The Realization of Natural Language with Pragmatic Effects

Murray Watt

Technical Report CSRI-215
August 1988

Computer Systems Research Institute
University of Toronto
Toronto, Canada
M5S 1A1

The Computer Systems Research Institute (CSRI) is an interdisciplinary group formed to conduct research and development relevant to computer systems and their application. It is jointly administered by the Department of Electrical Engineering and the Department of Computer Science of the University of Toronto, and is supported in part by the Natural Sciences and Engineering Research Council of Canada.

The Realization of Natural Language with Pragmatic Effects

by

Murray Watt

Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
August 1988

A Thesis submitted in conformity with the requirements
for the degree of Master of Science

Abstract

Realization is the process that takes a conceptual representation of a text to a linguistic representation. Realization is not a simple mapping from a text plan to natural language, because it adds pragmatic content to the text. This thesis includes a principled account of what aspects of meaning should be derived from the text plan and what should be drived from control in realization. From this account a model of realization is developed that distinguishes between texts with the intent of generating an optimal realization. The model is implemented in a system called ELOQUENCE that organizes the choices available to the realization process and measure the appropriateness of texts for a given set of pragmatic goals. ELOQUENCE separates text planning from realization. However, processing constraints apply when most appropriate and do not necessarily reflect module boundaries.

Acknowledgements

The opportunity to study computer science at the University of Toronto has been an intellectual privilege. I am especially indebted to Graeme Hirst for laboriously checking successive iterations of my work, and for being tolerant of my tendency to fluctuate from extreme view to extreme view. I also thank John Mylopoulos for additional comments.

Yves Lespérence and Evan Steeg stand out in my mind for inspiring by their pensive nature. I thank Judy Dick for encouragement, Barb Brunson for linguistic advice, Daniel Lyons and Mark Ryan for technical advice, and Stephen Harris for help in implementing the morphological component. I also enjoyed working with Chrysanne DiMarco, with whom I share some research interests. Finally, I thank Tu-Tran Banh for support and patience.

Financial support was gratefully received from the Natural Sciences and Engineering Council.

Contents

Chapter 1

Introduction

What is the role of a natural language generator? In the past, the answer to this question might have been "converting computer output to natural language". This reflected a concern for the user and his or her needs in contrast to the computer and its function. At first, data structures were used as *text templates*, but this required anticipating what degree of detail and what knowledge were appropriate to a specific audience. With this practical experience, the answer might have been "converting a fragment of the system's *knowledge* into text". Advances in discourse research led to the *dynamic* generation of text, but with this experience researchers realized that language is highly situation-dependent. Thus we have the modern phrasing, which states that the task of a natural generator is to map *communicative intent* into linguistic form. This gives us a statement sufficiently vague to encompass the complexity of human cognition, leaving us trying to design a generation system to cope with an indefinitely large number of partially interdependent pragmatic concerns.

These concerns straddle three processing objectives, which are independent of approach or architecture. The first is to select the content of the text. That is, a generator given a model of a situation including the system's own knowledge, its beliefs about the user's beliefs, and the context of the discourse must determine what information needs to be communicated, given a set of goals.

The second concern is to structure this information. More precisely, the information must be bound and ordered by rhetorical techniques for communication and the purpose of the text. In the sense that most of the planning tasks are subsumed by these two objectives, *the strategic level* is an appropriate term for these two objectives, although the name owes its origin to a particular choice of architecture.

The third objective of a generation system, *realization*, is to create natural language text from the ordered information. Since the first generators only had to produce short passages (usually single sentence responses for *question-answering* applications), most early work dealt solely with realization. The systems of Simmons and Slocum [1972] and Wong [1975] are examples of realization-based generation. They relied on linguistic knowledge encoded as an augmented transition network (ATN). The system would be given a fragment of a semantic network (or some other equivalent representation) to traverse, and a side-effect of this traversal would be a natural language utterance. Then, interest in applications requiring multisentence passages shifted the focus of research to issues of content selection and discourse planning.

Unfortunately, the role of the realization process has not been redefined with respect to these new applications. Therefore, we are left asking what a realization component should do, and how it should interact with the rest of the system. Thus, I shall develope a modern definition of the realization process to best suit the aims of text generation.

However, the problems of generation extend beyond processing strategies. Inherent weaknesses in current approaches to generation result mainly from current systems' lack of sensitivity to the hearer and the situation. McDonald, Vaughan, and Pustejovsky [1987, page 160] affirm this view when they state

> We recognize that many of the limitations on the competence of today's generators — their expressive capacity to aptly use a wide range of syntactic constructions and vocabulary — stem not from the generators' design but from limitations of the underlying programs to conceptually model and represent the range and richness of situations that people can.

However, it is not enough just to discriminate between situations. A generator must be able to distinguish differences in texts and relate these differences to situations. The successful characterization of situations, texts (from their communicative intent to their linguistic form), and relationships between the two will eliminate the *ad hoc* aspect of text generation. Thus, I shall examine what variations should be under the control of the realization process and to describe how the pragmatic constraints of a situation percolate down to realization.

In the remainder of this chapter, I provide a brief account of the current relationship of generation research to the rest of computational linguistics, an overview of several major generation systems that shall be examined in depth, and an overview of the architecture proposed in this thesis. I then examine the role of the realization process.

1.1 Generation Versus Interpretation

The recognition that many applications of natural language processing require flexible output has caused much activity in text generation research. However, this does not change the fact that natural language understanding has been a subject of research longer than generation has been. As a result, people are less familiar with the issues of language generation. Moreover, many mistakenly view generation as merely the inverse process of interpretation.

In the sense that generation is the mapping of communicative intent to linguistic form, generation *is* the inverse process of interpretation. Yet today's understanding systems do not attempt to recover any substantial part of a speaker's intent. Instead, most systems try to recover the direct propositional structure of a text by recognizing features (usually syntactic) that the system designer felt were significant to meaning.[1]

I do not refute the arguments that demonstrate the efficiency or the necessity of the bidirectionality for natural language processing devices. Jackendoff [1987] argues that both understanding and production make use of common information structures and mechanisms for coordination between levels of representation. For example, deciding how to structure a text has a clear and important counterpart in determining rhetorical relations in a text. However, current research often states the problems of modeling the understanding process in a manner that does not further research in generation. In

[1]Possibly, what makes generation an interesting area for research is that one cannot easily avoid real issues when building a generation system. While an understanding system is evaluated only by its behavior, the shortcomings of a generation system are directly visible in the text it produces.

particular, current models of understanding are only directed at a subset of the psychologically and linguistically motivated processing abilities. Even so, it is not clear that this limiting of understanding research necessarily simplifies the task. Feedback in current systems may be viewed as weakly modeling the missing direction of co-ordination.

However, the inverse process perspective can only be taken with a super-competence model. That is, although conceptual content, speaker goals, world knowledge, and context are concerns for both processes, competence in breadth and depth varies between the two. Understanding requires drawing inferences from a model of the discourse situation to a) resolve ambiguity, b) interpret what has or has not been expressed, and c) limit the number of hypotheses to be pursued. Generation must cope with a large number of pragmatic concerns, while creating an appropriate text. Understanding and generation are problems of a different character.

Consider the utterance, "Robert goes to the nightspots with the prettiest girls." An interpreter would try to decide between the interpretations of Robert going to places to find women, or Robert always having women accompanying him. In this example, the natural language system is dealing with a mapping from a single utterance to one of two distinct meanings. The possible counterpart for generation might be realizing how much ambiguity the utterance will create, if it is concerned about the clarity or the humor of a text. This aspect can usually be ignored, because people can disambiguate language well.

A generator has to consider subtle differences between utterances. For instance, the two utterances, "I do not believe you are telling the truth" and "I think you are lying" can be said to have similar meanings. However, it is the role of the generator to select an appropriate realization with the required directness. The possible counterpart for understanding might be using the manner in which the information was conveyed to help determine an agent's goals or to shift perspective.

However, although the basis of inquiry differs, certain research is relevant to both generation and understanding. This research clearly includes:

1. Descriptive theories of syntax, morphology, and phonology: Both generation and understanding are based the same linguistic structures, which thus form the most

obvious link between the two.

2. Knowledge representation: Not only is knowledge representation important, because it is believed that understanding and generation should share common information structures, but also because, like many problems in AI, those of natural language processing reduce to problems of deficiency in the underlying representations and their maintenance.

3. Descriptions of various discourse phenomena: Anaphora, focus, and rhetorical structure are topics of discourse research that have already contributed to both areas.

4. Models of language users in a dynamic context: Modeling situations is a special problem of representation, which deals both with what is relevant information that can be extracted from context, and with an agent's beliefs and how they change over time.

Thus, generation may share the same schemes for representation as understanding. It may even be that many of the processes pertaining to generation may correspond exactly to processes pertaining to understanding. However, the problems that limit the performance differ between the two.

1.2 An Overview of Related Research

Despite the short history of natural language generation as a research area, there are several notable generation systems including KAMP [Appelt 1985a, 1985b, 1985c, 1988], MUMBLE [McDonald 1983, 1985; McDonald, *et al* 1985a, 1985b, 1985c], PAULINE [Hovy 1987a, 1987b, 1988b], PENMAN [Matthiessen 1984a, 1984b, 1987; Mann 1984, 1987; Mann and Thompson 1987a, 1987b; Hovy 1988a], and TEXT [McKeown 1982, 1983, 1985a, 1985b]. These systems are introduced in this section.

To provide practical experience, I have been developing a system of text realization called ELOQUENCE, which I introduce in this chapter and discuss at length later in this thesis.

Some of the formalisms chosen in the implementation of ELOQUENCE, namely conceptual graphs [Sowa 1984] and Government and Binding [Sells 1985], were also used by Veronica Dahl in her work [Dahl, *et al* 1986a, 1986b; Brown, *et al* 1987], but from a different perspective. Therefore, it is appropriate for me to compare her work to mine. However, Dahl used a highly restricted domain[2], and thus avoided many of the processing issues that were central to this thesis.

Most of the question-answering systems of the 1970s do not merit extended case studies in this thesis. However, examples of such systems can be found in Simmons and Slocum [1972] and Wong [1975].

1.2.1 KAMP

The KAMP system [Appelt 1985a, 1985b, 1985c, 1988], generates natural language with a general planning mechanism, a feature that allows for the planning of non-linguistic actions together with linguistic actions. The system's model of a situation strongly affects generation of text by determining what actions are needed to achieve a goal and how the system refers to items.

Specifically, KAMP is a multi-agent planning system that creates plans to influence agents. The system starts with assertions about the initial situation. These assertions include information about what each agent knows and the relative locations of objects and agents. Then, by identifying with one agent, the system formulates a plan to achieve a goal state based on that agent's intentions.

KAMP adopts the following techniques to handle the problems of an exponential search space for plans[3], interacting subgoals, and resource allocation:

1. Actions are defined over a hierarchy with high-level actions that describe entire activities and low-level actions that describe detailed steps. Therefore, only a limited number of actions need be considered at a given time.

[2] Veronica Dahl *et al* have used conceptual graphs as an interlingual representation within the domain of error message generation. Government and Binding theory together with discontinuous grammars are used to map to utterances in an appropriate natural language.

[3] Given n possible actions, there are n^m plans of size m.

2. While the model's axioms describe actions precisely and thus are used to verify plans, *action summaries* (summaries of an action's preconditions and effects) guide the search for a plan. Well-designed action summaries are equivalent to distributing frame axioms over the hierarchy, thus allowing for non-linear expansion of the plan.

3. Critics are used to modify plans to account for global interactions of effects and to enforce efficient use of resources.

Since all goals over the hierarchy are treated in the same manner, the processing objectives are not handled by separate modules. In particular, linguistic decisions are distributed over the entire planning process.

1.2.2 MUMBLE

MUMBLE [McDonald 1983, 1985; McDonald, *et al* 1985a, 1985b, 1985c] takes as input a conceptual specification for a text. The specification is an instance of a rhetorical plan fleshed out with propositions that describe speech acts or key lexical items for communication. Since a specification is a type of text plan, the system can be said to represent only part of the generation process. However, this allows it to be used with a variety of applications with the addition of a component that co-ordinates what the application wants to express with the kinds of data structures MUMBLE expects. Clearly, the system is based on a strong separation of the processing objectives.

MUMBLE builds a linguistic representation of the text while traversing a specification. As more of the specification is processed, the new information is spliced into the linguistic representation. The location of the splice is called an attachment point. A list of active attachment points is maintained by the attachment process. MUMBLE can choose between several active attachment points in the list for each piece of information.

MUMBLE uses *realization classes* [McDonald and Pustejovsky 1985] to select between different syntactic forms. Figure 1.1 illustrates the type of choices available to it.

Realization	Examples
A *verb* B	Rome invaded England.
B *passive verb* by A	England was invaded by Rome.
A *gerundive* B	One of the most interesting events in history was Rome invading England.
B *passive gerundive* by A	England being invaded by Rome was one of the most interesting events in history.
A *nominal* of B	The Roman invasion of England was one of the most interesting events in history.
The *passive nominal* of B by A	The invasion of England by Rome was one of the most interesting events in history.

Figure 1.1: Possible Realizations in MUMBLE

1.2.3 PAULINE

To achieve expressive effects, PAULINE [Hovy 1987a, 1987b, 1988b] was directed at adding versatility to generation of text. That is, a single conceptual representation ought to be able to be expressed in various ways, by being sensitive to pragmatic aspects (interpersonal and situation-specific) of the situation. Table 1.1 and 1.2 from [Hovy 87b] summarizes the pragmatic aspects of the situation PAULINE addresses while generating text.

The pragmatic aspects do not determine the generated text directly. Rather, they activate rhetorical goals. For example, the *depth of acquaintance*, *relative social status*, and *atmosphere* determine what level of formality is desirable.

The most salient feature of PAULINE is its ability to deal with this large amount of situational information and translate into appropriate control of the generation process.

PAULINE recognizes that low-level realization decisions have significant pragmatic effects. To take advantage of difficult-to-anticipate opportunities, the system does *planning on demand.* That is, enough text planning is done to start realizing, and PAULINE then continues when the realization component needs further guidance.

PAULINE does not treat all goals the same. Some goals can be fully achieved (such as making an agent aware of a wrench on a table). Others can never be fully achieved (such as formality). Therefore, PAULINE does two types of planning, *prescriptive* and *restrictive.*

The effects of prescriptive planning (top-down planning) are typically long range. Thus, top-down planning is mainly for the planning tasks closest to topic collection and organization, where the goals are achievable and interact co-operatively.

Restrictive planning keeps track of all goals, even conflicting goals. The satisfaction of each goal is constantly monitored. Also, decisions are determined by evaluating the available options with respect to how they affect each goal. Thus, restrictive planning is mainly used with those tasks closest to the realization process, where the goals must be maintained throughout the text. PAULINE resolves conflicts of the restrictive planner's goals using a least-satisfied strategy. That is, the system choose from its options to satisfy the goal with the lowest satification rating.

Conversational atmosphere (setting):

Aspect	Possible characterizations
time	much, some, little
tone	formal, informal, festive
conditions	good, noisy

Speaker:

Aspect	Possible Characterizations
knowledge of the topic	expert, student, novice
interest in the topic	high, low
opinions of the topic	good, neutral, bad
emotional state	happy, angry, calm

Hearer:

Aspect	Possible characterizations
knowledge of the topic	expert, student, novice
interest in the topic	high, low
opinions of the topic	good, neutral, bad
language ability	high, normal, low
emotional state	happy, angry, calm

Speaker-Hearer Relationship:

Aspect	Possible Characterizations
depth of acquaintance	friends, acquaintances, strangers
relative social status	dominant, equal, subordinate
emotion	like, neutral, dislike

Table 1.1: Situatio-temporal information

Hearer:

Aspect	Possible characterizations
affect hearer's knowledge	teach, neutral, confuse
affect hearer's opinions of topic	switch, none, reinforce
involve hearer in the conversation	involve, neutral, repel
affect hearer's emotional state	anger, neutral, calm
affect hearer's goals	activate, neutral, deactivate

Speaker-Hearer Relationship:

Aspect	Possible characterizations
affect hearer's emotion toward speaker	respect, like, dislike
affect relative status	dominate, equal, subordinate
affect interpersonal distance	intimate, close, distant

Table 1.2: Interpersonal information

1.2.4 PENMAN

PENMAN [Matthiessen 1984a, 1984b, 1987; Mann 1984, 1987; Mann and Thompson 1987a, 1987b; Hovy 1988a] has been under development since the early 1980's and represents the consequences of many people's research. Thus, even without considering the research issues addressed by PENMAN's creators, the sheer magnitude of the system is impressive. For example, the system includes an implementation of a systemic functional grammar (NIGEL) that manages a large coverage of English supported by a network of over 250 systems.[4]

PENMAN includes the following four modules:

1. An acquisition module: This module searches a knowledge base for information relevant to a communicative goal. Although considered a preprocess to text planning, the planner can request additional information from the acquisition module

[4]For an explanation of the functional approach to English, see Halliday 85.

to satisfy subgoals, if the planner has used all the information it was given.

2. A text planning module: This module is a hierarchical planner that tries to create a plan that best covers all the relevant information. Plans are built from small general-purpose recursive schemas called RST relations. A schema is given a name reflecting the relations between a field called a nucleus and one or more fields called satellites. A schema definition consists of a name, constraints on the fields (preconditions), constraints on the relationships between between fields, and the effects of the schema.

3. A sentence generation module: This module is based on the systemic grammar previously mentioned. Like ATN's, the systemic grammar encodes linguistic knowledge in a process. The grammar is a network of systems of alternatives. Each set of alternatives is monitored by a *chooser* that directs inquires to the environment (anything external to this module, including the knowledge base, the text plan, and the discourse module). On the basis of the result of the inquiry, the chooser the system selects between alternatives.

4. An improvement module: This module has not been implemented. Its purpose is to revise the text plan based on an evaluation of aspects of text that are difficult to anticipate, such as readability.[5]

1.2.5 TEXT

TEXT [McKeown 1982, 1983, 1985a, 1985b] focuses on content selection and text planning and shows how focus can be used in generation of discourse. However, TEXT uses established mechanisms and procedures for realization. In particular, McKeown's method of realization is based on McDonald's work and related research [McKeown 1985a, page 6]. Yet McKeown rejects the traditional approach that posits realization as a separate post-process; she designed TEXT in such a way that minor changes would allow backtracking.

[5]Vaughan and McDonald [1986] describe an alternative use of revision, which is to reduce the number of constraints considered by a given process. Although the approach raises many questions, it seems to have a similar character to the distribution of frame axioms in hierarchical planning.

If a proposition could not be expressed syntactically at a particular point, then TEXT would back up and alter the conceptual plan.[6]

Content selection consists of recursive instantiation of schemata such as *compare and contrast*, *illustrate by example*, and *analogy*. Propositions are selected from the part of the knowledge base that is in global focus by the rhetorical rules encoded in the schemas and immediate focus.

1.3 An Overview of ELOQUENCE

The realization system developed in this thesis, ELOQUENCE, emphasizes the options available to the process and the process's control over them. The system is designed to interact with a text planner. ELOQUENCE would provide the planner with an evaluation of how well it can satisfy a set of restrictive goals, while the planner provides the following information as data and control:

1. A representation of the ordered information conveyed by a section of text plan.
2. A pointer into the text plan marking the point to which the plan has stabilized. Thus, the realization module commits itself to all decisions whose effect does not extend beyond that point.
3. A dynamic specification of the rhetorical goals (and their relative importance) for realization. The specification can change both in detail and nature.
4. The maximum number of structural options to maintain. ELOQUENCE maintains multiple possible realizations. However, the number of all possible realizations grows exponentially with the size of the plan. (A paragraph might have over a thousand variations.) I shall discuss discuss a number of methods for selecting which realizations to pursue, and how well each method searches for an optimal realization.

[6]This raises many questions. Is a proposition the unit of realization that determines when a plan has failed? Is the failure to realize a proposition the only feedback the realization module should provide. How does one fix a plan? From transcriptions of human speech it is clear that a speaker does considerable critiquing and replanning. Should a generator of written text act on failed plans and edit the result as the plan is debugged?

The difficulty with maintaining choice is that once realizations diverge, they confront different systems of options. To consider all realizations in parallel would be a major control problem. Thus, ELOQUENCE makes each realization a separate process. Each consists of a partial linguistic structure, a mapping between the structure and the text plan, an evaluation of the structure relative to the rhetorical goals, and a status (either a priority, "waiting for more planning", or "killed").

The ELOQUENCE also consists of a constructor, which maps parts of the plan to linguistic options, an evaluator, which judges how well a structure meets rhetorical goals, and an executive, which controls the process.

1.4 What is Realization?

Since the work of McKeown [1982, 1983, 1985a, 1985b] *realization* has been defined as the process that produces natural language text for a given content and discourse structure. This definition has two major shortcomings.

First, there is no accepted definition for this use of *content*. In examining systems that have a separate realization module, one gets the notion that content is a representation with a model-theoretic interpretation. Typically, the input representation is assigned interpretations with respect to an application. This practice is consistent with the semanticists' definition for the meaning and the content of a structure. Meaning is what is common to all uses of a structure and content is the structure's meaning with all the information that contributes to a particular use. However, the semanticists' stance is much too powerful for this use of content. The realization process makes decisions based on pragmatic constraints. Thus realization adds content to the input representation, which should not be viewed as the complete and specific content of the generated text.

Second, the current definition of realization implies that the realization process is no more than a mapping from content to natural language. It does not provide any specification for the interleaving of planning and realization. However, realization should provide feedback to the text planner.

Although many generation systems are built with a distinct *realization* module, the current definition is sufficiently vague that the particulars vary from system to system.

However, by comparing these systems it is possible to describe a prototypical realization module.

Unfortunately, a definition of the realization process based solely on *family resemblance* among a subset of system architectures would lack integrity. If there were a theory of generation a definition would follow. Instead, I have developed a model based on the tasks universally (or at least prototypically) associated with realization and the tasks they entail.

1.4.1 Generation and Meaning

The generation systems that have a separate realization module evolved from the early question-answering systems of the 1970s. Many of the weaknesses of today's systems can be traced back to the original architectures.

Early systems would generate a representation of a sentence with the appropriate interpretation relative to the model (usually a database). In turn, this representation would be mapped to natural language by a *grammatical* component. Similarly, post-process realizers map an entire discourse representation to multi-sentence passages. There is no relationship between the structure of the text and the selection of its content.

The systems of the 1970s were motivated by the linguistic theory of *autonomous syntax*. In particular, the tools of formal syntax and semantics are used to separate the rules that determine whether a sentence is well-formed from the rules of semantic interpretation. Some researchers who object to this approach argue that syntax should not be processed separately from semantics. However, theoretical modularity does not require one way constraining. Processing can constrain when most appropriate, not necessarily at module boundaries. It is legitimate to attack the concept that meaning is the denotational significance of a representation. The theory of speech acts [Searle 1969], which addresses language use, takes a different approach to meaning, but is compatible with a model-theoretic approach.

Some linguists, including Halliday [1985], believe that systemic grammar better models language use than a model-theoretic approach. Halliday cites PENMAN as a formalization of the systemic approach. Although PENMAN is a remarkable system, it does not capture meaning any differently than any other generation system. Recall that PENMAN is based on a systemic grammar [see section 1.2.4]. Instead of manipulating symbols that are mapped to a model, choices are made by inquiries to the *environment*. However, PENMAN's inquiries are directed to its knowledge base, text plan, and discourse model, which are given model-theoretic interpretations.

I recognize that language use is central to meaning. However, the procedural import of a systemic grammar or a Lisp function is not the set of possible outputs. Procedural import is better described as a relationship between states [Smith 82].

Jackendoff's Model of Language Processing

This section provides the linguistic motivation for characterizing human language generation as a mapping between levels of representation. In the next section I shall discuss the failures of this approach to adequately characterize the computational process.

Jackendoff [1987] discusses the evidence that led linguists to believe that there are autonomous levels of representation that range from speech signals on one extreme to what he calls meaning on the other. He than explores the relationships between these levels and the linguistic information they encode.

A level of representation consists of a primitive set of distinctions, plus a finite set of principles of combination. Different levels of representation can encode different information or the same information with different *grain sizes* of representations. In the latter case a primitive at one level may be "exploded" into more primitive concepts at another level.

I shall outline the argument he makes. To justify the claim that there are different levels of language representation, Jackendoff begins by presenting evidence for distinctions in speech sounds made by humans (phonological primitives) and their principles of combination. He than repeats the process with lexical and phrasal categories (syntax primitives). The principles of combination constrain the organization of the primitives. The resulting phonological structure is not compatible with the resulting syntactic structure.

Then, Jackendoff argues that the processing must be bidirectional for generation and

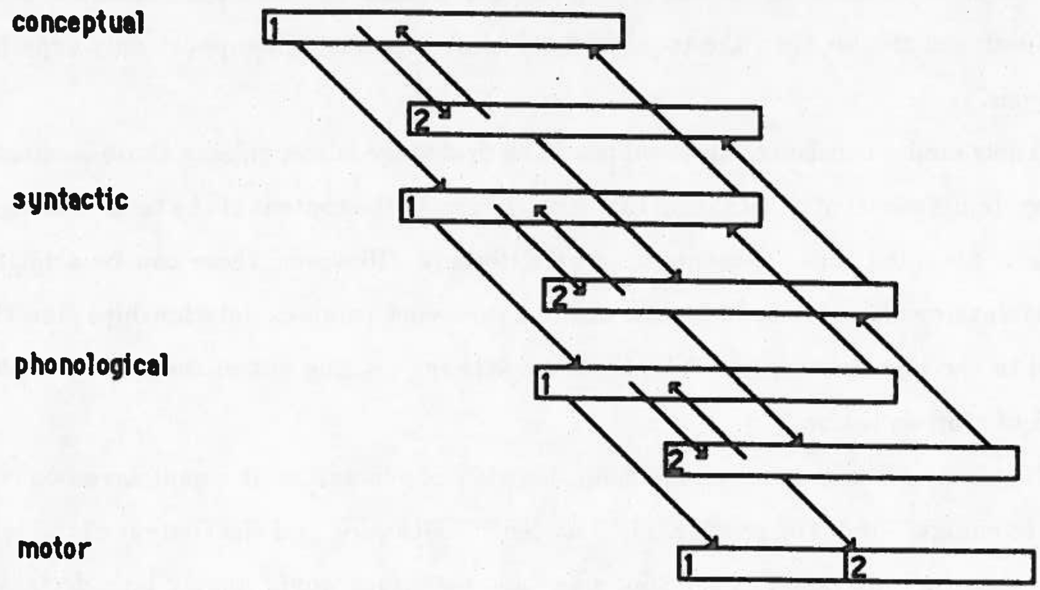


Figure 1.2: Human Language Production

understanding. This allows for both the top-down and bottom-up influences that are exhibited by human language processing.

Figure 1.2 illustrates the four levels Jackendoff presents for language production and their relationships. The conceptual level corresponds to a representation of the standard notion of semantic information. The motor level encodes distinctions in muscle activations that produce language.

Characterizing language by a set of representational levels allows us to directly recognize distinctions along different but not necessarily orthogonal dimensions.

1.4.2 Generation as a Mapping Process

McDonald, Vaughan, and Pustejovsky [1988] made an initial proposal describing a reference model for computer generation. Such a model would enable the comparison of alternative architectures.

They recognize that the varying focus of research projects makes it difficult to directly translate the systems into a reference model. Instead they propose an "abstract" model. My major concern with this approach is that if each of these projects have such a distinctive character then the required level of abstraction will support only superficial analysis.

Their model consists of three stages. The first stage is recognizing those elements of the system's model of its situation that are relevant to the content of the text. The second stage is mapping those elements onto an utterance. However, there can be arbitrarily many intermediate levels of representation involving complex relationships like those cited in the previous section. The third stage is the reading out of the most peripheral level of representation.

If this model were taken as the definitive view of generation, it would have two major shortcomings. First, the relationship between the situation and the content of the text is too strong. In one extreme, the content of any utterance would merely be a declaration of the state of affairs. In the other extreme, the process of "identifying relevant features" performs a much more complex role than their model leads one to believe.

Second, control (processing) is assigned the relatively minor role of efficient mapping between levels of representation. In other words the model emphasizes structures and only vaguely addresses pragmatic issues.

If generation were characterized as the creation of plans with linguistic actions then pragmatics would have a major role. However, from this perspective the situation would be too loosely related to content of the natural language produced, unless it were supported by a full theory of rational action and belief.

1.4.3 The Content of Linguistic Actions

In this section I relate the procedural import of the generator both to the system's model of the situation and to the representational content of the text. Thus I present an indirect relationship between the situation and the natural language produced. In the next section these concepts are used to define realization.

Instead of relating a situation directly to a natural language utterance, let us model the behavior of a generation system. The system's model of the situation directly affects the system's actions. As the system performs actions to achieve a set of goals, the situation and the system's goals change. Part of the change results from the actions of the system and part from the actions of other agents.

While maintaining their model of the situation, current generation systems ignore the possibility that their actions might not have the intended effects and other agents might be trying to achieve their own goals (which may conflict with those of the system). This simplifies planning and is reasonable for planning written text. More formally, all actions change the state of the world and the system's model of the situation in which it is embedded has a causal role on its actions.

Consider the following scenario. I am rushing to get my thesis to the binder's before the place closes and someone is blocking my path. I must decide between several courses of actions including: a) quietly saying "I would like you to please get out my way", b) shouting "Get out my way! I am coming through!", and c) bashing the person with a handy two-by-four. According to the semanticists' definition all three actions have content, because they arise from my relationship to the situation. Moreover, the three are communicative. However, the first two are special for more than their non-violent nature. Performing either is a linguistic act and creates a representation (a syntactic object). The first utterance represents a current desire. The second utterance represents two events I expect to occur in the near future. Although bashing is communicative, it does not represent anything.

Searle [1979] points out that not all speech acts have propositional content. For example, the para-linguistic utterance "Arrrgh" (often used by *Peanuts* characters) has no propositional content. It does have communicative content. Is it representational? I might try to force it into the schema by interpreting the utterance as a representation of an emotional state. However, the content of "Arrrgh", like the content of bashing, is determined by its use in a situation, but it does not directly represent that situation. More likely, the statement is part of the context from which we build an impression of the agent's state.

Now imagine that I am playing charades. After I have completed a series of complicated actions, you shout "Gone with the Wind!" Unless I acted out part of the story, which was followed by you classifying my actions, I have created a representation using symbols that I believed were meaningful to you, and you gave them a semantic interpretation.

For a final scenario, imagine I am being anti-establishment and say "Red isn't really red." From the statement you determine that I believe socialism as defined by Marx is dead. The statement only metaphorically represents my model of the situation and the direct propositional content cannot be true in any possible world. Yet the content relationship still remains.

Unfortunately, except for the work of Cohen and Levesque [1985] and Appelt [1985a, 1985b], there has been little research that provides theoretical insights into the relationship between the semantics of an utterance and its use. In most systems the relationship is encoded by the particular axiomatization of the system's hierarchy of actions.

In summary, a natural language generator system, like any intelligent system, has a representation of its situation that has a causal role in its behavior. Actions resulting from what is believed contribute to the change of the situation by creating with intent a linguistic structure. The semantic content of the linguistic structure can be derived from an agent's model of the situation and his goals.

1.4.4 A New Definition of Realization

Let us define natural language generation as a process that for a dynamic set of goals and a representation of the situation performs a sequence of primitive actions that are linguistic. This definition is only slightly different than that of section 1.1. One can define various kinds of generation by restricting the class of actions that can be performed and limiting the system's knowledge of the situation. For example, a generator of written text is limited to actions that produce written language and must reason about its effects on a prototypical reader. The definition can also be relaxed to allow actions such as pointing during conversations.

Since ELOQUENCE does not perform its own text planning, consider the following example from [Appelt 1988]. In the example there are two agents, a pump, a platform, a table, a tool, and a tool box. The names of the agent are Rob and John. They both know that the pump is bolted to the platform and the pump and platform are on the table. They also know that the tool is called a *box-end wrench* and that the tool box is under the table. However, only Rob knows that the box-end wrench is in the tool box. Since they are both initially in the same location, they both know that each other's location is in the vicinity of the pump. KAMP identifies with Rob, who has the goal that the pump be removed from the platform.

KAMP starts with a plan of John removing the pump form the platform with the precondition that John wants to remove it. The additional action of KAMP requesting that John remove the pump is added to the plan, which results in John removing the pump. The preconditions for John removing the pump include John being in same location the pump and John having the tool to remove it. Thus KAMP decides to inform John that the tool needed to remove the pump is the box-end wrench, which motivates John to get the tool. To enable John to gain possession of the wrench, KAMP decides to inform John that the wrench is in the tool box.

KAMP decides what information needs to be included in each utterance. For example, it decides that for the request that John remove the pump to be successful, the utterance must convey the predicate *remove* and activate the concepts of the pump and the platform. Activating the concept of the pump requires communicating enough properties about the pump to distinguish it from all other individuals relevant to the situation. Eventually KAMP generates the message "Remove the pump with the wrench in the tool box."

Notice that the high-level action of *requesting John remove the pump* does not specify the particularities of the utterance. However, it does capture its effects. On the other hand, activating the concept of the pump has a direct effect on the utterance. A generation system should be able to say "Remove the pump with the wrench. It's in the tool box." The same concepts and predicates are used in both utterances.

The low-level actions commonly associated with realization preserve the *propositional* content of the natural language. I don't mean that a propositional structure has to exist, only that the natural language produced by any refinement of the plan can be given the same semantic attribution. Thus, realization decisions are pragmatic, but not semantic.

(This is probably why they are restrictive [Hovy 1988].)

If the hierarchy of actions is properly axiomatized, then any realization will produce the effects prescribed by higher-level actions. Choices in realization provide additional pragmatic affect. The generator can take advantage of this by prescribing pragmatic goals for realization. If the realization goals are critical then their satisfaction will affect planning.

Since realization does not require the full hierarchy of the text plan, it can be summarized in terms of its semantic content. Thus, I shall define realization as the process that given the propositional structure of the text and a set of pragmatic goals makes decisions of linguistic form and determines the effect of those decisions on the system's goals. The particulars of what information should and should not be encoded in the propositional structure are discussed in chapter 3.

One can define various kinds of realization by restricting the variation in the propositional structure and the amount of look-ahead in the structure the realization process can have.

Chapter 2

Representation

In this chapter I examine some of the semantic complexities that must be addressed to support a useful coverage of natural language. Research in this area often addresses the proper treatment of a limited number of phenomena at one time. For example, at the 26th Meeting of the Association for Computational Linguistics papers were presented on the semantic treatment of plural noun phrases [Scha and Stallard 1988], comparative constructions [Manny and Rayner 1988], and verbal modifiers [Karlin 1988]. However, such an approach would not be suitable in designing a natural language generation system. Instead I examine a few of the more important issues in enough detail to achieve the desired coverage.

In implementing ELOQUENCE I chose to use Government and Binding Theory and conceptual graphs. Although the results of this work do not depend on a particular formalism, some understanding of both GB and conceptual graphs is needed to follow the examples in this thesis. This chapter also provides an overview of both formalisms. However, for a complete introduction to GB see Sells [1985] and for conceptual graphs see Sowa [1984]. For a detailed description of GB see Chomsky [1984, 1985, 1986]

2.1 Temporal Knowledge

Although the adverbs "before", "after", "during", and "while", like tense, can be used to specify time intervals, tense also reflects the mood of utterance, which is a function of the linguistic structure realized. The interrogative and imperative moods are used with the corresponding surface speech acts [see section 2.3]. The indicative and subjunctive moods are used with declarative statements.

Other temporal adverbials, such as the indices "tomorrow", "in 1983", and "for six years", convey a different type of information. The indices "tomorrow" and "in 1983" are for time intervals with respect to some date line, and "for six years" is a measure of a time interval's duration.

To determine the appropriate tense to use, a generator must determine the relationships between temporal intervals. James F. Allen has described how intervals can relate, using a set of seven possible relationships (and their inverses) [Allen 1983,1985]. These relations are summarized in Figure 2.1.

Relationship	Equivalent Relations on Endpoints
t before s	$t_b < s_a$
t equals s	$(t_a = s_a) \wedge (t_b = s_b)$
t meets s	$t_b = s_a$
t overlaps s	$(t_a < s_a) \wedge (t_b > s_a) \wedge (t_b < s_b)$
t during s	$((t_a > s_a) \wedge (t_b \leq s_b)) \vee$ $((t_a \geq s_a) \wedge (t_b < s_b))$
t starts s	$(t_a = s_a) \wedge (t_b < s_b)$
t finishes s	$(t_a > s_a) \wedge (t_b = s_b)$

Let t_a and t_b, and s_a and s_b represent the lesser and greater endpoint of the temporal intervals t and s respectively.

Figure 2.1: Allen's Temporal Relations

2.1.1 Interval Verses Point Representation

Some intensional logics assign a proposition a truth value relative to a point in time using operators P and F, which correspond to "it was the case in the past that" and "it will be the case in the future that". To examine these more closely, let ϕ be a proposition that is composed without any tense operators and let ψ be any well-formed proposition. ϕ is considered true at time t, if ϕ is assigned a value of true at t. A proposition $P\psi$ is considered true at time t if ψ is assigned a value of true at some time t' such that $t' < t$. Similarly $F\psi$ is considered true at time t if ψ is assigned a value of true at some time t' such that $t' > t$. Thus, the variable t serves as an anchor by which all other "times" are interpreted. A first reasonable intuition would have us assign to t a semantic value of the time an utterance is spoken or a written message is inscribed. This is often called the coding time. However, there are situations, such as in narratives or in pre-recorded messages, when this anchor is projected to some other time. For example, sentences [1] and [4] are anchored relative to the time of inscription, while sentences [2] and [5] are projected to the time of reception. This is often called the receiving time.

1. I am writing this letter, because I am yearning to see you.
2. I wrote this letter, because I was yearning to see you.
3. *I wrote this letter, because I am yearning to see you.
4. This show is being recorded today, August 24th, to be broadcast to you next week.
5. This show was recorded last week, August 24th, to be broadcast to you today.
6. *This show was recorded now, August 24th, to be broadcast to you today.
7. *This show is being recorded, to be broadcast to you today. (This incorrectly implies that the show is being broadcast the same day it is recorded.)

The error in [3] is subtle and one might likely overlook it, in the heat of the moment. Since the "writing" and the "yearning" linked in the same sentence, there is an explicit relationship between them. However, there is a shift in point of view resulting in "writing" and "yearning" being anchored relative to two different times. In order to prevent confusion, I shall refer to the temporal aspect of the deictic center (the speaker-context-time_triplet) as the speaker's temporal point of view.

2.1.2 Representing Temporal Information

We have seen that we need to represent temporal information relative to the speaker's point of view. As well, we often need the temporal relationships between propositions. However, not all temporal relationships need to be known. For example, only the temporal information relative to the speaker's point of view is needed to generate sentence [8], while several additional relationships are needed to generate sentence [9].

8. I have visited Athens, I have visited Cairo, but my life will not be complete until I have seen Mecca.

9. I was reading when she rang, so I put my book down, rushed downstairs, and answered the door.

Can the F and P operators express the range of temporal information desired? Hamblin[1] has shown using collapsing rules there are fourteen tenses that can be represented by sequences of F and P operators.[2] In addition, if these operators are embedded deeply within disjunctions of propositions embedded in turn within the tense operators, many temporal relationships can be represented.

Although they can represent complex temporal information, F and P cannot express the perfect tenses. This result follows from Kamp's theorem [Gabbay, *et al* 1979] that F and P cannot define the *since* or *until* operators, and from Gabbay's theorem that the expressive power of *since* and *until* is the same as the perfect tenses [Gabbay, *et al* 1979]. To resolve this problem, Kamp introduced *since* and *until* as operators. Each operator takes propositions as arguments. Kamp has shown that these two operators are sufficient to express the maximal temporal power of natural language. However, if I were to use *since* and *until*, I would have to explicitly define the relation between each pair of propositions that may occur in the same sentence. Moreover, because

[1]cited in Van Benthem[1985]

[2]Hamblin has four tense operators: F, P, G, H. However, $F\phi \Leftrightarrow \neg G \neg \phi$, and $P\phi \Leftrightarrow \neg H \neg \phi$.

since and *until* are binary predicates, one needs propositions that are vacuously true at particular times to represent the relationship between a proposition and the speaker's point of view. For example, if ϕ represents the proposition *today*("Murray and Tu-Tran moved to Toronto"), and ψ represents the proposition *resident*("Murray and Tu-Tran","Toronto"), *since*(ϕ, ψ) has the same truth value as "Murray and Tu-Tran have been living in Toronto." Such a scheme may be satisfactory to represent a speaker's knowledge, but the generator is expected to select the information to be communicated.

As an alternative approach, one could replace F and P with a single operator, $D^{[t_a,t_b)}$. $D^{[t_a,t_b)}\phi$ is true when ϕ is true for the interval from t_a to t_b. In order that one cannot have a proposition either both true and false or neither true nor false, one must consistently include one endpoint, t_a and t_b, but not the other. Figure 2.1 shows how endpoints can be used to determine the relationships between intervals.

Adverbials such as "at exactly twelve", "today", and "in 1603" represent points of time with different grains of representation. All such adverbials have fixed interpretation relative to either a calendar or the speaker's point of view and thus need to be considered on a case-by-case bases.

2.2 Quantification

Montague [Dowty, *et al* 1981] developed semantic representations for many natural language phenomena, including the expression of the determiners "no", "a", "the", and "every". Each determiner represents a function from properties of individuals to sets of individuals. The advantage of this approach is that proper scoping can be achieved. For example, both readings of "Some man loves every woman" can be represented, as illustrated in Figure 2.2. The basic mechanism for all determiner definitions is λ-abstraction. The λ symbol specifies a generic property as a formal parameter. The ˆ symbol provides the correct intensionality, but can be ignored for the simplicity of discussion.

Expression	
$\lambda P[\lambda Q \exists x[P\{x\} \wedge Q\{x\}]]$	"some"
$\lambda P[\lambda Q \forall x[P\{x\} \rightarrow Q\{x\}]$	"every"
$\lambda Q \exists x[man'\{x\} \wedge Q\{x\}]$	"some man"
$\lambda Q \forall x[woman'\{x\} \rightarrow Q\{x\}]$	"every women"
$love'(\hat{}\lambda Q \forall x[woman'\{x\} \rightarrow Q\{x\}]])$	"loves every women"
$\lambda Q \exists x[man'\{x\} \wedge love'(Q\{x\})]$	"Some man loves"

Reading One:

$$\exists x[man'\{x\} \wedge love'(\hat{}\lambda Q \forall x[woman'\{x\} \rightarrow Q\{x\}])]$$

"Some man loves every women." or
"There is a man who loves all women."

Reading Two:

$$love'(\hat{}\lambda Q \forall x[woman'\{x\} \wedge Q\{x\}]])(\hat{}\lambda Q \exists x[man'\{x\} \rightarrow Q\{x\}])$$

"Some man loves every women" or
"For each woman there is a man who loves her."

Figure 2.2: The Two Readings of "Some man loves every woman."

It is easy to show that this approach can be used for a larger set of determiners. For example, "exactly two" can be represented by the function:

$$\lambda P[\lambda Q \exists x \exists y [\forall z [P\{z\} \leftrightarrow z = x \vee z = y] \wedge x \neq y \wedge Qx \wedge Qy]$$

and "at least three" by the function:

$$\lambda P[\lambda Q \exists x \exists y \exists z [P\{x\} \wedge P\{y\} \wedge Pz \wedge x \neq y \wedge x \neq z \wedge y \neq z \wedge Qx \wedge Qy]].$$

Unfortunately, we run into problems with utterances such as "Many men are bald", "Many men in the hospital are bald" and "Many women in the hospital are bald", because determiners such as "many" are context-sensitive. As a first approximation, we can have the number of individuals required depend on the properties of the determiner's arguments. For example, one might decide that many is at least four for "Many men are bald", at least three for "Many men in the hospital are bald", and at least two for "Many women in the hospital are bald." Fuzzy set theory might be a good formalism to capture the numerical aspect of the information. However, it is the context-sensitivity of the quantifiers that is problematic.

2.3 Surface Speech Acts

This section address two issues. First, where in a generation system should the selection of surface speech acts be made? Second, how can the content of the speech acts be represented propositionally?

2.3.1 Selection of Speech Acts

At first, it would seem that the choice of surface speech acts is one of realization. After all, it seems that one can account for the difference in "Will you open the door?" and "Please, open the door" in terms of variation in realization. The concept of the hearer, the concept of the door, and the predicate "open" are active in both of these utterances.

However, the commonality resides in the beliefs of the agents. The speaker's goals must be derived from the utterance and the context. The utterance "Do you know what time it is?" takes on different readings depending on whether it is being said to someone who is late, or being said by someone who does not have a watch.

Implicit in a variation in the type realization of a speech act is a variation in the speaker's goals. If a particular type of act cannot be realized with a specific propositional content under a set of pragmatic constraints then the underlying text plan must be revised. Thus, the categorical choice between declarative, interrogative, and imperative statements should be made by the text planner.

2.3.2 Interrogatives

There has been some work in representing the propositional content of interrogative utterances, resulting in two basic theories; the *propositional approach* and the *categorical approach.*[3]

In the propositional approach, a question represents set propositions. A wh-question such as "Who is boring?" denotes the a set whose members are the propositions denoted by "Sadie is boring", "Liz is boring", "Hank is boring", and so on for all individuals. The set denoted by yes/no-questions, like "Is Sadie boring?", has as members the denotations of "Sadie is boring" or "It is not the case that Sadie is boring".

The idea of the categorical approach is that the propositional content of the whole question-answer sequence corresponds to a proposition. Put differently, questions are functions from categorical answers to propositions, with different categorical answers, such as noun phrases or verbs, belonging to different types of questions. For example, "Who is John's friend?" could be represented by $\lambda X_{NP}(X_{NP}$ is John's friend).

After presenting the two basic approaches, Bäuerle [1979] put forth a mixed approach, with wh-questions represented by a categorical approach and yes/no questions by a

[3] Bäuerle 1979 cites the work of Egli 1974, 1976, Hamblin 1973, Hausser 1976, 1977, 1978, and Karttunen 1977a, 1977b.

propositional approach. In addition, Bauërle shows that alternative questions, such as "When did you see him, yesterday or the day before?", can be handled in the same way as yes/no questions.

2.3.3 Imperatives

Like interrogatives, the propositional content of an imperative does not correspond simply with some state of the world. That is, an imperative need not be understood or obeyed.

To account for this one can introduce a function, *let*. (A more semantically explicit name might be *ordered.to.do.by.speaker*.) In relational terms *let* is a two-place predicate over agents and actions for a given time. For example;

- "Open the door." is a realization of
  "$D^{[t1,t2]}(let(\hat{}open'(\hat{}the(\hat{}door')))(HEARER)))$"

- "Let John open the door." is a realization of
  "$D^{[t1,t2]}(let(\hat{}open'(\hat{}the(\hat{}door')))(j)$"

Notice, that an order can be projected through the hearer to a third person (or to the speaker). In some languages, such as ancient Greek, there are first and third person imperatives, but in English we have to settle for a degenerate form.

2.4 Government and Binding Theory

The aim of Government and Binding theory is to use a minimal number of general principles to provide a descriptive coverage of natural language. Even without further justification this is a valid scientific method (Occam's Razor). However, GB gains a great deal of additional utility, because it can explain language acquisition.

Even if one were to approach computational linguistics from an engineering perspective, there is merit in using GB theory. First, it has an excellent coverage of written text. Second, it has the advantages of being modular. Third, the general principles and sub-theories have epistemological significance, which is evident from their predictive power.

Government and Binding Theory was initially developed by Chomsky and is a descendant of Transformational Grammar. However, the transformational nature has diminished importance in the new formalism and many other aspects of the earlier theory have changed dramatically. As the theory is further revised to account for new data we can expect the changes to be less dramatic. The central properties of the theory are stable, because the general principles and sub-theories provide the theory with modularity and a hierarchy of abstraction.

Since X-bar theory and θ-theory are critical to understanding the example in this thesis, I shall provide a description of both followed by a brief overview of GB.

2.4.1 X-bar theory

Since Chomsky presented his early work thirty years ago, people have tried to explicitly characterize language as a set of *phrase-structure rules*. These rules state the composition of constituents of a language. Some constituents can be composed of other constituents or a single word (or morpheme). The following are phrase-structure rules, which describe the possible compositions of a prepositional phrase and a sentence, respectively:

PP → P NP

S → NP VP

X-bar theory tries to abstract away from the particulars of the syntactic categories, creating cross-category templates for characterizing phrasal structure.[4] Figure 2.3 shows the scheme used in current GB.

The X in figure 2.1 is an abstraction away from a syntactic category. It could be replaced by any category, such as V (verb) or N (noun). X is called the head, because it gives the essential character to the structure (defines the function of the phrase). X′ and X″ are the first and second projections of X respectively. In particular, if we replace X

[4] Beside introducing regularity, this scheme introduces the restrictions that constrain the phrase-structure rules.

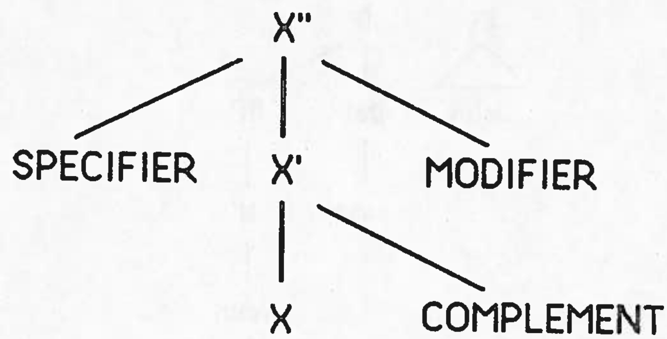


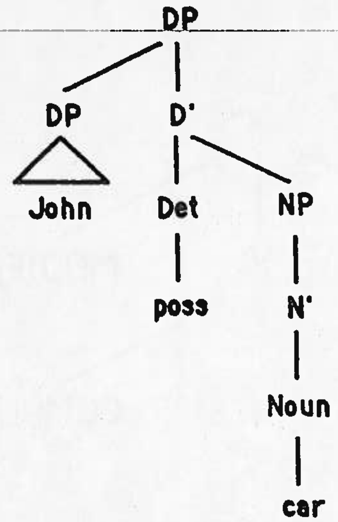
Figure 2.3: A Second-Order X-Bar Structure

by V, we would have to replace X′ by V′ and X″ by V″ (also written VP). The specifier, complement, and modifier positions define hierarchical relationships, such as dominance and sisterhood, but do not imply a linear ordering for a given level of constituents. The three positions, in turn, may be filled by a phrasal structure, depending the lexical item selected for X.

Determiners were problematic for initial accounts of GB. I take the stance of a growing number of authors that a determiner selects a noun phrase and is not an NP specifier. Abney [1987] argues that this allows determiners to be treated in a non-defective manner. Similarly, Cowper [1987] extends this notion to other phrasal categories to account for additional data. Phrases that were traditionally believed to occupy the specifier position in adjective and adverb phrases select these phrases. These revisions allow structures like those of figure 2.4.

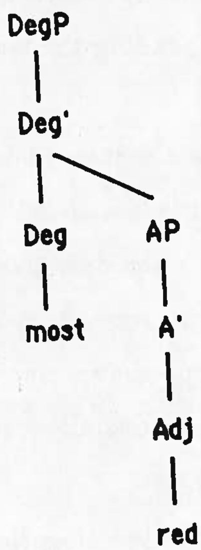
Abney distinguishes between functional and thematic syntactic categories with a category feature, F. Functional elements[5] are +F and thematic elements are −F. One difference between these two classes is that the functional categories have no descriptive content. However, the projections IP, CP, and DP inherit descriptive content from their

[5]This would include syntactic categories "I", "C", and "D" for INFL, complementizer, and deteminer respectively.



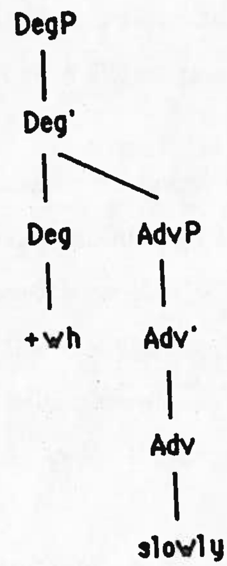
Reading: John's car

Figure 2.4a: A Determiner Phrase



Reading: reddest

Figure 2.4b: An Adjective Degree Phrase



Reading: how slowly

Figure 2.4c: An Adverb Degree Phrase

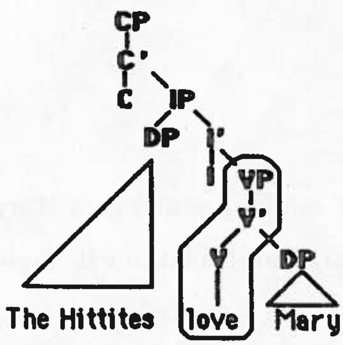


Figure 2.5a: C-projection

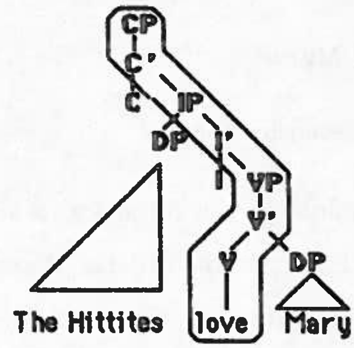


Figure 2.5b: S-projection

arguments. This distinction allowed Abney to define two forms of projection, *c-projection* ("category projection") and *s-projection* ("semantic projection"). A node's c-projection is its syntactic projection in the usual sense: the maximal c-projection of V is a VP, of I is a INFL phrase (IP), and of D is a determiner phrase (DP). However, the s-projection of V is a complementizer phrase (CP). To illustrate this, the c-projection of V is circled in Figure 2.5a and its s-projection is circled in Figure 2.5b. This distinction allows one to capture the intuition that V is the head of a sentence [Jackendoff 1978] without supposing that S=VP or S=V‴.

Cowper [1987] argues that the concept of two heads is not needed if the descriptive content of functional projection is derived from the percolation of features that are not specified by the functional head. However, I shall retain the notion of s-projection and two heads for the following reasons: First, s-projection defines the range of percolation. Second, the notion is consistent with the concept of a head giving phrasal structure its essential character.

2.4.2 θ-theory

θ-theory plays an important role in meaning in that it describes semantic argument structure. Consider the following sentences:

1. John kissed Mary.

2. Mary was kissed by John.

In sentence 1, John is the subject and Mary is the object, while in sentence 2 Mary is the subject and John is the object. Describing only the structural relations in these sentences misses the fact that they both describe the same event. In particular, both sentences describe an event of kissing involving an agent John and a patient Mary.

The roles agent and patient are called thematic relations (theta-relations). GB does not provide a definitive set of thematic relations. Other currently accepted thematic relations include experiencer, goal, location, source, theme, and temporal.

The assignment of theta-relations depends on the argument structure of a given head. Thus, written into the lexical entry for a head is a list of sets of theta-relations. A head can assign multiple theta-relations to an argument. A set of such theta-relations is called a θ-role. For example, the lexical entry for *send* would include the θ-roles {agent,source}, {theme}, and {goal}, which are assigned to *I*, *flowers*, and *Tu-Tran* in sentence 3.

3. I sent flowers to Tu-Tran.

Unfortunately, a simple list of θ-roles does not convey enough information. First, while verbs do not subcategorize[6] for subjects, they may assign (indirectly) a θ-role to the subject position. By convention this argument, known as the *external* argument. Second, one θ-role may be assigned directly to an argument, while others are assigned by a preposition or mediated by the composition of the whole verb phrase. These other θ-roles are merely required (or allowed) by the head. Third, some arguments are optional.

This more complicated structure, known as a θ grid contains the extra information needed to determine the status of each θ-role. For example, ({agent,source}, **{theme}**,

[6]Subcategorization is a filter that restricts the range of maximal projections that can be an argument of a head.

({goal})) represents the θ-grid associated with the verb *send.* Underlining denotes an external argument, boldface a directly assigned role, and parentheses an optional role.

As θ-theory has developed it has become clear that there is not a simple one-to-one correspondence between syntactic positions and semantic relations. First, Jackendoff [1972] demonstrated that a single syntactic position can bear more than one thematic relation. More recently, Brunson [1987] demonstrated that certain θ-roles may be distributed over the syntactic structure. This is the case with oblique θ-roles (θ-roles assigned by prepositions). For example, the prepositional phrases each of the sentences 4, 5, and 6 compositionally bear a θ-role, namely {location}, {temporal}, and {location} respectively.

4. John saw Mary at noon on Tuesday.

5. John saw Mary in the park near the tree.

6. On Tuesday, John saw Mary at noon.

The current implementation of ELOQUENCE cannot generate distributed θ-roles. However, this is not a short-coming of ELOQUENCE's design. The distribution of θ-roles provides an interesting source of structural variation.

2.4.3 An Overview of Government and Binding

GB provides four levels of representation: d-structure, s-structure, phonetic form, and logical form. Well-formedness of these representations is determined by a set of sub-theories and a set of principles. The sub-theories include the following:

- X-bar theory: *see section 2.4.1.*

- Case theory: Case theory determines the distribution of NPs and some other maximal projections. According to the theory, every NP must be assigned a feature called Case. (This feature corresponds well with the traditional concept of case.) For example, in English verbs assign accusative Case and prepositions assign accusative or oblique Case.

- θ-theory: *see section 2.4.2*

- Government theory: Government theory defines the local domain of a head. That is, β is in the domain of α if and only if α governs β. Proper government is a restricted form of government in which the governer in a lexical category.

- Movement theory: In Movement theory many of the transformations of TG have been replaced by a single rule, Move-α, which is *move anything anywhere*. However, other components of GB restrict Move-α to *wh*-movement and NP-movement.

- Bounding theory: Bounding Theory constrains movement by requiring that Move-α does not operate over too great a distance in a single application of the rule. In English, too great a distance means crossing two nodes of either the category NP or the category CP.

- Binding theory: Binding theory characterizes the relations between NPs. It relates anaphors and pronominals to their antecedents. In particular, Binding theory partitions the class of NPs with two binary features, *anaphoric* and *pronominal*. The presence of the former requires an antecedent, while the presence of the latter allows an antecedent. The theory then defines for each class the environment in which an antecedent, if there is one, may occur.

- Control theory: Control theory is concerned with choices of antecedents for PRO.[7] PRO has many unique properties stemming from it having both anaphoric and pronominal features. That is, in some cases PRO requires an antecedent, and in other cases it does not.

The principles included the following:

- The Projection Principle: Representations at each syntactic level are projected from the lexicon, in that they observe the subcategorization properties of the lexical items [Chomsky, 1984, page 29].

[7] PRO is a phonetically null pronoun that occupies the subject position in infinitival clauses.

- The θ-criterion: Each argument bears one and only one θ-role, and each θ-role is assigned to one and only one argument [Chomsky, 1984, page 36].

- The Empty Category Principle: Every trace must be properly governed. This principle is generalized for PRO [Chomsky, 1984, page 274].

Although some principles and sub-theories such as the projection principle and the θ-criterion are pervasive to all levels of syntactic representation, others only affect a subset of representational levels.

D-structure is a level of argument structure where thematic structure is encoded in syntactic structure by X-bar theory and θ theory.

S-structure is mapped to d-structure by Move-α. However, s-structure is further constrained by Case theory, Binding theory, Control theory, and the Empty Category Principle.

Logical form is considered a meaning representation, because it represents, among other things, scoping and quantification. Since these are not represented unambiguously in s-structure, there is a one-to-many mapping from s-structure to logical form. Logical form is also restricted by Binding theory, Control theory, and the Empty Category Principle.

Phonetic form represents phonetic realization. In other words, it encodes how s-structure should be said.

2.5 Conceptual Graphs

Chapters 1 discusses the role of realization in the generation process. Chapter 3 expands the definition of this role by isolating decisions made by realization from those of text planning. However, the disparity in detail between my definition of generation and the control of an actual text planner can result in unreasonable processing expectations. To counter this problem I chose conceptual graphs to represent the realizer's declarative input. That is, I chose conceptual graphs [Sowa 84] not only for their expressive power, but also because the formalism was designed for computation. With the same justification I could have chosen either scripts, semantic nets, or frames (which all have a similar spirit).

In the remainder of this section I provide an overview of conceptual graphs, describe modifications I made to the formalism, and cite some problems with it.

2.5.1 An Overview of Conceptual Graphs

A conceptual graph is a bipartite graph with two kinds of nodes, concept nodes and conceptual relation nodes. For example, the following graph has the same semantic interpretation as *John loves Mary*:

[PERSON:John] ← (EXPER) ← [LOVE] → (PATIENT) → [PERSON:Mary]

Concepts are organized by the hierarchic method of Aristotle. At the top of the lattice is a universal supertype and at the bottom is an absurd type, which is a subtype of all other types. (Since the absurd type is a subtype of everything an individual of this type would have to be cat, and a dog, and a table) STATE, EVENT, and ENTITY are general types and are subsumed only by the universal type. The type EVENT, for example, has a sub-type ACT, which in turn has sub-types WALK, EAT, GIVE, and TALK.

The formalism allows complex types and relations to be defined in terms of simpler ones. Thus, reasoning can be preformed with different *grain sizes* of primitives.

To specify the referents for concepts, the conceptual graphs formalism supports over 23 kinds of quantifiers for countable and measurable concepts including fuzzy and question quantifiers. The scope of a quantifier is determined by the nesting of context, which is delimited by brackets that enclose one or more graphs. For example, graph 1 represents the reading *There is a man who loves every women* of *Some man loves every woman*, and graph 2 represents the reading *For each woman there is a man who loves her.*

1. [MAN: *x] —
   [[*x] ← (EXPER) ← [LOVE] → (PATIENT) → [WOMAN: ∀]]

2. [WOMAN: *x] —
   [[MAN: ∃] ← (EXPER) ← [LOVE] → (PATIENT) → [*x]]

To solve the problem of repeated references Sowa uses co-reference links to tag referents. This is equivalent to Kamp's approach of using reference markers [Kamp 1985]. I shall use superscript lower case letters to mark the endpoints of co-reference links in my examples.

2.5.2 Modifications to Conceptual Graphs

Brown, Pattabhiraman, Boyer, Massam, and Dahl [1986] introduce several features to provide specific representation for certain essential linguistic phenomena including tense, aspect, and relative clause constructions.

Tense and aspect are represented by two relations TENSE and ASPECT, which are both sentential operators. Together they determine the main verb's form. I did not take this approach, because it does not provide a semantic interpretation for relations and these extensions are too rigid. In other words, it is merely a substitution of one syntax form for another.

Instead I replaced Sowa's POINT-IN-TIME and DURATION relations with a single relation TIME that relates states and events to time periods. This allows ELOQUENCE to determine verb forms by reasoning with Allen's temporal operators [see section 2.1] and the type of surface speech act used.

Dahl *et al* represent relative clauses with a REL-MOD relation that relates a concept to a proposition. Relative clauses can be *restrictive* or *non-restrictive.* The former variety identifies a particular object, while the latter provides additional information. For example, *The man who is eating a shark* is restrictive, while *The man, who was tired, left* is non-restrictive. They assume that whether the clause is restrictive or non-restrictive does not have bearing on the surface structure produced. This not the case with ELOQUENCE, which can realize the latter utterance either as *The man, who was tired, left* or *The man was tired and he left.*

To make this distinction, I represent non-restrictive relative clauses as two conceptual graphs with a co-reference link, but use the new REL-MOD relationship for restrictive clauses. For example, the following graphs, with the co-reference link *a*, represent the non-restrictive clause above:

3. [TIRE] —

→ (PATIENT) → [MAN: #][a]

→ (DURATION) → [TIME-PERIOD1]   (TIME-PERIOD1 < the current time)

4. [LEAVE] —

→ (AGENT) → [MAN: #][a]

→ (DURATION) → [TIME-PERIOD2]   (TIME-PERIOD2 < the current time)

Conceptual graphs as defined by Sowa can represent the conjunction of concepts with quantifiers. For example, the sentence *Jack and Jill danced on the beach* is represented by the following graph:

5. [DANCE] —

→ (AGENT) →   [PERSON:{Jack,Jill}]

→ (LOCATION) → [BEACH:#]

→ (DURATION) → [TIME-PERIOD1]   (TIME-PERIOD1 < the current time)

However, it cannot represent sentential conjunction. To represent this phenomenon I use an abbreviated notation. For example, graph 8 is an abbreviation for the two graphs 6 and 7 and can be realized as *John loves Mary and sunny days.*

6. [LOVE] —

→ (AGENT) → [PERSON:John]

→ (PATIENT) → [PERSON:Mary]

7. [LOVE] —

→ (AGENT) → [PERSON:John]

→ (PATIENT) → [Day:{*}]

→ (CHRC) → [SUNNY]

8. [LOVE] —

→ (AGENT) → [PERSON:John]

→ (PATIENT) → {and}

→ [PERSON:Mary]

→ [Day:*]

→ (CHRC) → [SUNNY]

Chapter 3

Variation in Linguistic Form

In this chapter I examine some constraints on linguistic structure. In particular, I discuss style, lexical selection, and syntactic projection, which are phenomena co-ordinated by ELOQUENCE [see chapter 4].

3.1 Style

Concurrent with work on ELOQUENCE, a project to improve the quality of machine translation output has been underway at Toronto [DiMarco and Hirst 1988]. The research focuses on the translation of an author's intent in terms of stylistic goals. Although I do not propose any new stylistic analysis, ELOQUENCE chooses between different realizations on the basis of the intent of the text in terms of stylistic goals.

Style is "the selection and organization of features of language for expressive effects" [Encyclopedia Britannica, 1973, 21, page 332]. DiMarco and Hirst [1988] follow Hendrix's [1986] distinction between two categories of style: *individual* style and *group* style. Individual style refers to those aspects of writing specific to an author, while group style is the specific manner in which a "type" of text is written and how particular linguistic goals (such as compactness and conciseness) are achieved. Group style is not simply the mean of the individual styles of a set of authors. It is the locus of linguistic features that are present in any one individual's style, but lacks the individuating features. While individual style classifies individuals, group style classifies functions.

Individual style has an objective consistency that enables one to "fingerprint" an author by performing a statistical analysis of his work. However, it is group style which is more relevant to realization. Word-frequency studies are useful for literary or forensic analysis of text, but do not bring to light any information about how well a linguistic goal is achieved. Unfortunately, many people's conception of style is based on individual style. Thus arises the misconception that a technical document has no style or a neutral style, when it has a particular group style. Since this thesis is concerned with realization of text, I refer to *group* style generically as style.

Not all aspects of style are found in the domain of realization. Choices in rhetorical structure and semantic content are used more often to achieve a desired effect. Such choices are in the domain of content selection and discourse planning. Thus, I shall make the distinction by referring to rhetorical style as ***strategy*** and narrowing my application of the term ***style*** to realization. Strategy concerns aspects of the topic one decides to develop and how one organizes them. Style is the set of linguistic constructs (the selected words and syntactic patterns) that make strategy work.

The study of diction concerns that part of style that deals with choice of words. For example, obsolete words can be used to create an air of dignity (or possibly a mystic or pedantic air). Besides selection of words, style deals with the arrangement of words (selection of syntactic form). For example, if qualifying words, phrases, and clauses are placed nearer words they modify, the meaning is more readily apparent to the reader and is therefore deemed to be clearer [Kane 1986].

I concentrate on a representative class of basic stylistic goals: clarity, emphasis, formality, and strength. Clarity is a measure of how easily meaning can be perceived from an utterance, concision is a measure of the brevity of an utterance relative to the amount of information conveyed, emphasis is a measure of the stress in an utterance, formality is measure of the conformity of an utterance to the social aspects of a situation and strength is a measure of force in an utterance.

Emphasis functions differently than the other styles I consider. While strength or clarity may be goals for an entire interval of text, emphasis specifies goals for subsections of the interval text. That is, emphasis marks what ought to be ***figure*** and what ought to be *ground*.

3.2 Lexicalization

Recall that KAMP represents each individual of its domain as a set of predicates that are true of the individual [see chapter 1]. When the system wishes to create a referring expression, it must decide which predicates must be communicated by the utterance. Pustejovsky and Nirenburg [1987] and Miezitis [forthcoming] have also addressed the semantic coverage of expressions. Pustejovsky and Nirenburg explored the distinction between generating *open* and *closed* lexical items. They define open-class items to be lexical items with "content", such as verbs, nouns, and adjectives. Closed-class items are defined as "functional" lexical items, such as determiners and prepositions. They conclude that the two classes are not only conceptually different, but should be processed differently. In particular, Pustejovsky and Nirenburg derive prepositions from case roles and anaphoric pronouns from contextual information. However, they treat open-class items in the same spirit as KAMP's referring expressions.

Miezitis has developed a system called LOG for lexical option generation. The system is designed to provide a natural language generator with lexical and idiomatic options, for a given semantic content. The input to LOG is a set of frames that represent the information to be conveyed. The output is a mapping between expressions and the input. Each expression conveys some or all the information of the frame.

Although the input to ELOQUENCE does not include lexical items, it requires that part of the lexicalization process be performed already. In other words the lexicalization process is distributed between the realizer and the planner. In particular, the concepts of the text plan represent the semantic content for an open class item. It is then up to ELOQUENCE to choose a lexical category for the concept and then later a particular item. Closed-class items follow directly from the text plan. This approach would incorporate a system like LOG. However, instead of mapping a set of semantic primitives to a lexical item, the primitives would be mapped to high-level concepts. For example, the modified system would map proposition 1, which represents some individual in the system's model, to a new definition as in proposition 2. Proposition 2 does not encode a particular lexicalization of proposition 1, although in this case the noun "flower" may be the only words that captures the properties of proposition 1. The semantic properties of an emergence event, [EMERGE], can be captured by the noun "emergence", or one of the verbs "emerge", "surface", and "arise". This does not mean that these word are equivalent. The verb "surface" is a valid lexicalization of the concept of altering the exposed area of an object, but the verb "emerge" is not. In summary, there is a many-to-many mapping between concepts and lexical items.

1. IND#3: PLANT(IND#3) ∧ PART(IND#3,x) ∧ STEM(x) ∧ PART(IND#3,y) ∧ PETALS(y)

2. IND#3: FLOWER(IND#3)

It is not clear how to define this mapping. Is it necessary to define the semantics of every lexical item, or does one simply encode a thesaurus? Must a lexical item match the concept exactly, or is minor variation allowed to improve the system's flexibility? I shall not address these questions. Each would require its own thesis.

Distributing the lexicalization process has the following advantages:

- An underlying planner or knowledge base could reason with low-level semantic properties of an individual without low-level properties appearing in the text plan. In other words the representation reflects the type of reasoning being performed.

- There exists an accepted linguistic theory in which the levels of linguistic representation are projected from the lexicon, namely Government and Binding Theory. The semantic representation includes lexical grain-size concepts and is thus compatible with at least Government and Binding Theory. In particular, the thematic information can be directly derived from the text plan.

- The representation captures the semantic structure, but only minimally limits the syntactic form of the text. This is unlike other systems which explicitly encode key lexical items.

The addition of high-level concepts makes the input representation and the induced mapping language-dependent. In English concepts such as "computer" and "calculator" can be referred to by a single lexical item. However, in Vietnamese a computer is referred to by "may dien tu" (an automatic electronic machine) and a calculator is referred to by "may tin" (a counting machine). On the other hand in Vietnamese the relationship between a parent and the in-laws of his or her child is referred to by "xu" as in "ba xu", which refers to the mother of the family, and "ong xu", which refers to the father of the family. A Vietnamese text planner could build an argument structure in which the "xu" of "ba xu" would be a single argument. D-structure can then be generate directly from the text plan using X-bar Theory and θ-Theory.

3.3 Syntactic Variation

MUMBLE [McDonald *et al* 1983, 1985a, 1985b, 1985c, 1987] maps key lexical items to syntactic patterns using realization classes [section 1.2.2]. With the flexibility of the distributed lexicalization process and the clean integration of Government and Binding theory, the range of realizations can be derived from principles. Utterances 1 and 2 differ at the semantic level, even though 1 can be inferred from 2.

1. Tu-Tran is asleep.
2. Murray reads mail if and only if Tu-Tran sleeps; Murray is reading mail.

Also, 3 and 4 convey the same semantic information, each being used under different pragmatic constraints. However, it is less clear whether the difference between 5 and 6 is in the conceptual level or in the choice in realization. I shall now present justification for considering the difference between 5 and 6 to be semantic.

3. Robert is believed by his followers.
4. Robert's followers believe him.
5. Only in the fairly recent past have the Hittites emerged from obscurity.
6. Only fairly recently have the Hittites emerged from obscurity.

Consider the following sentences:

7. Nero's destruction of Rome disgusted the senate.

   [$_{DP}$ [$_{DP}$ Nero's] [$_{NP}$ destruction [$_{PP}$ of Rome]]] disgusted the senate.

8. Nero's destroying Rome disgusted the senate.

   [$_{DP}$ [$_{DP}$ Nero's] [$_{VP}$ destroying [$_{DP}$ Rome]]] disgusted the senate.

9. That Nero destroyed Rome disgusted the senate.

   [$_{CP}$ That [$_{IP}$ [$_{DP}$ Nero] [$_{VP}$ destroyed [$_{DP}$ Rome]]]] disgusted the senate.

10. Rome's destruction by Nero disgusted the senate.

    [$_{DP}$ [$_{DP}$ Rome's] [$_{NP}$ destruction [$_{PP}$ by Nero]]] disgusted the senate.

In each case, the sense of "destroy" is s-projected to the top of the substructure. Also, the thematic relationships to the semantic head "destroy" are maintained. That is, the sense of the agent of the destruction, "Nero", is present in each sentence as is the theme of the destruction, "Rome". Thus, there is a consistency in thematic assignment. This is even true of the passive version 10, wherein "Nero" is assigned the agent role by the preposition "by", because it is in turn θ-identified by *destruction*. These sentences illustrate the first criterion for judging whether two realizations have the same underlying conceptual structure.

> *All realizations of a conceptual structure have the same thematic structure and the corresponding roles are filled by elements of the same sense.*

Now consider the following structures:

11. The Hittites' recent emergence from obscurity

12. Recently, the Hittites emerged from obscurity.

13. The Hittites have recently emerged from obscurity.

The adverb *recently*, unlike the corresponding adjective, has several locations where it can still be modifying the s-projection of *emerge*. In general, there are three basic surface positions in a sentence for adverb modification.[1] A given class of adverbs will

[1]See Jackendoff [1972] for a detailed analysis.

occupy only a subset of these positions. Occasionally, an adverb will have different senses that will belong to different classes. The regularity of these changes suggest that some general principles might be extracted. In utterances 11 the adjective "recent" modifies the emergence event. In 13 the adverb "recently" modifies the emergence event. In utterance 12 "recent" modifies the emergence event because it modifies the s-projection of the noun "emergence". Since "recent" and "recently" represent the same concept, the sentences are semantically equivalent.

The following is the second criterion for judging whether two realizations have the same underlying conceptual structure:

> *For each element that is not θ-assigned, other than a bound functional element, that occurs in at least one realizations of a conceptual structure, there must be a corresponding element of the same sense in each of the other realization of the conceptual structure. In addition, corresponding elements must modify s-projections of the same sense.*

Bound functional elements include the +POSS in 7, the *-ing* in 8, and the +TENSE in 10. Free-word functional elements would include lexical determiners, such as "the", and lexical INFL's, *i.e.* the modals. These elements are highly significant in realization, because they dramatically reduce the number of possible syntactic structures that are realizations of the conceptual structure. For example, there is no simple sentence equivalent to *The destruction of Rome by Nero.* Although one might feel the uniqueness of the event can be inferred by context in *Nero destroyed Rome*, this is insufficient to justify alternative realizations of the same conceptual structure.

Next, consider the following three sentences:

14. Recently the Hittites emerged from obscurity.

15. It was recently that the Hittites emerged from obscurity.

16. It was in the recent past that the Hittites emerged from obscurity.

Since the "it" in 15 is non-referential, it is not assigned a θ-role. Therefore, by the criteria I have outlined, I would conclude that 14 and 15 are realizations of the same conceptual structure. They have the same thematic structure with a sense of "recent" modifying the s-projection of *emergence*. In sentence 16, on the other hand, "recent" modifies "past" not "emerge". Thus, according to the criteria, 16 fails to match in two ways. Thus I differentiate between the modifier, "recent", and the deictic reference, "recent past".

Finally, consider the following utterances, which we intuitively see as alternative realizations of the same conceptual structure.

17. I saw a red car.

18. I saw a car. It was red.

19. I saw a car [$_{CP}$ that O_i [$_{IP}$ t_i was red]].

A chunk of semantic structure need not always be realized as a single sentence. To do so would limit flexibility in realization. To account for the variation in sentences 17, 18, and 19, we need only modify the criteria slightly. First, we must consider modification of pronouns and the empty operator to be the same as modification to any co-referenced θ-roles. Second, the role of non-lexical functional elements must be taken into account. For example, the tense that dominates "car" must also dominate "red". Although ELOQUENCE currently uses heuristics to check the scope of functional elements, extension of Asher's [1986,1987] work on discourse structure should result in a formal account of this phenomenon.

Chapter 4

ELOQUENCE

In this chapter I describe the implementation of ELOQUENCE, the system in this thesis. I then run through two sample realizations to illustrate its operation. Both sample realizations supposedly represent typical excerpts from general reference material. CYC [Lenat 1987], a knowledge system for encyclopedias, encodes information of this type and could benefit from a system like ELOQUENCE. The current implementation consists of over 4000 lines of Lisp code and runs on a SUN 3/60 work station. The system is still in the production stage and includes a front end that allows a user to simulate a text planner.

The stylistic goals for reference material should include formality. However, not all of ELOQUENCE's evaluation functions have been implemented and the system cannot yet deal with formality.

4.1 The Implementation

ELOQUENCE maintains multiple possible realizations for a text. Each of the possible realizations is treated as a separate process. ELOQUENCE interacts with the text planner through shared data structures that include:

- A queue of conceptual graphs that represent the semantic content for the text
- A pointer into the text plan that marks the point to which the plan has stabilized

- A list of rhetorical goals

- A goal satisfaction vector for each process that represents how well the process satisfies the goals

Each process maintains its own data structures including:

- A partial syntactic realization

- A mapping between the leaf nodes of the syntactic structure and the concept nodes of the underlying conceptual graphs

- A stack that monitors the process's traversal of the graphs.

As the text plan stabilizes, ELOQUENCE starts printing out the realization that is most appropriate for the text's rhetorical goals. Those processes that are inconsistent with the text produced are killed. I shall now discuss the role of each of ELOQUENCE's major components. Figure 4.1 depicts ELOQUENCE's architecture. Arrows represent the flow of information, rectangular boxes represent procedures, and rectangular boxes with the top-right corner removed represent structures.

4.1.1 The Processes

When control is passed to a process, it remains active long enough to realize one concept node. The interval over which control belongs to the process is called "a quantum of realization". Upon relinquishing control, the process forks into as many processes as there are syntactic variations created by the node's realization. Each leaf node of the syntactic structure represents a set of lexical items. ELOQUENCE selects a lexicalization as it reads out the structure. This enables a single structure to represent several possible realizations.

Each frame of the traversal stack, the stack that monitors the process's traversal of the conceptual graphs, consists of:

1. A pointer to a concept node of a graph

2. A pointer to the concept node that immediately dominates the concept node referenced by 1

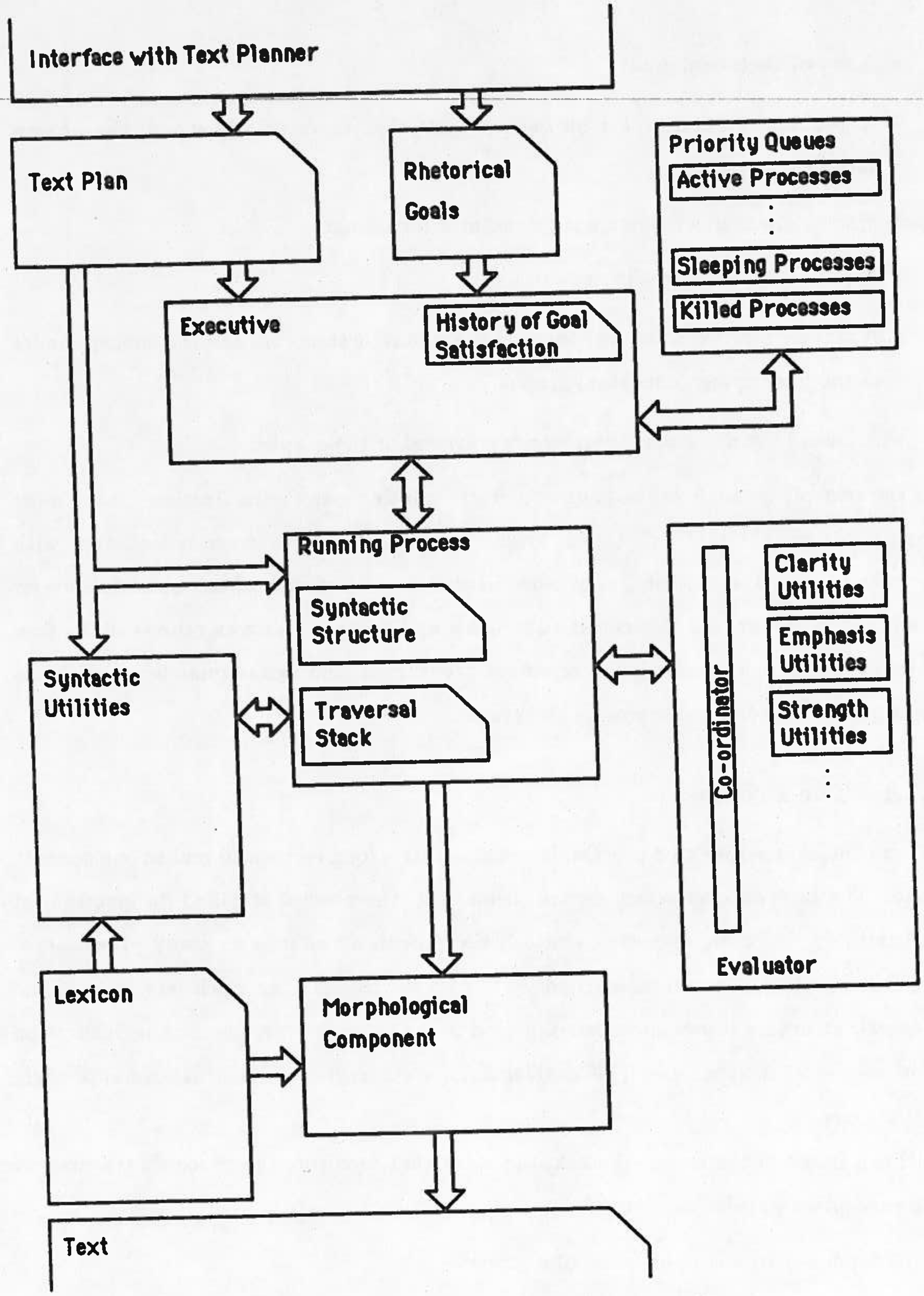


Figure 4.1: ELOQUENCE's Architecture

3. The relationship between the node referenced by 1 and 2

When a stack frame is popped, the first node of the frame becomes the ***active node*** (the node to be realized). A stack frame is then created for each of the children of the active node. If any concept dominates the active node it will have already been realized. Its realization and the relationship between it and the active node limit the manner in which the active node can be realized.

A concept node can be realized in two manners. The first method is to create a fragment of syntax for the concept. The fragment is a syntactic projection of the lexical category chosen for the concept, which is then assimilated (in a manner consistent with the underlying semantics) into the structure being created. Consider the example where the active node is [COUNTRY: Rome] in graph 1. ELOQUENCE checks its lexicon and finds only one lexical category to realize the active node (a determiner phrase). Proper names have a lexical category of DP, which is a maximal projection. If [INVADE], which dominates the active node, were realized as a complementizer phrase then sentence 2 would be one possible assimilation of the active node into the syntactic structure. If it were realized as a determiner phrase then sentence 3 would one possible assimilation of the active node into the syntactic structure.

1. [INVADE] —
   →(AGENT)→ [COUNTRY: Rome]
   →(PATIENT)→ [COUNTRY: England]
   →(DURATION)→[TIME-PERIOD1]

2. Rome invaded England.
3. Rome's invasion of England

The second method, *assimilation*, can be used with a concept node that co-references a previously realized concept. The active node can be realized by assimilating two fragments of the structure that has already been created. Consider the example where graph 4 has been realized as sentence 6, the active node is [PERSON: John] of graph 5, and the rest of graph 5 has been realized as 7. ELOQUENCE can combine 6 and 7 into sentence 8.

4. [EMPLOY] —
    →(AGENT)→ [FATHER]—
        →(POSS)→[PERSON:Mary][a]
    →(PATIENT)→[PERSON:John][b]

5. [LOVE] —
    →(EXPERIENCER)→[PERSON:John][b]
    →(PATIENT)→ [PERSON:Mary][a]

6. John is employed by Mary's father.
7. XP love Mary.
8. John, who loves Mary, is employed by Mary's father.

4.1.2 The Executive

The idea behind creating separate processes is to separate the rules of realization from the procedures that control processing. The component called "the executive" handles administrative aspects of the realization. It co-ordinates the realization processes by performing the following functions:

1. It decides when a process can be removed from the priority queues and given control.

2. If a process requires more data but none is available, the executive will put it to sleep until the planner can provide more data.

3. After a process has performed one quantum of realization, the executive invokes a sub-system called "the evaluator", which updates the goal satisfaction vector for

the process and creates a vector for each newly created process.

4. If the number of active processes exceeds a preset limit, the executive kills off processes that are unlikely to produce the best realization, on the basis of information provided by the evaluator.

Of these functions, the last is the most complex. Suppose there is a limit of twenty processes. An inefficient approach would be to simply keep the twenty realizations that best satisfy the rhetorical goals. Structures that differ only slightly will often have roughly the same effect. If only the "best" structures are chosen, then there will not be the variety needed to avoid local maximums in the search space. Although certain structural decisions may dramatically improve the text, they can limit the variety of choices available in the realization of the rest of the text. For example, there are fewer locations where modifiers can be attached in a nominalization than in a complementizer phrase. The system needs both variety and quality. However, it is impossible to anticipate which variations are the most promising. In addition, it is difficult to compare texts and quantify their differences. Instead, the executive keeps track of the history of the realizations. ELOQUENCE trades off the quality of the texts against the trend of the satisfaction vector.

4.1.3 The Evaluator

The evaluator determines the pragmatic effects of a realization's syntactic structure. For each stylistic aspect [section 3.1], the evaluator assigns a weight to the structure. The weights are normalized to a value between 0 and 1, where for each aspect 1 is assigned to the realization that is most appropriate for that aspect. The executive checks a structure's satisfaction of a stylistic goal by examining the weight of the aspect corresponding to the goal. To obtain a cumulative evaluation of the structure, ELOQUENCE computes the mean of the weights associated with the stylistic goals. This scheme can modified to check the deviation of the weights.

To create each of these weights the evaluator must co-ordinate several evaluation functions. Each function examines a structure for particular properties. For example, to determine the clarity of the structure there are functions that check the complexity of sentences, it looks for balanced or parallel syntactic patterns, and measures the distances of modifiers from their concepts. The relative contribution of these functions to ELOQUENCE's weights is based on intuition. It is not possible to collect, for this thesis, the statistics to create accurate weights.

4.1.4 Reading Out The Linguistic Structure

ELOQUENCE requires that the text planner maintain a pointer that marks the point to which the plan has stabilized. ELOQUENCE commits itself to realization decisions for the text prior to this point. First, it determines which realization best satisfies the rhetorical goals. Then, it reads out as much of the realization as it can without realizing any part of the plan that has not stabilized. At this point, the executive can kill all the processes that are not consistent with the text already generated. To determine which processes to kill, ELOQUENCE must determine which structure was the oldest ancestor of the chosen realization that subsumes the read out part of the chosen realization's structure. This information is derived from the historical record of the realizations kept by the executive. Since a realization is consistent with the text produced if and only if it is a descendent of this ancestor of the chosen realization, any processes which are not descendents of this ancestor are killed.

Recall that the leaf nodes are sets of lexical items [section 4.1.1] The lexical entry for a lexical item includes weights for each stylistic aspect. Before reading out the structure, ELOQUENCE maximizes the lexicalization of the structure for the stylistic goals. It then traverses, in order, the linguistic structure. It halts when it comes to a node that does not map to a stable section of the plan. As it descends through the structure, ELOQUENCE keeps track of the scope of functional information. In other words, the determiners, INFLs, and DEGs are stacked. Lexical items are passed with any relevant functional information to a morphological component. In turn, the morphological component inflects the lexemes and prints them out.

The original morphological component was implemented in Common Lisp by Stephen Harris. Since the rest of the system was written in Franz Lisp, the reading out function has not yet been added to the system. Instead the current output is in the form of parse trees.

4.2 Sample Realization 1

For the purpose of illustration, the examples are realized in a post-process manner. I minimize the number of structures examined by concentrating on the structural decisions that are eventually taken. In this section, I examine the realization of the text plan of Figure 4.2. There are three co-referent occurrences of [HITTITE:{*}], as well as two of [OBSCURITY:#] and two of [REFERENCE:{*}]. In addition, [RECENT] and [POINT-IN-TIME: 19th Century] are marked for emphasis by a superscript "E". The plan can be realized as the following text:

> Fairly recently, the Hittites emerged from the obscurity which enveloped them for almost 3000 years. Until the Nineteenth Century, knowledge of the Hittites was confined to the references in The Old Testament, which numbered them among the minor tribes of the hill country of Palestine.

Currently ELOQUENCE does not have a component to pronominalize co-references. Thus, the two occurrences of "them" would appear as "the Hittites" in ELOQUENCE's realization of the text. The stylistic goals for the text are clarity and emphasis, and all temporal intervals are prior to the temporal aspect of the deitic centre.

The executive starts the first process with no syntactic structure and [EMERGE], [ENVELOP], [CONFINE], and [RECKON] in the traversal stack. When control is passed to this process, [EMERGE] is popped off the stack and [HITTITE:{*}], [OBSCURITY], and [RECENT] are pushed onto the stack. [TIME-PERIOD1] provides functional information, which is realized during the same quantum as [EMERGE]. That is why [RECENT] is pushed onto the stack and not [TIME-PERIOD1].

The lexicon informs the syntactic utilities that [EMERGE] can be realized as a noun or a verb. This results in structures 1 to 5 being created. (These structures can be found starting on page 93). In structure 1 and 2 the temporal information is used to determine the tense. In structure 2 the tense restricts the structure to a non-extended

```
[EMERGE] —
    → (TIME)               → [TIME-PERIOD1] —
                                → (ATTR) →      [RECENT]^E —
                                                    → (CHRC) → [MODERATE]
    → (AGENT-THEME) → [HITTITE:{*}]^a
    → (SOURCE) →           [OBSCURITY:#]^b → (REL-MOD) —

                → [PROPOSITION: [ENVELOP] —
                      → (TIME) →       [TIME-PERIOD2] —
                                            → (LENGTH) →  [YEAR: @almost 3000]]
                      → (AGENT) →      [OBSCURITY:#]^b
                      → (PATIENT) → [HITTITE:{*}]^a

[CONFINE] —
    → (TIME) →             [TIME-PERIOD3] —
                                → (UNTIL) —
                                      → [POINT-IN-TIME1: 19th Century]^E
    → (AGENT) →            [*Null*]
    → (PATIENT) →          [KNOWLEDGE:*] —
                                → (PERT-TO) → [HITTITE:{*}]^a
    → (LOCATION) →         [REFERENCE:{*}]^c —
                                → (CONT) →      [DOCUMENT: The Old Testament]

[RECKON] —
    → (TIME) →             [TIME-PERIOD4]
    → (AGENT) →            [REFERENCE:{*}]^c
    → (PATIENT) →          [HITTITE:{*}]^a
    → (LOCATION) →         [TRIBE: {*}]
                                → (ATTR) →      [MINOR]
                                → (POSS) →      [HILL-COUNTRY:#]
                                                    → (PART-OF) —
                                                          → [COUNTRY: Palestine]
```

Figure 4.2: Text Plan 1

form. In particular, the structure of 2 is that of "emerging" and not that of "having emerged". The DET slots of structures 3, 4, and 5 are currently empty, since [EMERGE] has no quantifier. Unless a DET slot is later filled, as it is by the possessive element in "The Hittites' emergence", the structure will be killed. At the end of the quantum of realization there exists a process for each structure. I shall continue with the first structure. The others lead to convoluted sentences, which conflict with the goal of clarity, such as "Recent is the Hittites' emergence from the obscurity which enveloped them for about 3000 years."

ELOQUENCE checks the lexicon and finds only the category noun for [OBSCURITY:#]. Thus it can only be realized as a determiner phrase. Next, there is a REL-MOD relationship between [OBSCURITY] and the graph enclosed by [PROPOSITION:...]. As in section 3.3, the reader is referred to Asher [1986,1987] for a general treatment of propositional attitude. The REL-MOD, which takes an entire proposition as an argument, is one of the extensions of Dahl *et al* [see section 2.5.2]. The enclosing of a graph in a [PROPOSITION...] context is part of the standard formalism defined by Sowa [1984].

ELOQUENCE starts realizing the enclosed graph as it own structure. Three quanta of realization later, the [ENVELOP] and [HITTITE:{*}] nodes will have already been realized and incorporated into the structure, as in the syntactic tree of structure 7. ELOQUENCE must then decide how to realize the [OBSCURITY:#] node of the [ENVELOP] graph. [OBSCURITY:#] could normally be base-generated either externally or internally to the verb phrase. However, since the [OBSCURITY:#] node in the [ENVELOP] graph is a co-referent of the [OBSCURITY:#] node outside the [PROPOSITION:...], structure 6 can be combined with structure 7 as in structure 8. When the realization of [YEAR: @almost 3000] is attached, the location closest to [ENVELOP] is preferred, and structure 9 results.

After realizing everything on the [OBSCURITY:#] branch of the [EMERGE] graph, ELOQUENCE starts to traverse the [HITTITE:{*}] branch. The [HITTITE:{*}] represents a proper reference and thus can only be realized as a determiner phrase. At this point ELOQUENCE can only base-generated [HITTITE:{*}] externally to the verb phrase, corresponding to the active structure 10, since [EMERGE] cannot be passivized.

[RECENT] can be realized as either an adjectival or an adverbial phrase. The former can be incorporated into a determiner phrase. Since in the structure under investigation [EMERGE] has been realized as a complementizer phrase, the adverbial phrase is chosen. However, the phrase can be attached to more than one location. Structures 12 and 13 result in clearer attachments, while structure 11 results in a more emphatic attachment. As the process continues other decisions will affect clarity. Since [RECENT] is one of only two concepts to be emphasized, its attachment is more critical to the goal of emphasis than that of clarity. With the realization of [MODERATE] 11 becomes structure 14.

To summarize the next few steps, [CONFINE] and [REFERENCE:{*}] are realized as in structure 15. Since [DOCUMENT:The Old Testament], like [HITTITES], is a proper reference, it can only be realized as a determiner phrase. After it is realized, a preposition is chosen for the relationship between it and [REFERENCE:{*}]. Then the resulting prepositional phrase is attached in the complement position of the realization of [REFERENCE:{*}]. The argument of [KNOWLEDGE:*], [HITTITE:{*}], is realized in a similar manner and structure 16 results.

Since the (AGENT) position of the [CONFINE] graph is not overt, the sub-structure associated with [KNOWLEDGE] will move to the subject position. [POINT-IN-TIME:19th Century] is realized as the prepositional phrase "until the Nineteenth Century" and as with [RECENT] the sentence-initial position is chosen. The realization now looks like 17 and the traversal stack contains only [RECKON].

As this realization proceeds, [RECKON] is realized as a complementizer phrase with [HITTITE: {*}] and [TRIBE:{*}] realized internally to the verb phrase. Next, [MINOR] is realized as an adjectival degree phrase modifying [TRIBE:{*}]. There exists a belongs-to relation between [TRIBE:{*}] and [HILL-COUNTRY:#]. This would enable the realization of [HILL-COUNTRY:#] to occupy a specifier position of a determiner phrase with the possessive element in the DET or to occupy a complement position of a determiner phrase as part of an "of" prepositional phrase. However, the quantifier from [TRIBE:{*}] already occupies the DET position. Thus, only structure 18 is allowed.

ELOQUENCE realizes [COUNTRY:Palestine] as a complement of [HILL-COUNTRY:#]. It then takes advantage of the two occurrences of [REFERENCE:{*}]'s co-reference and combines the [CONFINE] graph with that of the [RECKON] graph. Thus, this process's final structure 19 is produced.

Structure 1:

```
CP   ->CBAR ->IP   ->IBAR ->INFL -> simple past

                   ->VP   ->VBAR ->VERB -> [emerge
                                            surface
                                            arise  ]
```

Reading: emerged or surfaced or arose

Structure 2:

```
DP   -> -ing

     ->IP   ->IBAR ->INFL -> simple

                   ->VP   ->VBAR ->VERB -> [emerge
                                            surface
                                            arise  ]
```

Reading: emerging or surfacing or arising

Structure 3

```
DP   ->DBAR ->DET  -> —

            ->NP   ->NBAR ->NOUN -> [emergence]
```

Reading: emergence

Structure 4

```
DP   ->DBAR ->DET  -> —
           ->NP   -> -ing
                  ->VP   ->VBAR  ->VERB -> [emerge
                                             surface
                                             arise]
```

Reading: emerging

Structure 5

```
DP   ->DBAR ->DET  -> —
           ->NP   ->NBAR ->NOUN -> -ing
                               ->VERB -> [emerge
                                          surface
                                          arise]
```

Reading: emerging

Structure 6:

```
CP  ->CBAR ->COMP -> —

        ->IP   ->IBAR ->INFL -> simple past

                   ->VP   ->VBAR ->VERB -> [ emerge  ]
                                           [ surface ]
                                           [ arise   ]

                             ->PP   ->PBAR ->PREP -> from

                                       ->DP   ->DBAR ->DET  -> definite

                                                 ->NP   ->NBAR ->NOUN  -> [ obscurity    ]
                                                                          [ haze         ]
                                                                          [ formlessness ]
```

Reading: The Hittites emerged from the obscurity.

Structure 7:

```
CP  ->CBAR ->COMP -> —

        ->IP   ->IBAR ->INFL -> simple past

                   ->VP   ->VBAR ->VERB -> [ envelop
                                             shroud
                                             cloak ]

                             ->DP   -> the [Hittite]s
```

Reading: enveloped the Hittites

Structure 8:

```
CP  ->CBAR ->COMP -> —

         ->IP   ->IBAR ->INFL -> simple past

                      ->VP   ->VBAR ->VERB -> [emerge  ]
                                              [surface ]
                                              [arise   ]

                                   ->PP   ->PBAR ->PREP -> from

                                                ->DP   ->DBAR ->DET  -> definite

                                                             ->NP   ->NBAR ->NOUN -> [obscurity   ]
                                                                                     [haze        ]
                                                                                     [formlessness]

                                                                       -> See Sub-structure 8a
```

Sub-structure 8a:

```
   ->CP   ->CBAR ->COMP -> which_i

             ->IP   ->DP   -> t_i

                  ->IBAR ->INFL -> simple past

                       ->VP   -> [envelop] the [Hittite]s
                                 [shroud ]
                                 [cloak  ]
```

Reading: Emerged from the obscurity which enveloped the Hittites

Structure 9:

```
CP  ->CBAR ->COMP -> —

        ->IP   ->IBAR ->INFL -> simple past

                    ->VP   ->VBAR ->VERB -> [ emerge
                                              surface
                                              arise ]

                              ->PP   ->PBAR ->PREP -> from

                                          ->DP   ->DBAR ->DET  -> definite

                                                     ->NP   ->NBAR ->NOUN -> [ obscurity
                                                                               haze
                                                                               formlessness ]
```

-> See Sub-structure 9a on the next page

Sub-structure 9a:

```
->CP   ->CBAR ->COMP -> which_i

             ->IP   ->DP   -> t_i

                    ->IBAR ->INFL -> simple past

                           ->VP   ->VBAR ->VERB -> [ envelop ]
                                                   [ shroud  ]
                                                   [ cloak   ]

                                        ->DP   -> the [Hittite]s

                                  ->PP   ->PBAR ->PREP -> for

                                               ->DP   ->QP   ->QBAR ->Quan -> almost

                                                                    ->DP   -> 3000

                                                      ->DBAR ->DET  -> plural

                                                             ->NP   ->NBAR ->NOUN -> [year]
```

Reading: emerged from the obscurity which enveloped the Hittites for almost 3000 years

Structure 10:

```
CP  ->CBAR ->COMP -> —

        ->IP   ->DP   -> The [Hittite]s

                   ->IBAR ->INFL -> simple past

                        ->VP   ->VBAR ->VERB -> [emerge ]
                                                [surface]
                                                [arise  ]

                                  ->PP   -> from the [obscurity    ] which [envelop]d the [Hittite]s
                                                     [haze         ]       [shroud ]
                                                     [formlessness ]       [cloak  ]

                                            for almost 3000 [year]s

                             ->DEGP ->DEGBA->DEG  -> —

                                       ->ADVP ->ADVBA->ADV  -> [recently]
```

Reading: The Hittites emerged recently from the obscurity which enveloped the Hittites for almost 3000 years.

Structure 11:

```
CP  ->CBAR ->COMP -> —

         ->IP   ->DP   -> The [Hittite]s

                ->IBAR ->INFL -> simple past

                       ->VP   ->VBAR ->VERB -> [emerge ]
                                                [surface]
                                                [arise  ]

                              ->PP   -> from the [obscurity   ] which [envelop]d the [Hittite]s
                                                 [haze        ]       [shroud ]
                                                 [formlessness]       [cloak  ]

                                        for almost 3000 [year]s

   ->DEGP ->DEGBA->DEG  -> —

                ->ADVP ->ADVBA->ADV  -> [recently]
```

Reading: Recently the Hittites emerged from the obscurity which enveloped the Hittites for almost 3000 years.

Structure 12:

```
CP  ->CBAR ->COMP -> —

        ->IP   ->DP   -> The [Hittite]s

               ->IBAR ->INFL -> simple past

                      ->VP   ->VBAR ->VERB -> [emerge ]
                                               [surface]
                                               [arise  ]

                             ->PP   -> from the [obscurity    ] which [envelop]d the [Hittite]s
                                                [haze         ]       [shroud ]
                                                [formlessness ]       [cloak  ]

                                       for almost 3000 [year]s

              ->DEGP ->DEGBA->DEG  -> —

                     ->ADVP ->ADVBA->ADV  -> [recently]
```

Reading: The Hittites recently emerged from obscurity which the Hittites for almost 3000 years.

Structure 13:

```
CP  ->CBAR ->COMP -> —

        ->IP   ->DP   -> The [Hittite] s

                   ->IBAR ->INFL -> simple past

                         ->VP   ->VBAR ->VERB -> [emerge
                                                  surface
                                                  arise  ]

                                ->PP   -> from the [obscurity   ] which [envelop] the [Hittite ]s
                                                    haze                 shroud
                                                    formlessness         cloak

                                  for almost 3000 [year]s

                          ->DEGP ->DEGBA->DEG  -> —

                                      ->ADVP ->ADVBA->ADV  -> [recently]
```

Reading: The Hittites emerged recently from the obscurity which enveloped the Hittites for almost 3000 years.

Structure 14:

```
CP  ->CBAR ->COMP -> —

         ->IP   ->DP   -> The [Hittite]s

                ->IBAR ->INFL -> simple past

                       ->VP   ->VBAR ->VERB -> [ emerge  ]
                                                [ surface ]
                                                [ cloak   ]

                              ->PP   ->PBAR ->PREP -> from

                                     ->DBAR ->DET  -> definite

                                            ->NP   ->NBAR ->NOUN -> [ obscurity    ]
                                                                    [ haze         ]
                                                                    [ formlessness ]
```

-> See Sub-structure 14a on the next page

Sub-structure 14a:

```
->CP   ->CBAR ->COMP -> which_i

              ->IP   ->DP   -> t_i

                     ->IBAR ->INFL -> simple past

                            ->VP   -> [envelop] the [Hittite]s for almost 3000 [year]s
                                      [shroud ]
                                      [cloak  ]
```

Reading: Fairly recently the Hittites emerged from the obscurity which enveloped the Hittites for almost 3000 years.

Structure 15:

```
CP  ->CBAR ->COMP -> —

        ->IP   ->IBAR ->INFL -> simple past

                   ->VP   ->VBAR ->VERB -> [confine
                                            limit]

                             ->PP   ->PBAR ->PREP -> to

                                      ->DP   ->DBAR ->DET  ->  define plural

                                                ->NP   ->NBAR ->NOUN -> [reference]
```

Reading: confined ____ to references

Structure 16:

```
CP  ->CBAR ->COMP -> —

         ->IP   ->IBAR ->INFL -> simple past

               ->VP   ->VBAR ->VERB -> [confine]
                                       [limit  ]

                           ->DP   ->DBAR ->DET  ->[generic]

                                       ->NP   ->NBAR ->NOUN -> [knowledge]

                                                   ->PP   ->PBAR ->PREP -> of

                                                               ->DP   ->  the[Hittite]s

                           ->PP   ->PBAR ->PREP -> to

                                       ->DP   ->DBAR ->DET  -> definite plural

                                                   ->NP   ->PBAR ->NOUN -> [reference]

                                                               ->PP   ->PBAR ->PREP -> in

                                                                           ->DP   -> [The Old Testament]
```

Reading: confined knowledge of the Hittites to references in the Old Testament

Structure 17:

```
CP  ->COMP -> —

    ->CBAR ->IP   -> [knowlege] of the [Hittite]s_i

           ->IBAR ->INFL -> simple past passive

                  ->VP   ->VBAR ->VERB -> [ confine ]
                                          [ limit   ]

                                ->DP   -> t_i

                                ->PP   -> to [reference]s in [The Old Testament]

    ->PP   ->DBAR ->PREP -> until

                  ->DP   -> [the Nineteenth Century]
```

Reading: Until the Nineteenth Century knowledge of the Hittites was confined to references in The Old Testatment.

Structure 18:

```
CP  ->COMP -> —

   ->CBAR ->IP   ->IBAR ->INFL -> simple past

         ->VP   ->VBAR ->VERB -> [ number ]
                                 [ reckon ]

                      ->DP   -> the [Hittite]s

                      ->PP   ->PBAR ->PREP -> among
```

-> See Sub-structure 18a on the next page

Sub-structure 18a:

```
->DP   ->DBAR ->DET  -> definite plural

              ->NP   ->NBAR ->NOUN -> [tribe]

                     ->PP   ->PBAR ->PREP -> of

                            ->DP   ->DBAR ->DET  -> definite

                                   ->NP   ->NBAR ->NOUN ->[ hill country]

          ->DEGP ->DEGBA ->DEG  -> —

                 ->AP   ->ABAR  ->ADJ  -> [minor
                                           unnoteworthy
                                           unimportant ]
```

Reading: numbered the Hittites among the minor tribes of the hill country

Structure 19:

```
CP  ->CBAR ->COMP -> —

        ->IP   -> the [Hittite]s [emerge ]d  from the [obscurity   ] which [envelop]d the [Hittite]s for almost 3000 [year]s.
                                 [surface]d           [haze        ]       [shroud ]d
                                 [arise  ]d           [formlessness]       [cloak  ]d

    ->DEGP -> [fairly    ][recently]
              [moderately]
              [relatively]

CP  ->COMP -> —

    ->CBAR ->IP   ->DP   -> [knowledge]of the  [Hittite]s_i

                  ->IBAR ->INFL-> simple past

                         ->VP   ->VBAR ->VERB -> [confine]
                                                 [limit  ]

                                      ->DP   -> t_i

                                      ->PP   ->PBAR ->PREP -> to

                                                   ->DP   ->DBAR ->DET  -> definite plural

                                                                 ->NBAR ->NOUN -> [reference]

                                                                        ->PP   -> in [The Old Testament]

                                                                        -> See Sub-structure 19a on next page

    ->PP   -> until  the Nineteenth Century
```

Sub-structure 19a:

```
->CP   ->CBAR ->COMP -> —
           ->IP   ->DP   -> which_i
                  ->IBAR ->INFL -> simple past
                         ->VP   ->VBAR ->VERB -> [number
                                                  reckon]
                                       ->DP   -> the [Hittite] s
                                       ->PP   ->PBAR ->PREP -> among
                                                     ->DP   ->DBAR ->DET  -> definite plural
                                                                   ->NP   ->NBAR ->NOUN -> [tribe]
                                                                                 -> See Sub-structure 19b on the next page
                                                                   -> [minor
                                                                       unnoteworthy
                                                                       unimportant]
```

4.3 Sample Realization 2

In this section, I examine the realization of the plan represented by figure 4.3, which can be realized as the following text:

> Although Bronze is an alloy of copper and tin, it is commonly thought that the early copper period is part of the Bronze Age. With weapons of bronze, violence and war entered history. The copper period was relatively peaceful. The Bronze Age ended in about 1000 BC, because man discovered a more deadly metal, iron.

The plan contains two sentential conjunctions (CONTRAST) and (CAUSE3). It also uses a predicate of propositional attitude [BELIEVE] and a predicate expressing a type of equality [BE-1]. I shall concentrate on the realization of the new features. Only the goal of clarity will used in the selection of the realization presented.

(CONTRAST) and (CAUSE3) are treated like more general sentential conjunctions. As it does in the case of {AND} of figure 4.4, ELOQUENCE departs from the rule that one concept results in the realization of one concept. In realizing figure 4.4 [DOG:{*}] and [CAT:{*}] are processed together. In realizing figure 4.3, the first [BE-1] and [BELIEVE] are realized together, and [FINISH] and [DISCOVER] are realized at the same time.

However, {and} is less restrictive than either (CONTRAST) or (CAUSE3). It does not take a fixed number of arguments. Nor does it restrict the syntactic category of its realization. Figure 4.4 can be realized as any of structures 20, 21, 22, 23, or 24. (These structures can be found starting on page 91.) Both (CONTRAST) and (CAUSE3) take exactly two arguments. When (CONTRAST) is popped from the stack, [BE-1] and [BELIEVE] can be realized as either two complementizer phrases as in structure 25, or as as a determiner phrase and a complementizer phrase as in structure 26.

Similarly, when (CAUSE3) is popped [FINISH] and [DISCOVER] are realized as two complementizer phrases as in structure 27. Sowa [1984] defines only one predicate of causality. I use a range of predicates, which is the scale approach to causality from Lowe [1987]. Although the selection of degree of causation is pragmatic, its effects are prescriptive and are best handled by the planner. That is, causality communicates the relationship between events in the situation.

The predicate [BE-1] has been used other implementations. In particular, Hirst [1987] and Dahl *et al* [1986] use such a predicate to represent the equality definition of the verb *to be*. The realization of the [BE-1] sub-graph of the (CONTRAST) graph ends up as structure 28. The other [BE-1] graph can be realized as structure 29. However, it is clearer (and stronger) to realize it simply as "iron", in "a more deadly metal, iron".

The [BELIEVE] predicate requires a proposition for one of its arguments. The particular argument of [BELIEVE] and the [TIME-PERIOD:#copper] graph are propositions that have commonly been represented using [BE-2] as a type of identity. Dahl *et al* argue that such propositions should be represented without using [BE-2]. The approach taken here, which Dahl *et al* use for a sub-set of these propositions, is more consistent with the linguistic data. In general, [BE-2] type information can be realized as either a small clause or a complementizer phrase. The proposition sub-graph of the (CONTRAST) graph can be realized as one of the following:

1. A complementizer phrase, as in structure 30, in which movement occurs from the subject position of the tenseless clause to the subject position of [BELIEVE]'s realization

2. A tensed clause, as in structure 31

3. A small clause, as in structure 32, in which movement occurs from the specifier position of the null INFL phrase to the subject position of [BELIEVE]'s realization

However, it cannot be realized as a small clause where movement does not occur, since nothing in the small clause can assign case to [TIME-PERIOD1:#copper]. If an overt angent were realized in the subject position of structure 32 then *the copper period* could not move and the resulting sentence would be ill-formed.

ELOQUENCE cannot take advantage of the co-reference of the two occurrences of [TIME-PERIOD1:#copper], because they occur in different attitude contexts. Since a small clause cannot exist independently as a fragment, the [TIME-PERIOD1:#copper] graph is realized as a complementizer phrase, as in structure 33.

```
(CONTRAST) —
    → [BE-1] —
            → (ENTITY) → [BRONZE:*]
            → (ENTITY) → [ALLOY:*] → (MADE-OF) → [METAL:{copper,tin}]
    → [BELIEVE] —
            → (AGENT) → [*Null*]
            → (THEME) → [PROPOSITION:
                            [TIME-PERIOD1:#copper]^a
                                    → (CHRC) →      [EARLY]
                                    → (PART-OF) —
                                            → [TIME-PERIOD1:the Bronze Age]]
            → (ATTR) →   [COMMON]

[ENTER] —
    → (TIME) →          [TIME-PERIOD2]
    → (AGENT) →        [SUPERTYPE:{violence,war}]
    → (DEST) →          [HISTORY:*]
    → (INSTR) →         [WEAPON:{*}] → (MADE-OF) → [BRONZE]

[TIME-PERIOD1:#copper]^a
    → (ATTR) →          [PEACEFUL] → (CHRC) → [MODERATE]

(CAUSE3) —
    → [FINISH] —
            → (TIME) →          [TIME-PERIOD3: about 1000 BC]
            → (PATIENT) →    [TIME-PERIOD1:The Bronze Age]
    → [DISCOVER]
            → (AGENT) →       [MANKIND]
            → (THEME) →       [METAL]^b
                                    → (CHRC) → [DEADLY: more] —
                                                    → (THEME) → [*Null*]
[BE-1] —
    → (ENTITY)     → [METAL:#]^b
    → (ENTITY)     → [IRON:*]
    → (DURATION) → [TIME-PERIOD4]
```

Figure 4.3: Text Plan 2

```
[LIKE] —
   → (AGENT) → {AND} —
                    → [DOG:{*}] —
                          → (CHRC) → [YOUNG]
                    → [CAT:{*}] —
                          → (CHRC) → [YOUNG]
   → (THEME) → [PET-FOOD:Brand X]
```

Figure 4.4: The young dogs and cats like Brand X.

Structure 20:

```
CP  ->CBAR ->COMP -> —

         ->IP   ->DP   ->DBAR ->DET  -> definite plural

                             ->NP   ->NBAR ->NOUN ->NOUN -> dog

                                               ->CONJ -> and

                                               ->NOUN -> cat

                             ->DEGP ->DEGBA->DEG  -> —

                                   ->AP   ->ABAR ->ADJ  -> young

            ->IBAR ->INFL -> simple present

                  ->VP   -> like Brand X
```

Reading: The young dogs and cats like Brand X.

Structure 21:

```
CP  ->CBAR ->COMP -> —

        ->IP   ->DP   ->DBAR ->DET  -> definite plural

                      ->NP   ->NBAR ->NBAR ->NOUN -> dog

                                    ->CONJ -> and

                                    ->NBAR ->NOUN -> cat

                             ->DEGP -> young

             ->IBAR ->INFL -> simple present

                  ->VP   -> like Brand X
```

Reading: The young dogs and cats like Brand X.

Structure 22:

```
CP  ->CBAR ->COMP -> —

          ->IP   ->DP   ->DBAR ->DET  -> definite plural

                              ->NP   ->NP   ->NBAR ->NOUN -> dog

                                            ->DEGP -> young

                                     ->CONJ -> and

                                     ->NP   ->NBAR ->NOUN -> dog

                                                   ->DEGP -> young

                ->IBAR ->INFL -> simple present

                       ->VP   -> like Brand X
```

Reading: The young dogs and young cats like Brand X.

Structure 23:

```
CP   ->CBAR ->COMP -> —
           ->IP   ->DP   ->DBAR ->DBAR ->DET  -> definite plural
                                       ->NP   ->NBAR ->NOUN -> dog
                                              ->DEGP -> young
                                 ->CONJ -> and
                                 ->DBAR ->DET  -> definite plural
                                       ->NP   ->NBAR ->NOUN -> cat
                                              ->DEGP -> young
                 ->IBAR ->INFL -> simple present
                        ->VP   -> like Brand X
```

Reading: The young dogs and the young cats like Brand X.

Structure 24:

```
CP  ->CBAR ->COMP -> —
        ->IP   ->DP   ->DBAR ->DBAR ->DET  -> definite plural
                                    ->NP   ->NBAR ->NOUN -> dog
                                           ->DEGP -> young
                      ->CONJ -> and
                      ->DP   ->DBAR ->DET  -> definite plural
                                    ->NP   ->NOUN -> cat
                                           ->DEGP -> young
               ->IBAR ->INFL -> simple present
                      ->VP   -> like Brand X
```

Reading: The young dogs and the young cats like Brand X.

Structure 25:

```
CP  ->CP   ->CBAR ->COMP -> —

                ->IP   ->IBAR ->INFL -> simple present

                        ->VP   ->VBAR ->VERB -> [be]

   ->CONJ -> [although]
              [however ]

   ->CP   ->CBAR ->COMP -> —

                ->IP   ->IBAR ->INFL -> simple present

                        ->VP   ->VBAR ->VERB -> [think  ]
                                                [believe]
```

Reading: Although ____ is ____ , ____ thinks ____ .
Or: ____ is ____ ; however, ____ thinks ____ .

Structure 26:

```
CP   ->CBAR ->COMP -> —

          ->IP   ->IBAR ->INFL -> simple present

                        ->VP   -> [think  ]
                                  [believe]

   ->PP   ->PBAR ->PREP ->[despite]

                ->DP   -> -ing

                        ->IP   ->IBAR ->INFL -> simple

                                      ->VERB -> [be]
```

Reading: Despite ____ being ____ , ____ thinks ____ .

Structure 27:

```
CP  ->CP   ->CBAR ->COMP -> —

                 ->IP   ->IBAR ->INFL -> simple present

                        ->VP   ->VBAR ->VERB -> [end
                                                 finish
                                                 complete]

   ->CONJ -> [because]

   ->CP   ->CBAR ->IP   ->IBAR ->INFL -> simple present

                        ->VP   ->VBAR ->VERB -> [discover
                                                 found]
```

Reading: ____ ended, because ____ discovered ____ .

Structure 28:

```
CP  ->COMP -> —

    ->CBAR ->IP   ->DP   ->DBAR ->DET  -> indefinite

                         ->NP   ->NBAR ->NOUN -> [bronze]

                  ->IBAR ->INFL

                  ->VP   ->VBAR ->VERB -> [be]

                         ->DP   ->DBAR ->DET  -> indefinite

                                ->NP   ->NBAR -> [alloy
                                                  fusion
                                                  amalgam]

                                       ->PP   ->PBAR ->PREP -> of

                                              ->DP   ->DBAR ->DET  -> indefinite

                                                     ->NP   ->NBAR ->NOUN ->NOUN ->[copper]

                                                                          ->CONJ -> and

                                                                          ->NOUN ->[tin]
```

Reading: Bronze is an alloy of copper and tin.

Structure 29:

```
CP  ->CBAR ->COMP -> —

          ->IP   ->DP   ->DBAR ->DET  -> definite

                              ->NP   ->NBAR ->NOUN -> [metal]

                ->IBAR ->INFL -> simple past

                ->VP   ->VBAR ->VERB -> [be]

                              ->DP   ->DBAR ->DET  -> indefinite

                                            ->NP   ->NBAR ->NOUN -> [iron]
```

Reading: The metal was iron.

Structure 30:

```
CP  ->CBAR ->COMP -> —
         ->IP   ->DP   -> [the copper period_i]
                ->IBAR ->INFL -> simple present
                       ->VP   ->VBAR ->VERB -> [think / believe]
                                     ->CP   ->CBAR ->COMP -> —
                                            ->IP   ->DP   -> t_i
                                                   ->IBAR ->INFL -> -tense
                                                          ->VP   ->VBAR ->VERB -> [be]
                                                                        ->DegP -> part of [the Bronze Age]
```

Reading: The copper period is thought to be part of the Bronze Age.

Structure 31:

```
CP  ->CBAR ->COMP -> —

        ->IP   ->IBAR ->INFL -> simple present

                    ->VP   ->VBAR ->VERB -> [think  ]
                                             [believe]

                              ->CP   ->CBAR ->COMP -> that

                                        ->IP   ->DP   -> [the copper period_i]

                                               ->IBAR ->INFL -> simple present

                                                     ->VP   ->VBAR ->VERB -> [be]

                                                     ->DEGP -> part of [the Bronze Age]
```

Reading: It is thought that the copper period is part of the Bronze Age.

Structure 32:

```
CP  ->CBAR ->COMP -> —

        ->IP   ->DP   -> [the copper period_i]

               ->IBAR ->INFL -> simple past passive

                      ->VP   ->VBAR ->VERB -> [think  ]
                                              [believe]

                                    ->IP   ->DP   -> t_i

                                           ->IBAR ->INFL

                                                  ->DEGP -> part of [the Bronze Age]
```

Reading: The copper period is thought part of the Bronze Age.

Structure 33

```
CP  ->CBAR ->COMP -> —

         ->IP   ->DP   ->[the copper period]

                ->IBAR ->INFL -> simple past

                       ->VP   ->VBAR ->VERB -> [be]

                                     ->DEGP -> [fairly    ][peaceful]
                                               [moderately]
                                               [relatively]
```

Reading: The copper phase was relatively peaceful.

Chapter 5

Conclusion

5.1 Inadequacies of Conceptual Graphs

While using conceptual graphs, I was concerned by the fact that many concepts, such as [SUPERTYPE:{violence,war}], are not naturally represented in the hierarchic method of Aristotle. Unfortunately, it is not in the scope of this thesis to provide a commentary on the appropriateness of the type hierarchy or the choice of primitives to represent. Nor is it possible to provide an analysis of the expressive power of conceptual graphs.

However, the fact that certain natural language phenomena could be not clearly encoded in the formalism caused problems during ELOQUENCE's implementation. First, conceptual graphs do not distinguish between the status of conceptual relations. While all conceptual relations have equivalent status, the status of a thematic relation depends on the order of argument composition [see section 2.4.2].

The representation's second problem is more subtle. It has to do with sentential conjunction. Consider the following utterances:

1. The young dogs like brand X and the young cats like it.
2. The young dogs and the young cats like brand X.
3. The young dogs and young cats like brand X.
4. The young dogs and cats like brand X.

All four represent different realizations of the conjunction of two propositions. However, even with the extension to different conceptual graphs introduced in section 2.5.2, only 1 and 2 can be directly represented. ELOQUENCE determines if sentences like 3 are well-formed by checking that the quantifier scopings of the two propositions match. To determine if 4 is well-formed ELOQUENCE must recursively match the quantifier scopings and all features subsumed by the conjunction. In particular, it must check that both the dogs and the cats are young before validating utterance 4. To directly represent the full range of sentential conjunctions, which includes all possible lexical and phrasal categories, would require a representation equivalent to a high-order logic.

5.2 Contributions

ELOQUENCE makes a clean separation between text planning and realization, without restricting realization to post-processing. It can interact constantly with the text planner through shared data structures. Thus constraints can be applied when appropriate. This provides more flexibility than either post-processing or interleaving. An interleaved approach allows the planner to take advantage of the effects part of a text's realization, while planning the rest of the text. ELOQUENCE can provide the planner with the range of effects that would result before the generation system commits itself to a text plan.

The choices of linguistic form available to ELOQUENCE are neither trivial nor arbitrary. Rather, they are derived from a principled theory of language, namely Government and Binding Theory. Both predicting how decisions of linguistic form affect the availability of future choices, and determining how close texts are to an optimal realization remain open issues. By maintaining multiple realizations, ELOQUENCE takes a "generate and test" approach. It is not clear that a less complex algorithm exists. When the number of possible realizations becomes too large ELOQUENCE tries to intelligently limit its search.

5.3 Further Research

Little work has been done on the nature of the realization process, since interest shifted to text planning. Although ELOQUENCE embodies an innovative approach to control and variation in realization, there are still many issues that need to be addressed, including the following:

- Stylistic grammars: Research is needed on the pragmatic effects of choices in linguistic form. DiMarco and Hirst [1988], while working in the context machine translation, have started to address this deficiency by developing stylistic grammars (goal-directed grammars) for French and English. Replacing ELOQUENCE's current evaluator with a full stylistic parser would improve the accuracy of its evaluations. The main algorithm would still be "generate and test". Since the aims of goal-directed grammars are orthogonal to those of standard semantically directed grammars, it is unreasonable to expect that they can be expressed in a single generative formalism.

- Shared realization structures: Despite the inherent difficulties, the possibility of having realization processes share common sub-structures is worth investigation. An algorithm would have to be found to recognize when two processes are realizing a sub-plan in the same manner. If an efficient method can be developed to do this it would have two major benefits. First, it would greatly reduce the memory and processing requirements for the processes. Second, the search space could be pruned more effectively. Given a set of structures that vary at the start of a plan and then converge, the system could evaluate the structures at the point of convergence.

- Raced realization: Currently ELOQUENCE only assigns one of three priorities to each process ("active", "waiting for more planning", or "killed"). However, the system is designed to support any number. As an alternative to pruning the search space, one could take advantage of a range of priorities to race the processes. At any given time, the most promising realizations would have the highest priorities. McRoy [1987] achieved an analogous result with parsing. Processing in her parser is influenced by limitations on working memory and structural preferences, such as Right Association, Minimal Attachment, Revision as Last Resort, and verb-frame preferences.

- Phonetic realization: Although ELOQUENCE does not account for phonological effects, they are as significant as syntactic effects. If ELOQUENCE is to be extended, a goal-directed grammar needs to be developed for phonology. The evaluator would stylistically parse both syntax and phonological structures of a possible realization to determine its appropriateness.

References

Steven Abney, 1987. "The English Noun Phrase in Its Sentential Aspect" Ph.D. dissertation, Department of Linguistics and Philosophy, MIT.

James F. Allen, 1983. "Maintaining Knowledge about Temporal Intervals." *Communications of the ACM* Vol. 26, No. 11, pp. 832-834.

James F. Allen and Johannes A. Koomen, 1983. "Planning Using a Temporal World Model." *Proceedings of the Eighth International Joint Conference on Artificial Intelligence, Kalsrule, West Germany*, pp. 741-747.

Douglas E. Appelt, 1985a. *Planning English Sentences.* Cambridge, MA: Cambridge University Press.

Douglas E. Appelt, 1985b. "Planning English Referring Expressions." In Grosz, *et al.*, 1985: pp. 501-517.

Douglas E. Appelt, 1985c. "Some Pragmatic Issues in the Plannning of Definite and Indefinite Noun Phrases." *Proceedings of the Twenty-Third Annual Meeting of the Association for Computational Linguistics, Chicago, Illinois*, pp. 69-97.

Douglas E. Appelt, 1988. "Referring Expressions." In McDonald, *et al.*, 1988: pp. 69-97.

Nicholas Asher, 1987. "A Typology for Attitude Verbs and Their Anaphoric Properties." *Linguistics and Philosophy*, Vol. 10, No. 2, pp. 125-197.

Nicholas Asher, 1986. "Belief In Discourse Representation Theory." *Journal of Philosophical Logic*, Vol. 15, No. 2, pp. 127-189.

Rainer Bäuerle, U. Egli and A. von Stechow (Editors), 1979. *Semantics from Different Points of View.* New York: Springer-Verlag Berlin Heidelberg.

Rainer Bäuerle, 1979. "Questions and Answers." In Bäuerle, *et al.*, 1979: pp. 61-74.

Michael Brady and Robert C. Berwick (Editors), 1983. *Computational Models of Discourse.* The MIT Press.

Charles Brown, T. Pattabhiraman, Michel Boyer, Diane Massam, and Veronica Dahl, 1986. "Tailoring Government and Binding Theory for Automatic Natural Language Generation." Technical Report 86-4, Simon Fraser University, Burnaby BC.

Charles Brown, T. Pattabhiraman, Michel Boyer, Diane Massam, and Veronica Dahl, 1986. "Tailoring Conceptual Graphs for use in NL Translation." Technical Report 86-14, Simon Fraser University, Burnaby BC.

Charles Brown, T. Pattabhiraman, Pierre Massicotte, 1987. "Tailoring Conceptual Graphs for use in NL Translation." Techinical Report 87-8, Simon Fraser University, Burnaby BC.

Barbara Brunson, 1987. "Constraints on Discontinuous Theta-Roles" *Toronto Working Papers in Linguistics*, Vol. 8, pp. 1-19.

Noam Chomsky, 1982. *Some Concepts and Consequences of the Theory of Government and Binding.* Cambridge, MA: MIT Press.

Noam Chomsky, 1984. *Lectures on Government and Binding.* Dordrecht: Foris Publications.

Noam Chomsky, 1986. *Barriers.* Cambridge, MA: MIT Press.

Philip R. Cohen and Hector J. Levesque, 1985. "Speech Acts and Rationality." *Proceedings of the Twenty-Third Annual Meeting of the Association for Computational Linguistics, Chicago, Illinois*, pp. 49-60.

E. Jeffrey Conklin and David D. McDonald, 1982. "Salience: The Key to the Salience Problem in Natural Language Generation." *Proceedings of the Twentieth Annual Meeting of the Association for Computational Linguistics, Toronto, Ontario.* pp. 129-135.

Elizabeth A. Cowper, 1987. "Pied Piping Issues in the Planning of Definite and Indefinite Noun Phrases." *The Canadian Journal of Linguistics*, Vol. 32, No. 4, pp. 1-41.

Chrysanne DiMarco and Graeme Hirst, 1988. "Stylistic Grammars in Language Translation." *To Appear in Coling 88.*

David R. Dowty, Robert E. Wall, and Stanley Peters, 1981. *Introduction to Montague Semantics.* D. Reidel Publishing Company.

Dov Gabbay and Christian Rohrer, 1979. "Do We Really Need Tenses Other Than Future and Past." In Bäuerle, *et al.*, 1979: pp. 15-20.

Barbara J. Grosz, Karen Sparck Jones and Bonnie Lynn Webber (Editors), 1985. *Readings in Natural Language Processing.* Morgan Kaufmann Publishers.

M. A. K. Halliday, 1985. *An Introduction to Functional Grammar.* London: Edward.

William O. Hendricks, 1976. *Style and Discourse Analysis.* North-Holland Publishing Company.

Graeme Hirst, 1987. *Semantic interpretation and the resolution of ambiguity.* Cambridge, Cambridge University Press.

Eduard Hovy, 1987a. "Some Pragmatic Decision Criteria in Generation." In Kempen, 1987: pp. 3-17.

Eduard Hovy, 1987b. "Generating Natural Language Under Pragmatic Constraints." Ph.D. dissertation, RR #521 Department of Computer Science, Yale University.

Eduard Hovy, 1988a. "Planning Coherent Multisentential Text." *Proceedings of the Twenty-Sixth Annual Meeting of the Association for Computational Linguistics, Buffalo, New York*, pp. 163-169.

Eduard Hovy, 1988b. "Two Types of Planning in Language Generation." *Proceedings of the Twenty-Sixth Annual Meeting of the Association for Computational Linguistics, Buffalo, New York*, pp. 177-186.

Ray Jackendoff, 1972. *Semantic Interpretation Generative Grammar.* Cambridge, MA: MIT Press.

Ray Jackendoff, 1987. *Conscisousness and The Computational Mind.* Cambridge, MA: MIT Press.

Hans Kamp, 1979. "Events, Instants and Temporal Reference." In Bäuerle, *et al.*, 1979: pp. 376-418.

Robin F. Karlin, 1988. "Defining the Semantics of Verbal Modifiers in the Domain of Cooking Tasks." *Proceedings of the Twenty-Sixth Annual Meeting of the Association for Computational Linguistics, Buffalo, New York*, pp. 61-67.

Gerard Kempen (Editor), 1987. *Natural Language Generation.* Dordrecht: Martinus Nijhoff Publishers.

Doug Lenat, Mayank Prakash, and Mary Shepherd, 1987. "CYC: Using Common Sense Knowledge to Overcome Brittleness and and Knowledge Acquisition Bottlenecks." *AI Magazine*, Vol. 6, No. 4, pp. 65-88.

Ivan Lowe, 1987. "Two Ways of looking at Cause and Reasons." In Monaghan, 1987: pp. 37-46.

William C. Mann and Sandra A. Thompson, 1987a. "Antitesis: A Study on Clause Combinig and Discourse Structure". USC/Information Sciences Institute, Technical Report RR-87-171, April 1987.

William C. Mann and Sandra A. Thompson, 1987b. "Rhetorical Structure Theory: Description and Construction of Text Structures." In Gerard Kempen, 1987: pp. 85-95.

William C. Mann, 1984. "Discourse Structures for Text Generation." *Proceedings of Coling 84, Stanford, California*, pp. 367-375.

William C. Mann, 1987. "Text Generation: The Problem of Text Structure." USC/Information Sciences Institute, Technical Report RR-87-181, March 1987.

Christian Matthiessen, 1984a. "Systemic Grammer in Computation." USC/Information Sciences Institute, Technical Report RR-83-120, February 1984.

Christian Matthiessen, 1984b. "How to Make Grammatical Choices in Text Generation." USC/Information Sciences Institute, Technical Report RR-83-120, February 1984.

Christian Matthiessen, 1987. "Notes on the Organization of the the Environment of a Text Generation Grammar." USC/Information Sciences Institute, Technical Report RR-87-177, April 1987.

David D. McDonald, 1983. "Natural Language Generation as Computational Problem: An Introduction." In Brady, *et al.*, 1983: pp. 209-266.

David D. McDonald, 1985. "Description Directed Control: Its Implications for Natural Language Generation." In Grosz, *et al.*, 1985: pp. 519-537.

David D. McDonald and James D. Pustejovsky, 1985a. "A Computational Theory of Prose Style for Natural Language Generation." *Proceedings of the Second Conference of the European Chapter for Computational Linguistics, Geneva, Switzerland*, pp. 187-193.

David D. McDonald and James D. Pustejovsky, 1985b. "TAG's as a Grammatical Formalism for Generation." *Proceedings of the Twenty-Third Annual Meeting of the Association for Computational Linguistics, Chicago, Illinois*, pp. 94-102.

David D. McDonald and James D. Pustejovsky, 1985c. "Description-Directed Natural Language Generation." *Proceedings of the Elventh International Joint Conference on Artificial Intelligence, Los Angeles, California.* pp. 799-805.

David D. McDonald, Marie M. Vaughan and James D. Pustejovsky, 1987. "Factors Contributing to Efficiency in Natural Language Generation." In Kempen, 1987: pp. 159-181.

Kathleen R. McKeown, 1982. "The Text System for Natural Language Generation: An Overview." *Proceedings of the Twentieth Annual Meeting of the Association for Computational Linguistics, Toronto, Ontario*, pp. 113-120.

Kathleen R. McKeown, 1983. "Recursion in Text and Its Use in Language Generation." *Proceedings of the Second National Conference on Artificial Intelligence*, pp. 270-273.

Kathleen R. McKeown, 1985a. *Text Generation.* Cambridge, MA: Cambridge University Press.

Kathleen R. McKeown, 1985b. "Discourse Strategies for Generating Natural-Language Text." In Grosz, *et al.*, 1985: pp. 479-499.

Susan Weber McRoy, 1987. "Generating English Sentences from Semantic Structures." MSc. Thesis,CSRI TR-209 Department of Computer Science, University of Toronto.

James Monaghan (Editor), 1987. *Grammar in the Construction of Text.* London: Frances Printer Publishers.

David Pesetsky, 1982. "Paths and Categories." Ph.D. dissertation, Department of Linguistics and Philosophy, MIT.

James D. Pustejovsky, and Sergei Nirenburg, 1987. "Lexical Selection in the Process of Language Generation." *Proceedings of the Twenty-Fifth Annual Meeting of the Association for Computational Linguistics, Stanford, California*, pp. 201-206.

Manny Rayner and Amelia Banks, 1988. "Parsing and Interpreting Compartives." *Proceedings of the Twenty-Sixth Annual Meeting of the Association for Computational Linguistics, Buffalo, New York*, pp. 49-60.

Remko Scha and David Stallard, 1988. "Multi-level Plurals and Distributivity." *Proceedings of the Twenty Annual Meeting of the Association for Computational Linguistics, Buffalo, New York*, pp. 17-32.

John Searle, 1969. *Speech Acts: An Essay in the Philosophy of Language.* Cambridge UK: Cambridge University Press.

John Searle, 1979. *Expression and Meaning: Studies in the Theory of Speech Acts.* Cambridge UK: Cambridge University Press.

Peter Sells, 1985. *Lectures on Contemporary Syntactic Theories: An Introduction to Government-Binding Theory, Generalized Phrase Structure Grammar, and Lexical-Functional Grammar.* Center for the Study of Language and Information, Standford University.

Robert Simmons and Jonathan Slocum, 1972. "Generation from Semantic Nets." *Communication of the ACM*, Vol. 15, No. 10, pp. 891-904.

John F. Sowa, 1984. *Conceptual Structures: Information Processing in Mind and Machine.* Addison-Wesley Publishers.

Johan Van Benthem, 1985. *A Manual of Intensional Logic.* The Center for the Study of Language and Information.

Maria M. Vaughan and David D. McDonald, 1986. "A Model of Revision in Natural Language Generation." *Proceedings of the Twenty-Fourth Annual Meeting of the Association for Computational Linguistics, New York, New York* pp. 90-96.

Harry K. T. Wong, 1975. "Generating English Sentences from Semantic Structures." MSc. Thesis, Department of Computer Science, University of Toronto.