned. Rumelhart's story grammars [30] are similar to schemata as they describe textual structure for stories. He uses the grammars to recognize the underlying structure of a story, as opposed to generating it, and to suggest a memory organization which summarizes the important events of a story. Thus, his purpose in developing the grammars were different from ours. Rumelhart's grammars also differ from schemata in that they include both a structural and a semantic component, the non-terminals of the grammar (e.g., setting, episode, event) do not correspond to the rhetorical predicates used for TEXT, and he captures the structure of narratives, while we are more interested in the structure of descriptions.

In sum then, very little previous work touches directly on the issues addressed in TEXT. TEXT differs from those few that do by providing a detailed computational model of discourse strategies that interact with focus of attention to produce the final text.

9. An Evaluation of Responses Generated by TEXT

To illustrate how our generation principles have been successful in producing coherent texts, as well as to identify defects in the text, consider the text generated in response to the question "What is a guided projectile?" (shown in Example 14). This text is representative of the range of generation TEXT can produce.

Example 14.
(1) A guided projectile is a projectile that is self-propelled. (2) There are 2 types of guided projectiles in the ONR database: torpedoes and missiles. (3) The missile has a target location in the air or on the earth's surface. (4) The torpedo has an underwater target location. (5) The missile's target location is indicated by the DB attribute DESCRIPTION and the missile's flight capabilities are provided by the DB attribute ALTITUDE. (6) The torpedo's underwater capabilities are provided by the DB attributes under DEPTH (for example, MAXIMUM OPERATING DEPTH). (7) The guided projectile has DB attributes TIME TO TARGET & UNITS, HORZ RANGE & UNITS and NAME.

The overall structure of the text is dictated by the *constituency* schema and the paragraph is bracketed by discussion of the guided projectile (first and last sentence). The inner portion of the text relates the guided projectile to its sub-classes and discusses each of these in turn. The overall structure is thus nicely balanced, providing a clear organizational framework for the reader.

Now consider the content of the text. The system has correctly, although not surprisingly, included only information that is directly or indirectly related to the guided projectile. This is the result of using the relevant knowledge pool. More significant, of all the information that could have been included from information related to the guided projectile, the system has selected only that which directly supports its goal of defining the object. Of 11 pieces of information related to the guided projectile (Fig. 22), the system has chosen to include only 5 pieces of information (i.e., its superordinate (sentence 1), its sub-classes (sentence 2), and 3 of its attributes (sentence 7)). Of 26 pieces of information associated with the missile and at least that many for the torpedo, the system has chosen to select only 2 for the torpedo and 4 for the missile (1 defining attribute and 1 database attribute for the torpedo and 2 defining attributes for the missile). This is due partly to the use of the *constituency*

An English translation of these 11 pieces is:

1. The guided projectile is a self-propelled projectile.
2. Attributes relating to self-propulsion include FUSE TYPE (possessed by the torpedo).
3. The DB attribute SPEED INDICES (possessed by the missile) also indicate properties of self-propulsion.

4–9. It has 6 DB attributes associated with it (counted here as 6 pieces of information): HORZ RANGE & UNITS, TIME TO TARGET & UNITS, HORZ RANGE, HORZ RANGE UNITS, TIME TO TARGET, TIME TO TARGET UNITS.

10. It is carried by water-going vehicles.
11. There are two types of guided projectiles: missiles and torpedoes.

FIG. 22.

schema. It determines that the superordinate of the guided projectile should be identified, its sub-classes described, and that defining attributes of both the missile and torpedo should be included (sentences 3 and 4). The focus constraints also play a role in the selection of information. They ensure, for example, that when database attributes are selected for sentences 5 and 6, only attributes are selected that support the definitional attributes presented in the previous sentences.

The surface text is influenced by the focus constraints such that constructions are selected that increase coherency. In this particular text, the use of there-insertion in sentence (2) was selected on the basis of an introduction of a set into focus. Similarly, the passive is used in sentences (5) and (6) to allow continued focus on the missile and torpedo. In some texts, the use of a particular rhetorical technique will force the selection of a sentential connective. This does not occur in this text.

Many of the defects of the text are due to limitations in the surface text generator (i.e., the tactical component). The text could be improved, for example, by combining sentences (2) and (3), emphasizing the contrast (e.g., "The missile has a target location in the air or on the earth's surface while the torpedo has an underwater target location."). Alternatively, the switch in focus back to the guided projectile could be more clearly signalled, at the same time linking the proposition to previously conveyed information by using a phrasing such as "The DB attributes common to all guided projectiles include . . .". The generation of more sophisticated phrasings such as these requires further theoretical work for the tactical component, addressing the question of why these phrasings are preferable to others.

On the organizational level, improvement could be achieved by grouping together statements about the missile and statements about the torpedo. This would involve a change to the constituency schema so that the system could group together statements about an element of a set when more than one statement occurs. Such a change would allow the tactical component to pronominalize second references to both the missile and the torpedo, thus reducing some of the ponderous feeling of the text.

More significant improvements to the text can only be made by dramatically improving the capabilities of the system. Some of these facilities would include inferencing (e.g., if the system can recognize and state the target location of the missile and torpedo, it should be able to infer that although both weapons have a target location, the exact location differs), varying detail (e.g., do all readers need to know about missiles and torpedoes or would the information about the guided projectile alone be sufficient?–see [27]), and finer determination of relevance (e.g., if the reader already knows about bombs, a guided projectile should be compared against this existing knowledge, requiring the system to elaborate further on the fact that guided projectiles are self-propelled, while bombs are not).

10. Future Directions

One of the tenets of this research has been that the final structure of a text is influenced by interactions between different sources of information, one of which is a speaker's knowledge about usual strategies for communication as encoded by the schemata. In the TEXT system as currently implemented, the final structure of the text is also influenced by the semantics of what is to be said (affecting which schema is selected for the answer and which predicates of the schema can be instantiated) and constraints on how focus of attention can shift. One influence on the final structure of the text which was not taken into account is a model of the user. Information about the user could be taken into account when determining which alternative to follow in a schema (thus, a proposition could be selected if it was determined to be most appropriate for the given user). An analysis is currently being made of the ways in which information about a particular user affects the level of detail needed in a response [24, 27].

Another current direction of research is an analysis of how and when global focus can shift in conjunction with the recursive use of the schemata. This is also expected to rely in part on a model of the user. Other open questions for the use of focusing concerns the nature of the mechanisms for maintaining and shifting immediate focus. More complex structures may be needed for some situations. For example, a speaker may introduce an item into conversation, but specify that he will continue to talk about it at a later point (see [14]).

11. Conclusions

The use of rhetorical strategies and a focusing mechanism provides a computationally tractable method for determining what to include in a text and how to organize it. This goes beyond earlier work in generation by examining production *strategies* for satisfying a particular goal (in this case, responding to one of three classes of questions). The generation method described not only provides methods for determining high-level choices about order and content, but it also provides information that can be used by the tactical component to make decisions about various surface level choices.

The use of schemata to encode knowledge about discourse structure embodies a computational treatment of rhetorical strategies that can be used to guide the generation process. This reflects the hypothesis that the generation process does not simply trace the knowledge representation to produce text, but instead uses communicative strategies that people are familiar with. This has the consequence that the same information can be described in different ways for different discourse purposes.

The use of a focusing mechanism as well as the schemata illustrates how the final structure and content of a text is influenced by an interaction between structural and semantic constraints. In the TEXT system, semantic constraints are provided by the relevant knowledge pool. It constrains the possible content of the text to a small subset of the entire knowledge base. A preferential ordering on immediate focus constrains possibilities for further utterances on the basis of what has already been said. The interaction of focus constraints with the schemata allows for the construction of a greater variety of paragraph-length responses to questions about database structure.

ACKNOWLEDGMENT

Special thanks goes to my advisor, Dr. Aravind K. Joshi and to Dr. Bonnie Webber for their advice and guidance in pursuing this work.

REFERENCES

1. Allen, J.F. and Perrault, C.R., Analyzing intention in utterances, *Artificial Intelligence* **15** (1980) 143–178.
2. Appelt, D.E., Planning natural language utterances to satisfy multiple goals, Ph.D. Dissertation,, Stanford University, Stanford, CA, 1981.
3. Bossie, S., A tactical component for text generation: sentence generation using a functional grammar, Tech. Rep. MS-CIS-81-5, University of Pennsylvania, Philadelphia, PA, 1981.
4. Chafe, W.L., The flow of thought and the flow of language, in: T. Givon (Ed.), *Syntax and Semantics, Discourse and Syntax*, **12** (Academic Press, New York, 1979).
5. Chen, P.P.S., The entity-relationship model – towards a unified view of data, *ACM Trans. Database Systems* **1** (1) (1976).
6. Codd, E.F., Arnold, R.F., Cadiou, J.-M., Chand, C.L. and Roussopoulos, N., Rendezvous Version 1: An experimental English-language query formulation system for casual users of relational databases, Tech. Rept. RJ2144(29407), IBM Research Laboratory, San Jose, CA, 1978.
7. Cohen, P., On knowing what to say: Planning speech acts, Tech. Rept. No. 118, University of Toronto, Toronto, Ont., 1978.
8. Davey, A., *Discourse Production* (Edinburgh University Press, Edinburgh, 1979).
9. *Encyclopedia Americana* (Americana Corporation, New York, 1976).
10. Goldman, N.M., Conceptual generation, in: R.C. Schank (Ed.), *Conceptual Information Processing* (North-Holland, Amsterdam, 1975).
11. Grimes, J.E., *The Thread of Discourse* (Mouton, The Hague, 1975).

centering, in: *Proceedings Seventh International Joint Conference on Artificial Intelligence*,

12. Grishman, R., Response generation in question-answering systems, in: *Proceedings 17th Annual Meeting of the Association for Computational Linguistics*, La Jolla, CA (August 1979) 99–102.
13. Hobbs, J., Coherence and coreference, SRI Tech. Note 168, SRI International, Menlo Park, CA, 1978.
14. Joshi, A. and Weinstein, S., Control of inference: role of some aspects of discourse structure – centering, in: *Proceedings Seventh International Joint Conference on Artificial Intelligence*, Vancouver, BC, August 1982.
15. Kay, M., Functional grammar, in: *Proceedings Fifth Annual Meeting of the Berkeley Linguistic Society*, Berkeley, CA, 1979.
16. Malhotra, A., Design criteria for a knowledge-based English language system for management: an experimental analysis, Rept. MAC TR-146, MIT, Cambridge, MA, 1975.
17. Mann, W.C. and Moore, J.A., Computer generation of multiparagraph English text, *Amer. J. Comput. Linguistics* **7** (1) (1981).
18. Mann, W.C., An overview of the Nigel text generation grammar, in: *Proceedings 21st Annual Meeting of the Association for Computational Linguistics*, Cambridge, MA (1983) 74–78.
19. Matthiesson, C.M.I.M., A grammar and a lexicon for a text-production system, in: *Proceedings 19th Annual Meeting of the Association for Computational Linguistics*, Stanford, CA (1981) 49–56.
20. McCoy, K.F., Automatic enhancement of a data base knowledge representation used for natural language generation, M.S. Thesis, University of Pennsylvania, Philadelphia, PA, 1982.
21. McDonald, D.D., Natural language production as a process of decision making under constraint, Ph.D. Dissertation, draft version, MIT, Cambridge, MA, 1980.
22. McKeon, R., *The Basic Works of Aristotle* (Random House, New York, 1941).
23. McKeown, K.R., Generating natural language responses to questions about database structure, Ph.D. Dissertation, Tech. Rept. MS-CIS-82-5, University of Pennsylvania, Philadelphia, PA, May 1982.
24. McKeown, K.R., Focus constraints on language generation, in: *Proceedings Eighth International Joint Conference on Artificial Intelligence*, Karlsruhe, Germany (August 1983) 582–587.
25. Meehan, J.R., TALE-SPIN, an interactive program that writes stories, in: *Proceedings Fifth International Joint Conference on Artificial Intelligence*, Cambridge, MA (August 1977). 91–98.
26. Moore, R.C., Reasoning about knowledge and action, SRI Tech. Note No. 191, SRI International, Menlo Park, CA, 1980.
27. Paris, C., Determining the level of expertise of a user in a question answering system, Tech. Rept., Columbia University, New York, 1984.
28. Paterson, J., *The Hamlyn Pocket Dictionary of Wines* (Hamlyn, New York, 1980).
29. Rich, E.A., User modeling via stereotypes, *Cognitive Sci.* **3** (1979) 329–354.
30. Rumelhart, D.E., Notes on a schema for stories, in: D.G. Bobrow and A. Collins (Eds.), *Representation and Understanding: Studies in Cognitive Science* (Academic Press, New York, 1975) 211–236.
31. Shipherd, H.R., *The Fine Art of Writing* (Macmillan, New York, 1926).
32. Shipley, L., Transcripts of mother-child dialogues, Unpublished manuscript, University of Pennsylvania, Philadelphia, PA, 1980.
33. Sidner, C.L., Towards a computational theory of definite anaphora comprehension in English discourse, Ph.D. Dissertation, MIT, Cambridge, MA, 1979.
34. Simmons, R. and Slocum, J., Generating English discourse from semantic networks, *Comm. ACM* **15** (1972) 891–905.
35. Swartout, W.R., Producing explanations and justifications of expert consulting programs, Tech. Rept. MIT/LCS/TR-251, MIT, Cambridge, MA, January 1981.
36. Tennant, H., Experience with the evaluation of natural language question answerers, Working Paper 18, University of Illinois, Urbana-Champaign, IL, 1979.

37. Terkel, S., *Working* (Avon Books, New York, 1972).
38. Thompson, H., Strategy and tactics: a model for language production, in: *Papers 13th Regional Meeting Chicago Linguistic Society*, Chicago, IL, 1977.
39. Weiner, J.L., BLAH, A system which explains its reasoning, *Artificial Intelligence* **15** (1980) 19–48.
40. Wilensky, R., A knowledge-based approach to language processing: a progress report, in: *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, Vancouver, BC (August 1981) 25–30.
41. Williams, W., *Composition and Rhetoric* (Heath, Boston, MA, 1983).
42. Winograd, T., *Language as a Cognitive Process: Syntax*, Vol. I (Addison-Wesley, Reading, MA, 1983).
43. Woods, W.A., Transition network grammars for natural language analysis, *Comm. ACM* **13** (10) (1970) 591–606.

Received January 1983; revised version received April 1984