Speaking of Actions: Choosing Rhetorical Status and Grammatical Form in Instructional Text Generation

Keith Vander Linden

CU-CS-654-93 July 1993

University of Colorado at Boulder
DEPARTMENT OF COMPUTER SCIENCE

Speaking of Actions: Choosing Rhetorical Status and Grammatical Form in Instructional Text Generation

Keith Vander Linden

CU-CS-654-93 July 1993

University of Colorado at Boulder

Technical Report CU-CS-654-93
Department of Computer Science
Campus Box 430
University of Colorado
Boulder, Colorado 80309

Speaking of Actions: Choosing Rhetorical Status and Grammatical Form in Instructional Text Generation

by

Keith Neil Vander Linden

B. A., Central College, 1983
M. S., University of Iowa, 1985

A Thesis submitted to the
Faculty of the Graduate School of the
University of Colorado in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Computer Science

June 1993

In memory of my father

Abstract

This thesis addresses a problem fundamental to the development of text generation systems, the problem of managing diversity of expression at the textual level. In any genre, writers employ an extensive set of lexical and grammatical forms of expression, choosing the one they feel will be the most effective given the specific communicative context. The task of the text generation researcher is to identify these forms and the contexts in which they are used, and then to encode this information in the text generation system itself.

A temporary solution to these problems, one which is typical of current text generation systems, is for the researcher to choose a small set of lexical and grammatical forms that are sufficient to disambiguate the types of information that the system must express. It is then a simple matter of allowing type of information to determine the appropriate expressional form.

These simplifications allow the researcher to address other issues in generation, but do not provide an effective way to manage diversity of expression. The current study specifically addresses this problem. It contends that the only principled way to identify both the relevant expressional forms and the contexts in which they are used, is to perform a detailed linguistic study of a corpus of text from the relevant genre. This thesis proposes the following four step process for identifying both the relevant forms of expression and the contexts in which they are systematically used: (1) Collect a corpus of the relevant text type; (2) Perform a detailed linguistic study of this corpus; (3) Implement the results of this study in a text generation system; (4) Compare the output of the system with the text found in the corpus. This process is performed iteratively until a sufficient match is found between the generator's output and the original corpus text.

This approach is demonstrated in the context of instructional text, resulting in the construction of IMAGENE, an instructional text generation system. IMAGENE embodies a model of the effective forms of expression consistently used by instructional text writers over a broad range of instruction types.

Contents

List of Figures

List of Tables

Acknowledgements

A project of this magnitude can only succeed in the appropriate research environment, particularly when the project is interdisciplinary in nature as this one is. In this regard, I would like to thank my thesis advisors Jim Martin from Computer Science and Susanna Cumming from Linguistics, both of whom had a deep impact on this work. I thank them for lending their considerable expertise from both sides of the Computational Linguistics fence, and for being flexible in allowing the work to find its own balance between Computational and Linguistic concerns. In retrospect, it is simply unclear, in some cases, just whose brain child a particular concept is. I take this as a characteristic of how active and productive our collaboration has been.

I would also like to thank my committee members, Paul Smolensky, Clayton Lewis, and Gerhard Fischer, for lending their own unique viewpoint throughout this project. Their unwillingness to allow me the luxury of making simplifications or of ignoring relevant work from related fields such as Connectionism, Psychology, and Human Computer Interaction, has immeasurably improved this work.

Thanks are due to the generation community as a whole for setting the stage for this work and welcoming a new kid on the block. I would particularly like to thank the Penman Language Generation group at USC/ISI for long-distance help with taming MacPenman. I also gratefully acknowledge financial support from NSF Grant IRI-9109859, and the Institute of Cognitive Science and Department of Computer Science of the University of Colorado, Boulder.

I would like to thank my family for their support throughout these many years of work. Thanks to my mother, Rua, for always believing in my ability to accomplish what I set my mind to, in spite of what I perceived as incontrovertible evidence to the contrary. Thanks also to my father, Lee, to whose memory this thesis is dedicated. My two sons, Lukas and Christian, have given me the opportunity to watch some long-forgotten reruns through many an endless night, but through it all have given me a joy that school work and conferences cannot provide. Finally, and most importantly, thanks to my wife, Brenda, for contributing much more than "IMAGENE", the computer program's name. She lent me her patience when I had lost mine and built me up when I was run down. I could not have completed this project without her.

Chapter 1

Introduction

This thesis addresses a problem fundamental to the development of text generation systems, the problem of managing diversity of expression at the textual level. In any genre, writers employ an extensive set of lexical and grammatical forms of expression, choosing the one they feel will be the most effective given the specific communicative context. The task of the text generation researcher is to identify these forms and the contexts in which they are used, and then to encode this information in the text generation system itself.

A temporary solution to these problems, one which is typical of current text generation systems, is for the researcher to choose a small set of lexical and grammatical forms that are sufficient to minimally communicate the types of information that the system must express. It is then a simple matter of allowing the type of information to determine the appropriate expressional form. For example, if the text generator must express purpose relationships, then a single, unique purpose form, such as the non-fronted "in order to" infinitive clause, will be implemented in the generator for use in this context. Any time a purpose relation must be expressed, this form will be used, as in "Do X in order to achieve purpose Y."

This is a partial solution to the problems involved in constructing a text generation system. A set of forms is provided to the generator, one of which is used to express purpose, another to express condition, and yet another to express result, etc. There are, however, two problems that typically arise in such an approach. First, the methods for determining the small set of expressional forms are largely ad hoc, being based on introspective analyses of the particular text genre. The resulting forms, although not bad, are far from realistic. Secondly, even if the researcher provides some alternate forms, the text generator is typically not capable of utilizing all the contextual information relevant to making the rhetorical and grammatical choice among them. Various textual and interpersonal issues, for example, have great effect upon the lexical and grammatical form of the text, but are seldom allowed to act in this capacity in current generation systems.

These simplifications allow the researcher to address other issues in generation, but ride rough-shod over the problem of managing diversity of expression. The current study specifically addresses this problem. It contends that modeling the diversity employed by writers is important for the generation of effective text. This is based on the observation that diverse forms of expression are, in fact, used by writers, and, more importantly, are used systematically. Further, it contends that the only principled way to identify both the relevant expressional forms and the contexts in which they are used is to perform a detailed linguistic study of a corpus of text from the relevant genre. This is based on the difficulty of using introspective analyses in this context. Introspective analyses of language cannot be expected to shed enough light on language in use. Consider, for example, purpose expressions in instructional texts, the specific genre studied in this thesis. An introspective analysis of such expressions results in the following partial list of conceivable purpose expressions:

(1) Pull out sharply, *in order to remove the phone.*

(2) *To remove phone*, pull out sharply.

(3) Pull out sharply *for phone removal.*

(4) Pull out sharply *for removing the phone.*

(5) *For the phone,* pull out sharply.

(6) *Remove phone by* pulling out sharply.

(7) *Remove the phone.* Pull out sharply.

(8) The purpose of pulling out sharply is *to remove the phone.*

(9) Pulling out sharply achieves the purpose of *removing the phone.*

(10) *Removing the phone involves* pulling out sharply.

(11) *The method for removing the phone* is to pull out sharply.

As one quickly realizes, this list could be expanded ad infinitum, bringing us face to face with the fundamental characteristic of language: it is infinite. Introspective analyses provide no principled means of determining which of these forms would be the most effective in any given communicative context. As an alternative, this thesis proposes and demonstrates the following four step process for identifying both the relevant forms of expression and the contexts in which they are used:

1. Collect a corpus of text from the relevant genre, and encode a full range of the rhetorical and grammatical features of all of the text in a convenient form;

2. Perform a linguistic analysis of part of the corpus. This analysis involves determining the range of forms used in the corpus and then using an iterative cycle of hypothesis formation and testing to determine the communicative contexts in which each is used;

3. Implement the results of this analysis in the text generation system;

4. Compare the output of the system with the text found in the corpus, differentiating between the predictions concerning text that was specifically used in the analysis, called the training set, and text that wasn't, called the testing set.

This process begins and ends with the corpus, providing an empirically based approach to identifying the range of rhetorical and grammatical forms that are used in real text and to determining the contextual issues that are relevant to choosing among them. It is repeated until the output of the system matches the text in the training portion of the corpus to a sufficient degree. The system's predictions concerning the remainder of the corpus are then tested, giving a measure of how far the predictions can legitimately be applied. The overriding assumption in all of this work is that the forms consistently used by technical writers are those which are the most effective for expressing actions.

This introduction will give an overview of how these four steps have been applied in the domain of instructional text. It will discuss how a corpus of instructional text was collected and represented, how the corpus analysis of action expressions was performed, how the results were implemented in IMAGENE, an instructional text generation system, and how well IMAGENE's predictions match the text in the corpus. It will conclude with an outline of the structure of the thesis.

1.1 Speaking of Actions

The first step of the four step approach advocated in this thesis is the collection of a corpus of the text passages from the relevant genre. As mentioned above, focusing on a particular genre is important because a general solution to the problem of managing the diversity of expression in all text is far beyond the state of the art in text generation. The current study has focused on instructional text. The predictions resulting from this study are, therefore, much more detailed than would be possible with a broader target text type.[1]

This section will discuss the specific features of instructional text that have been addressed in the current study, giving several examples which will serve as illustrations both of the complexity of the

[1]The importance of this concern for the characteristics of a specific genre was termed *domain communication knowledge* by Rambow (Rambow, 1990; Kittredge et al., 1991).

procedural relationships that are routinely expressed in instructional text and the great variation of forms that are used to express them. It will then justify the use of text from instructional manuals, a text type reputed to be very poor, as a source of characteristics of good instructional text. Finally, it will review how the relevant features of the corpus were represented.

1.1.1 Instructional Text

In this thesis, the definition of *instructional text* is more restrictive than is commonly the case, referring exclusively to written, procedural directions prescribing the performance some sequence of actions to the reader. It does not include what will be termed *general instructional material*, such as text books and complex object descriptions. Neither does it include interactive discourse, even of an instructional nature. Obviously, this latter type of text is closely related, but has significantly different characteristics resulting from the distinct features of the interactive context.

Instructional text is a common part of everyday life. Because we repeatedly find ourselves dealing with it, the automation of its generation in the most effective way is greatly desirable, both from the point of view of the manufacturer who must produce it, and also the consumer who must read it. This automation, however, requires a deep knowledge of the rhetorical and grammatical forms of expression that most effectively convey actions and the procedural relationships among them.

This deep knowledge, however, is extremely difficult to determine because the function of instructional text itself is extremely difficult. It must employ the largely linear medium of text to express actions with very distinct characteristics, including varying duration, possible concurrency, different agents, complex conditions of execution, and varying levels of familiarity to the reader. This complexity of the subject matter and the communicative environment has motivated language users to develop a diverse array of expressional forms for expressing the functions in context. Of many lexical and grammatical issues of choice that could be studied in the domain of instructional text, this thesis addresses how this complexity affects expressional form at the rhetorical level in instructional text: How are the procedural relations between actions expressed at the rhetorical level in instructional text?

The corpus developed for this study was taken from various types of instructional text, including instruction booklets, recipes, and auto-repair manuals. It contains approximately 1000 clauses (6000 words) of instructions. These sources of text represent a vast array of process types, but all involve the expression of actions and the procedural relationships between them.

1.1.2 Some Sample Passages

This section will discuss a number of sample passages taken from various sources of instructional text. These samples indicate some of the procedural issues that the current study deals with and the range of rhetorical and grammatical forms that are used to express them. They will also serve as running examples throughout the thesis. IMAGENE's approach to their expression will be discussed in appropriate chapters throughout the thesis and then reviewed in the conclusion.

Here is an excerpt from a cordless telephone manual (Code-a-phone, 1989) which will be called the Installed-Phone text:

Installed-Phone Text:

> When the 7010 is installed and the battery has charged for twelve hours, move the OFF/-STBY/TALK [8] Switch to STBY. The 7010 is now ready to use. Fully extend the base antenna [12]. Extend the handset antenna [1] for telephone conversations.

This span of text includes expressions of preconditions, results, purposes and sequential actions, employing a large array of expressional forms. The problem is to discern the factors that affect the choice of rhetorical and grammatical form. Why was the installing action expressed as a precondition ("When the 7010 is installed") and not an imperative action? What is the procedural relationship between the action of making phone calls ("for phone conversations") and the extending action in the last sentence ("extend the handset antenna"), and what determined the particular nominalized expressional form used there?

Here is an excerpt from the instructions for the GTE Airfone (Airfone, 1991) which will be called the Remove-Phone text:

Remove-Phone Text:

> When instructed (approx. 10 sec.) remove phone by firmly grasping top of handset and pulling out. Return to seat to place calls.

The text span gives an example of just one of the many possible variations of expressional form that are possible in instructional text. The purpose of removing the phone is actually stated as an imperative, rather than as a "to" infinitive, ("remove the phone") and the sub-actions are expressed in participial form in a by prepositional phrase ("by firmly grasping top of handset and pulling out"). The purpose of placing calls ("to place calls") is not fronted and expressed as a "to" infinitive. The problem, then, is to determine the contextual features that motivate the choice of these forms as opposed to alternate forms.

The final excerpt is from a instruction manual for a common household furnace which will be called the Furnace text:

Furnace Text:

> Depress knob and hold for 60 seconds after pilot has been lighted. Release knob and turn to ON position.

This process includes actions that are to be performed concurrently. The point of interest here is the procedural status of the action of lighting of the pilot ("after pilot has been lighted"). Is the reader expected to manually light the pilot or not? Because earlier text indicates that the pilot is a manual lighting pilot, the reader is indeed expected to perform the lighting, making it clear that the rhetorical status and grammatical form used in the expression of the lighting action fail to effectively convey the procedural relationships involved. More will be said about this example in Chapter 8.

These few examples illustrate the complexity of actions and their procedural relationships and the variety of lexical and grammatical forms with which they can be expressed. The question is how to first determine the appropriate range of forms to generate and then choose from among them the most effective rhetorical and grammatical form for expressing each action in context. A number of approaches to this problem can be found in the Cognitive Science literature. Linguistics has provided the theoretical basis for the analysis of relations between contextual features and linguistic forms, but has done little work specifically on instructional text. Psychology has developed an experimental approach dedicated to testing the relative efficacy of various expressional forms in experimental settings. Artificial Intelligence has focused on the use of automated planners for the construction of plans to be expressed. The place of the current study in this range of study will be reviewed in Chapter 2.

1.1.3 Aren't Instructions Poorly Written?

At this point, it is important to address the claim that because instructions are poorly written, they should not serve as a source for good writing practices at all. Wright (1981) has identified three areas in which she feels instructions are poorly written:

- Some information is out of date or simply wrong.
- Some information is incomprehensible.
- Some information is poorly structured.

It is certainly impossible to deny that these indictments are, to some degree, true of instructions; we all have personal experience that affirms the allegations. The question is, do the allegations prevent us from using a corpus study to determine effective forms of lexical and grammatical expression for actions? The answer in the context of the current study is 'no'. The first two of Wright's indictments will now be discussed with respect to the current study; the last, the problem of poorly structured information, does not pertain to the concerns of the current study.

Information is rendered out of date when the instructions refer to processes or objects that no longer pertain to the device. This, however, is not an indictment of the expressional form of the action expressions at all; the grammatical form of an imperative or of a referring expression is not rendered useless because it is occasionally used to refer to a action or object that is out of date. The expressional form is still useful, provided it expresses relevant information. Thus, this concern does not affect the results of the current study concerning the form of expression of procedural relations in instructions. Further, note that the automation of instructional text generation is specifically designed to alleviate this problem by allowing up to date manuals to be produced more easily.

Information is rendered incomprehensible when the terms of the expression are not readily understandable to the reader. Wright points out that instructions may employ action verbs that are hard to understand, as in using "irrigate the eye" rather than "wash the eye with cold water" (Wright, 1981, page 132). At the level of rhetorical and grammatical form, however, these two expressions are equivalent; they are both imperatives, thus preserving the results of the current study.

Information is also rendered incomprehensible when the process is simply hard to discern from reading the text (Wright, 1981; Rettig, 1991). This is primarily an indictment of language in general, not specifically of expressional form; it is simply hard to write about complex, possibly concurrent processes, even when using the appropriate lexical items and grammatical forms. The deictic capabilities of written language, for example, are limited. Certainly, there are cases where inappropriate forms are used, such as in the Furnace text, but such cases are infrequent; the forms of expression in instructional text are typically quite good. These problems make it attractive to avoid verbal instructions if possible, but do not deny the importance of understanding the most effective forms of expression for those cases where verbal instruction is necessary.

Therefore, although instructions have a reputation of being a bad source for principles of effective expression, a corpus-based study of a broad range of instructional texts, as produced by trained technical writers, will uncover effective forms of expression at the rhetorical level and also the precise contexts in which they are used. Examples of text that appear to be poorly written, like the Furnace text (given above), are typically isolated examples of the poor instructional form that are not repeated on a consistent basis in instructional manuals. Thus, the result of performing a rhetorical and grammatical analysis of a broad range of instructional text will be a set of generalizations concerning the forms that technical writers have found most useful in expressing specific types of actions and procedural relationships.

1.1.4 Representation of the Corpus

This section will discuss the methods for representing the grammatical and rhetorical forms of the instructions in the corpus. Both of these aspects of the text in the corpus are represented in a relational database for text. The database is made up of three tables, representing information on clauses, noun and prepositional phrases, and rhetorical propositions. The details of the structure of this database are discussed in Chapter 3. The nature of the information represented is outlined here.

The representation for the grammatical form of the clauses and phrases is based on traditional principles of syntax and semantics. Clauses and phrases are represented in separate tables. Links within the clause table are used to indicate subordinate relations, and links between the clause and phrase tables are used to represent relative clauses and predicate-argument relationships. The goal was to represent any element of the semantic or syntactic context that might be relevant in the analysis.

The rhetorical description coded in the corpus requires a bit more explanation. Mann and Thompson's Rhetorical Structure Theory (RST) has been used to encode the rhetorical relationships between expressions in the corpus (Mann and Thompson, 1989; Mann and Thompson, 1988; Mann and Thompson, 1987a). There are, of course, a number of schemes for representing text and discourse structures, which will be outlined in Chapter 2. The following attributes of RST, which will be discussed in turn, were the most attractive:

- It was specifically designed for the description of text rather than interactive discourse;

- It focuses on relations between spans of text rather than on intentions;
- It is a flexible representation of various types of rhetorical relations;
- It represents various levels of the textual hierarchy;
- It is able to represent relations between elements of equal importance, called joint relations, and relations between elements of unequal importance, called nucleus-satellite relations;
- It focuses on writer oriented, functional definitions of the relations, rather than on grammatically oriented definitions. This allows some relations to have no surface grammatical cues.
- It has been and continues to be extensively used in the computational generation community both as a descriptive and a constructive tool.

The current study focuses on written text, rather than interactive discourse. RST has this same focus. Although there has been interest in extending RST to address interactive discourse, this was never done, primarily because RST's mechanisms are not suitable for representing the unique features of interactive discourse such as interruptions and non-verbal cues. RST's constructs are well tailored for the text analysis performed in the current study.

There is considerable debate in the discourse representation field concerning the representation of intentions and rhetorical relations (Moore and Pollack, 1992), most representation systems focusing on one or the other. Instructional text tends to have a fairly simple intentional structure, and a more complex rhetorical one. This matches RST's explicit representation of the rhetorical relations, with the intentional information being represented implicitly.

The current study requires a representation system that is tailorable and extendable to the domain of instructional text. RST provides this because of the possibility of adding new rhetorical relations that are of the appropriate granularity of representation. As will be seen later, RST was modified slightly in order to represent precisely the rhetorical issues that became relevant to the current study. The exploitation of this modifiability, also exploited by Rösner and Stede (1992a), is seen by some as a source of problems for the field of discourse analysis (Knott and Dale, 1992), but, given the lack of a comprehensive theory of discourse relations, was a practical necessity for the current study.

One criticism that has been made of the flexibility of RST analyses is that they are ambiguous, that is, there may be more than one analysis of the same text. This problem is most evident in analyses of the macro-structure of persuasive texts, where the motives of the writer are not always clear. The specific genre studied in the current study tends not to exhibit this problem. First, the analysis is done at the local level, that is at the leaf level of the RST tree, where clauses and phrases are related, rather than at the highest levels, where large text spans are related. At this level, the difficulties of macro-level analysis are not relevant. Secondly, the analysis was done on written instructional texts, which have a heavy reliance on the structure of the process being expressed. The ambiguity which may arise in the RST analysis of some genres tends not to occur here.

RST represents text hierarchically, using the same set of rhetorical relations at each level. Although the current study addresses the local rhetorical level, the capability to address structure at lower and higher levels is important for future work on macro-structure of instructional text.

Instructional text includes rhetorical relations that specify a most important element (e.g., purpose clauses) and those in which the elements are equally important (e.g., action sequences). RST provides the nucleus-satellite and the joint schemas respectively to address these issues.

RST relations may be marked grammatically in a number of ways, perhaps with no markings at all. This was important in the current study because of the various grammatical forms that were considered for each of the rhetorical relations in question. It would be inappropriate to require that certain grammatical forms be present before positing a particular relation.

The features of RST mentioned so far are not unique to RST, but rather can be traced back to a number of earlier studies. Precursors to RST include Beekman et al, (1974; 1981), Longacre (1976; 1983), Reichman (1981; 1984), Grimes (1975), Martin (1983), and Hobbs (1982; 1979; 1985). More complete analyses of the contributions of these works to RST can be found elsewhere (Mann and Thompson, 1989; Mann and Thompson, 1986). The unique contribution of RST is that it is a particularly well developed

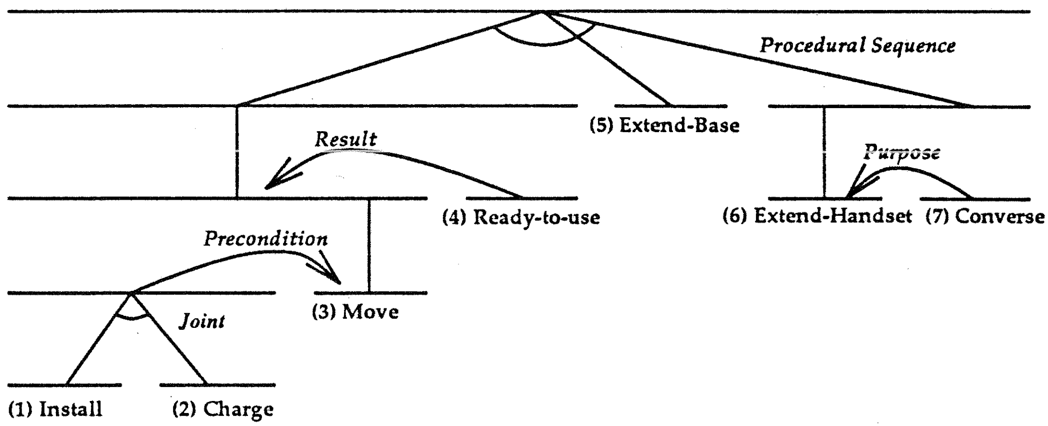

Figure 1.1: The RST Representation of the Installed-Phone Text

result of this previous work, incorporating all of the most important features. For a more recent look at interaction between these various lines of work, see Mann and Thompson (1992).

The final important observation is that a large number of researchers have used RST for the purpose of text generation (Moore and Paris, 1988; Hovy, 1989; Hovy and McCoy, 1989; Scott and Souza, 1990; Cawsey, 1990; Dilley et al., 1992; Rösner and Stede, 1992b). This testifies not only to RST's usefulness but also to the direct applicability of the results of the current study to the field of text generation. Because of the common use of RST, the unique results of the current study with respect to both the description and construction of text[2] can be more easily applied to other work in this area. In particular, the current study's focus on the precise forms of expression of rhetorical relations fills an important gap in current work.

As an example of RST description and how it was used in the current study consider the Installed-Phone text, repeated here in segmented form:

1. When the 7010 is installed
2. and the battery has charged for twelve hours,
3. move the OFF/STBY/TALK [8] Switch to STBY.
4. The 7010 is now ready to use.
5. Fully extend the base antenna [12].
6. Extend the handset antenna [1]
7. for telephone conversations.

The first problem in any such description is the segmentation of the text into spans that will serve as the atomic units of description. In RST, these spans have typically been clauses. This text, however, can be analyzed more fully if certain phrases with propositional content are considered as well. Consider the last text span "for telephone conversations." This text is expressing the fact that the action expressed in the previous clause, that of extending the handset antenna ("Extend the handset antenna"), is intended to allow the execution of the action in the current phrase, that of engaging in telephone conversations. This constitutes an expression of purpose, regardless of whether the purpose expression is a phrase or a clause. The phrase in this case is propositional by virtue of its use of the nominalization "conversations". The RST representation of this text is shown in Figure 1.1.

[2]For a discussion of the distinction between using RST as a descriptive and as a constructive tool see Mann (1987b).

This representation makes use of a set of rhetorical relations not exactly taken from the RST specification, namely purpose, precondition, result, sequence, and concurrent. These rhetorical relations are based on the procedural relations found in typical plan representations, and are discussed in more detail in Chapter 4.

This example illustrates the facility provided by RST for describing the rhetorical relations used in text. The representation is hierarchical, that is rhetorical relations are indicated between spans of text of various sizes. The procedural sequence schema at the top of the text hierarchy indicates that there are three spans of text that express a sequence of three actions. The spans themselves can be expressed as single clauses, as is the case with the second child of the sequence, or as more complex spans, as is the case with the other two spans in the sequence. This fact is traditionally displayed graphically by extending the horizontal line of a span to cover all of its subordinate spans.

The representation also includes nucleus-satellite as well as joint schemas. The joint schema used to represent the procedural sequence which contains no item that is considered superordinate or subordinate to any other. The precondition, result, and purpose relations do designate a central part, called the nucleus, and a more peripheral part, called the satellite. This latter relationship is represented graphically with a directed arrow. The satellite points to the nucleus. Further, the nucleus is the node which attaches to the parent.

The critical issue in this style of text analysis is that the rhetorical relations that are posited within the text are explicitly represented while the intentions that may have given rise to them are implicit in the relation types. This matches well with the requirements of the current study.

The grammatical features of this text are also determined and represented in the manner specified in Chapter 3. The representations for the Installed-Phone text are given in detail in Chapter 3) and will not be given in detail here.

As a brief example of this part of the representation, the record for text span 3 ("move the OFF/STBY/TALK [8] Switch to STBY.") contains a unique identifier for the clause and an indication of its syntactic form, including the fact that it has no linker, is in imperative mood, and has affirmative polarity. The argument structure is indicated as well, using pointers to the phrase records that specify the patient ("the OFF/STBY/TALK [8] Switch"). Other phrase records specify the oblique destination prepositional phrase ("to STBY"). Certain semantic features are stored as well, such as the fact that the move predicate is a material action.

The relational database representation of these rhetorical and grammatical issues provides a convenient tool for use in the analysis of the corpus. The investment of time in producing it was considerable, but benefit to the current study and to related future studies of the same text type provide ample repayment for the effort.

1.2 Corpus Analysis

The next step of the four step approach advocated in this thesis is the analysis of the corpus. In order to properly inform the text generation process, it must provide a detailed linguistic characterization of instructional text, addressing two specific issues:

1. The range of expressional forms commonly used by instructional text writers;

2. The precise communicative context in which each of these forms are used.

Given that the corpus is represented in a relational database, the first task is relatively simple. It is easy to query the database for all the purpose expressions and catalogue the various forms that are returned. This approach to determining relevant expressional forms tends to identify a much larger range of possible forms than is typically addressed by current text generation systems. On the other hand, it produces only a subset of the full list of conceivable forms. For example the current corpus contains seven different forms for purpose expressions, most of which may be in either initial or final position. This range of forms is significantly greater than the one or two forms supported in other text generation systems. It is also, however, significantly more constrained than the theoretically infinite list of conceivable purpose expressions given at the beginning of this chapter, making the task of the generation system tractable.

The second task, that of determining the functional context in which each of the forms is used is more difficult. The current study employs a hypothesis generation and test cycle, such as the one advocated by Cumming (1990), in an attempt to identify correlations between the contextual features of communicative environment on the one hand, and the lexical and grammatical form on the other.

Given that instructional text can be seen as the expression of a set of actions bearing procedural relationships with one another, this task can be broken into two sub-tasks: first, to choose, for each action expression, the rhetorical relation that best conveys its procedural relationships with other actions, and, secondly, to choose the precise grammatical form that will signal this rhetorical relation. These tasks are not necessarily strictly ordered.

Other instructional text generation systems, dealing with other issues in instruction generation, have tended to hard-code both of these choices (Dale, 1992; McKeown et al., 1990; Mellish and Evans, 1989; Reiter et al., 1992). Each action is assigned a single rhetorical status, based solely on its procedural status in the hierarchical plan structure that is input to the generator. Further, the single relation that is chosen is, in turn, mapped to a single grammatical form. This approach is at variance with the practice in actual instructional manuals, where actions are systematically expressed within any one of a number of rhetorical relations, depending upon the functional context, and, similarly, each rhetorical relation is expressed in any one of a number of grammatical forms, again, depending upon context.

This section discusses both the choice of rhetorical status and the choice of grammatical form, particularly with respect to how the corpus study has helped to determine the elements of the functional context — in the terms of systemic linguistics: ideational, textual, and interpersonal information (Halliday, 1985) — that can be used to make these choices. Each part will include a specific discussion of the leverage that the analysis brings to bear upon the Installed-Phone text.

1.2.1 Rhetorical Status Selection

The first level of choice, that of the appropriate rhetorical status for an action expression, is addressed to some degree in current instructional text generation systems such as COMET (McKeown et al., 1990), and Mellish and Evans' generator (Mellish and Evans, 1989). Both COMET and Mellish's generator are capable of producing purpose, precondition, and result expressions. They appear to have the capability to either state an action as an imperative, or to give more detail by stating the action as a purpose with a set of sub-actions, presumably when the reader is believed to need more detail. This is an example of using the functional context to determine the rhetorical status of an action expression. These systems are, however, not so flexible when expressing preconditions and results. In both cases, the rhetorical status of preconditions and results is explicitly represented in the input to the system (in the form of a hierarchical plan with preconditions and results explicitly marked). The systems would never, for example, decide to express an action as a precondition or vice-versa. This, however, does occur in real instructional texts. Consider the following sentence, as produced by IMAGENE:

(12) *Return the OFF/STBY/TALK switch to the STBY position after your call.*

Here, IMAGENE has *rhetorically demoted* the action of making a call from the standard imperative clause form to a dependent precondition form ("after your call"). IMAGENE did this because it was told that the reader was assumed to know about the process of making a call. This flexibility allows IMAGENE to express actions in whatever rhetorical status is appropriate for the functional context.[3]

In the Installed-Phone text, there are a number of issues that are representative of the value of the analysis as a whole. This text, as seen in the previous section, includes seven action expressions: two preconditions, one result, one purpose, and three sequential actions. Each of these choices for rhetorical status will now be discussed.

The two preconditions could very easily be expressed as imperatives ("Install the phone" and "Charge the battery"). One observation, however, is that these commands have already been given in

[3]Outside of the instructional domain, systems designed to be sensitive to the reader's knowledge (Paris, 1988) or to plan texts based on writer and reader intentions (Moore and Paris, 1989; Hovy, 1989) also allow variance of rhetorical structure.

the manual, previous to the current expressions. On further investigation, this status of being previously mentioned correlates well with precondition status; of the 699 actions expressions in the corpus, 21 are previously mentioned and appear as preconditions in the various texts.

The single result expression ("The 7010 is now ready to use") is an unusual case, not being based on an action at all, but rather being the expression of the state of readiness of the phone for normal operation. Although this *Ready-To-Go-On* condition expression did not occur frequently in the corpus (there were 3 instances of it), facility was added to IMAGENE to express it because of its interesting properties.

The purpose expression in the final sentence ("for phone conversations") is an example of a non-clausal, but nonetheless propositional, expression. Most action expressions are clauses, but there are a number of cases where such phrases are used to express actions, particularly purposes. This issue will be discussed in more detail in the next section. The key observation for purposes is that they always express some high-level action which has specifically expressed sub-actions in the text, in this case the sub-action of returning to the seat ("Return to seat"). All of the 119 purpose expressions in the corpus express some form of high-level action.

The three sequential actions are expressed, as one might expect, in temporal order. This is, however, not universally true. There are a number of specific contexts in which actions are not expressed in temporal order, having to do with textual and interpersonal issues. The precondition expressed in example (12) ("after your call") is an example of such a case. For rhetorical reasons, as discussed above, this actions is expressed out of temporal sequence. Other situations where this phenomenon occurs are discussed Chapters 5 and 6.

1.2.2 Grammatical Form Selection

The second level of choice, that of the appropriate grammatical form for an action expression, is less commonly addressed in current generation systems either for instructional text or for other forms of text. At the lexical level, choosing between the surprising number of lexical items and phrasal forms that convey basically the same meaning is an open and active area of research (Cumming, 1988; Reiter, 1990; Wanner and Bateman, 1990; Nogier and Zock, 1991). Unfortunately, this level of interest in the subtle differences between forms of expression does not extend to the rhetorical level. Multi-sentence generation systems have tended to map each rhetorical relation that they are capable of dealing with to a single grammatical form of expression. All of the rhetorical information is there, but the constant repetition of the same form is not only annoying, but difficult to understand. Human produced text is characterized by the variety of forms that are used to express each rhetorical relation and how these forms are systematically used.

For example, in the text generators mentioned in the previous section, purposes are typically expressed using one or two forms. Mellish's system chooses "involves" expressions, as in "Doing X, involves doing Y, doing Z, ...". COMET chooses fronted imperatives followed by a colon as in "Do X: First, do Y, Next do Z. ...." or sometimes the fronted "in order to" infinitive as in "In order to do X, do Y. ....". These forms are sufficient to disambiguate the procedural input to the systems, but do not adequately represent the forms used in real instructional text on a regular basis. None of them, in fact, are produced by IMAGENE because they did not appear in the corpus at all.

The choice of grammatical form for the Installed-Phone text represents a more diverse set of possible forms than found in these other text generators. The discussion of IMAGENE's prescriptions for this example will be deferred to the next section because that section presents the IMAGENE's actual output.

1.3 IMAGENE

The third step of the four step approach advocated in this thesis is the implementation of the results of the corpus study in a computational architecture. This step forces the researcher to encode the results of the corpus study in quantifiable terms. The architecture designed for IMAGENE is shown in Figure 1.2.

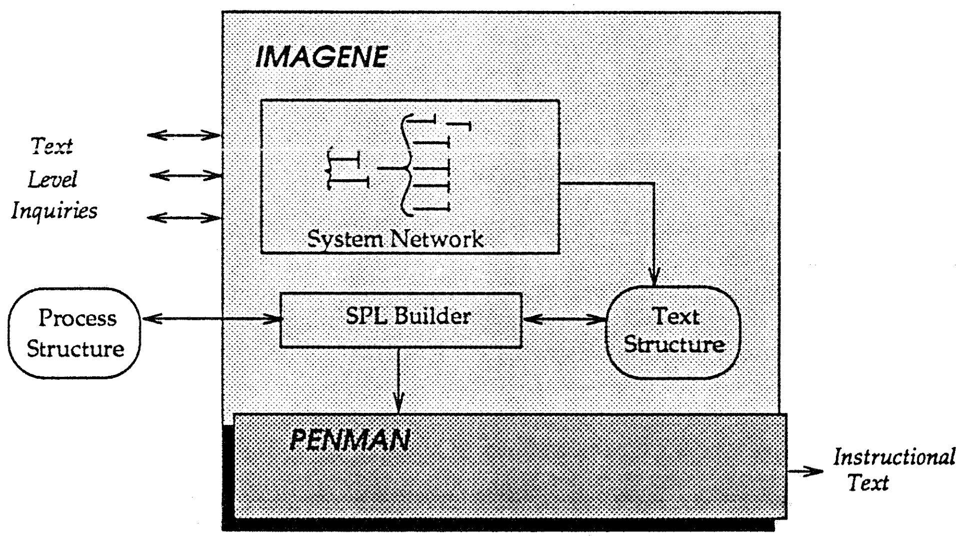

Figure 1.2: The Architecture of IMAGENE.

This section will discuss the motivation for using a pre-existing sentence realization system and the specific features of Penman that made it the logical choice for the current project. It then discusses the use of the System-Network formalism for encoding the results of the corpus study. Finally, it presents an annotated run of IMAGENE for the Installed-Phone text.

1.3.1 Penman

As discussed above, the fundamental generation problem for the current study is that of choosing the rhetorical and grammatical forms that best express procedural relationships. The solution to this problem requires a full coverage sentence-level text generator, the construction of which would be beyond the scope of the current study. Thus, Penman, a flexible, full-coverage sentence-level text generator developed at the USC Information Sciences Institute, was chosen for use in IMAGENE (Mann, 1985; Penman, 1989; Mann and Matthiessen, 1985; Matthiessen, 1985; Matthiessen, 1988). Penman was originally attractive because of its broad coverage of English syntax, probably the most comprehensive of any readily available text generator. Although this full coverage is the most fundamental contribution of Penman to the current study, there were a number of other features of the Penman environment that made it particularly attractive for the current project:

- In addition to broad coverage, specific utilities are provided for the addition of the relevant lexical items and domain concepts. These proved useful in building a lexicon and domain model for the instructional domain. It also proved possible to add the relatively few grammatical forms required for instructional text that were not part of the original distribution;

- It is based on a Systemic-Functional view of language (Halliday, 1976). This view will be discussed in some detail in Chapter 2, but for now it suffices to say that the approach is distinctly functional, that is, it is particularly interested in using contextual features to map communicative goals to acceptable grammatical forms.

- An extremely useful by-product of this view of language, is that it contains a well-developed implementation of the *System Network*, a tool for representing grammar. IMAGENE reused this implementation to produce text structures, in a manner similar to the way Penman already produces sentences.

The architecture for IMAGENE is an outgrowth of the Penman architecture. It implements a system network for the construction of instructional text structures that match those found in the corpus analysis. The precise nature of the system network formalism and how it has been used in IMAGENE will be the subject of later chapters.

1.3.2 The System-Network Formalism

IMAGENE uses the system network, as implemented in Penman, as a single formalism to address both the issue of rhetorical status selection and that of grammatical form selection. The system network has proven useful in text generation because of its concern with mapping ideational, textual and interpersonal issues onto grammatical forms. IMAGENE uses this formalism to allow functional considerations to determine the appropriate rhetorical structure for a set of actions and then to determine the appropriate grammatical form for each of the involved rhetorical relations.

There are two fundamental inputs to IMAGENE: (1) the specification of a hierarchical procedural plan which will serve as the basis for what needs to be prescribed to the reader in text, called the *Process Structure*, and (2) the features of the functional context that affect the form of expression of the plan, called the *text-level inquiries*. IMAGENE's high-level architecture, including these two inputs, is shown in Figure 1.2.

The *Process Structure* is represented using IMAGENE's Process Representation Language (PRL). This representation is like that produced by a procedural planner, but is currently built by hand. This simplification has allowed the current study to focus on the problem of expression rather than planning. The PRL is represented in LOOM (LOOM, 1991), a descendent of KL-ONE (Brachman, 1979; Brachman and Schmolze, 1985).

The second input to IMAGENE is a set of manual responses to the *text-level inquiries*, analogous to Penman's sentence-level inquiries. Penman uses the inquiry interface to explicitly specify the effect of the elements of the functional context on the form of the sentence being generated. Each Penman inquiry asks a single question concerning the functional context in which the sentence is being expressed, and then either constructs or modifies various aspects of the form of the sentence based on this context. These inquiries can either be answered manually, or automatically. IMAGENE works in an analogous way on rhetorical and grammatical structures for instructional text.

Currently, the data structures and code necessary to respond to IMAGENE's inquiries automatically have not been implemented. Rather, the inquiries are answered manually, allowing the current study to focus on determining the appropriate set of inquiries and the precise rhetorical and grammatical consequences of the results of these inquiries. The responses to these inquiries are used by IMAGENE to construct an appropriate structure for the text, called the *Text Structure*. The Text Structure is represented in IMAGENE's Text Representation Language (TRL), which is also implemented in LOOM.

The fact that these inquiries are not implemented raises an important issue: the inquiries must be expressed in terms that are, in principle, implementable. For example, the level of reader knowledge concerning a particular action is important in determining rhetorical status. Determining this level of knowledge is an extremely difficult task, and is the subject of a number of studies in text generation, particularly in the context of interactive discourse (Cawsey, 1990; Moore and Paris, 1988; Paris, 1987; Paris, 1988; van Beek and Cohen, 1990). The current study does not specifically address this task in order to more fully address the equally interesting issue of managing diversity of expression. A serious attempt was made in the current study to specify inquiries whose answers are, in principle, derivable from the context of generation. Only further work on implementing the inquiries, however, will show if this effort has been effective.

After IMAGENE has built the Text Structure, the SPL Builder translates it into the corresponding Sentence Planning Language (SPL) command to be passed to Penman for surface realization. Separate commands are produced for each sentence in the text structure and are passed to Penman individually, thus using Penman as a sentence-level text realization component. The nature of these plans is discussed more fully in Chapter 4.

As a review then, the flow of data in IMAGENE, as shown in Figure 1.3, starts with the Process Structure. IMAGENE uses the system network, which inquires about elements of the Process Structure

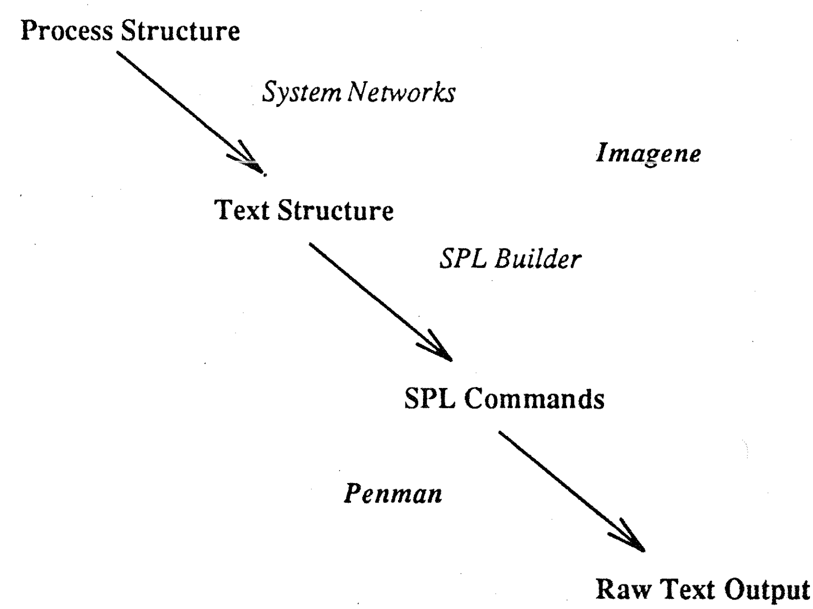

Figure 1.3: Data-flow in IMAGENE.

and other aspects of the communicative context, to produce the Text Structure. The SPL builder then translates the Text Structure into the sequence of SPL commands which result in the raw text output.

1.3.3 The Installed-Phone Example

As an example of the data structures used in IMAGENE's operation, consider the PRL representation of the actions from the Installed-Phone text, depicted graphically in Figure 1.4. Note that the PRL structure specifies most of the lexical choice issues so IMAGENE may concentrate on the rhetorical level. Given this structure, representing a sequence of five actions, the system networks use the contextual distinctions discussed in the previous section and others to produce the TRL structure for the example, shown in figure 1.5.

TRL allows the Text Structure to include a representation of the hierarchical structure of the text in terms of RST, including both nucleus-satellite and joint schemas. In addition, TRL specifies the textual order and clause combining using additional *New-Sentence* and *Continue-Sentence* links. For example, the **Install**, **Charge**, and **Move** nodes are all combined into one sentence in Figure 1.5. Finally, TRL specifies the grammatical form of each action expression using three features which may be attached to expressible nodes in the structure. The **Form** feature specifies the general grammatical form. For example, the **Install** is marked as **Passive** to indicate that the agentless passive should be used. The **Linker** and **Tense** markers are also used to mark the appropriate linker and tense of the expression.

The SPL Builder, shown in Figure 1.2, then translates the TRL structure into a sequence of Sentence Planning Language (SPL) commands, one for each sentence, and passes them to Penman. Note that at this point in the generation process, there is no question of whether Penman can generate the specified sentence because the system networks have been careful not to specify any grammatical form or lexical item that is not available to the generator. Neither is there any question of the system failing to consider any lexico-grammatical form that might have been more expressive in the context of instructions. This is how IMAGENE addresses the problems of "expressibility" and "expressiveness" as articulated by Meteer (1991). IMAGENE's system networks are designed to specify only those lexical and grammatical forms that are available to Penman, and the forms that are specified represent the full measure of expressiveness consistently displayed in real instructional text as determined in the corpus study. The

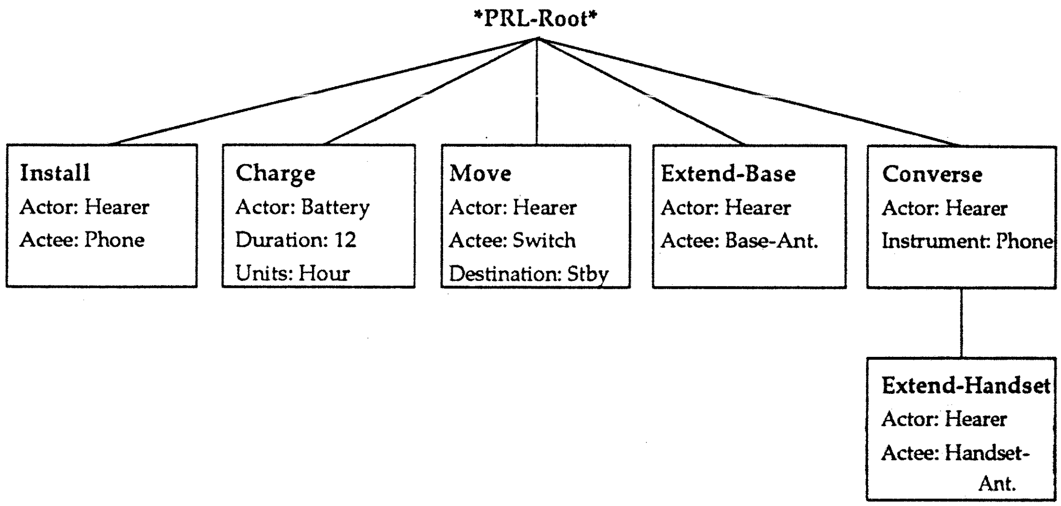

Figure 1.4: The Process Structure for the Installed-Phone Text

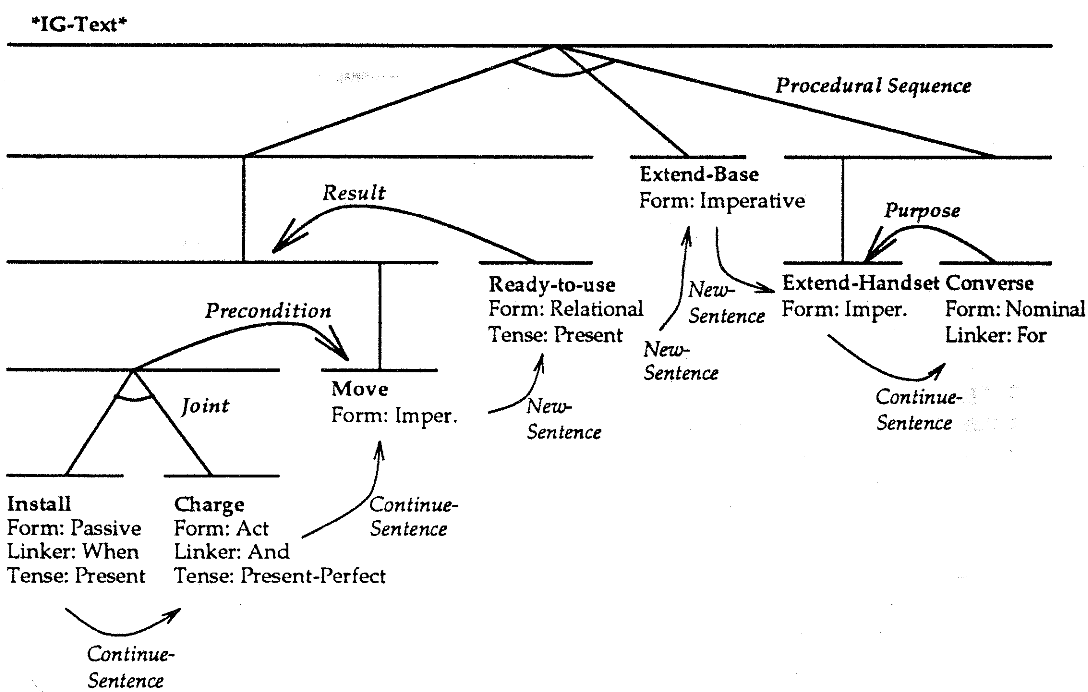

Figure 1.5: The Final Text Structure for the Installed-Phone Text

generated text for this example is shown here:

(13) *When the phone is installed, and the battery is charged, move the OFF/STBY/TALK switch to the STBY position. The phone is now ready to use. Extend the base antenna. Extend the handset antenna for phone conversation.*

The resulting text is very similar to the text in the original, particularly at the rhetorical level, but there are some interesting differences. In the original text, the action of charging the battery ("and the battery has charged for 12 hours") is expressed as a present-perfect tense, while the simple passive was used in IMAGENE's output ("and the battery is charged"). This was done because the perfective form was simply not used very frequently in the expression of preconditions in the corpus. One last difference at the rhetorical level is the use of the singular form for the nominalization "conversation" in the last sentence. IMAGENE does not produce plural nominal forms, again, because they were not commonly used in the corpus.

At the lexical and phrasal level there are a number of differences, including the lexical items chosen for the object references and the use of determiners. These differences arise from the fact that the current study has not specifically addressed them. Currently, IMAGENE uses simple algorithms for pronominalization and the use of determiners, which are not based on a detailed corpus study of the forms and functions of the object reference domain. A study of referring expressions, similar to the current study's work on rhetorical expression, would allow the development of a more principled solution to this problem.

1.4 Verifying the Output

The final step of the four step approach advocated in this thesis is the comparison of the output of the text generator with the text in the corpus. This is a critical step because it forces the researcher to address the phenomena that actually exist in the corpus. It is very easy to lose sight of this goal when building the text generator.

The process of comparing IMAGENE's prescriptions with the actual text in the corpus involves running IMAGENE's system networks on all or a subset of the examples in the corpus. In the case of the Installed-Phone text, IMAGENE makes specific predictions concerning the form of each of the seven action expressions. The previous section discussed IMAGENE's output and discussed where it was correct and where it failed, but that was largely an anecdotal discussion. Such a discussion is certainly a step in the right direction, provided the real corpus text is consulted (as was the case in that discussion), but makes it difficult to draw any quantitative conclusions. A more quantitative verification process is needed.

In the current study, IMAGENE's system network was re-run for every action expression in the corpus, both those from the training set and those from the testing set, and statistics were kept on the accuracy of its specifications in the TRL structure. The correctness was judged on four separate lexical and grammatical issues: Linker, Form, Slot, and Clause Combining. The resulting TRL structure had to specify the correct linker (either preposition or conjunction), form (tense and aspect), slot (textual order), and combining (whether the expression was combined with the following sentence or not). These measures are discussed in more detail in Chapter 7.

The results for the Installed-Phone text are shown in figure 1.6. The function `Compute-Stats` takes a list of examples (in this case the identification codes of the seven expressions in the Installed-Phone text) and looks in the collected results of the test runs to determine the percentage of correct predictions for each of the four lexical and grammatical issues. In this case, the example was handled entirely correctly except for the form of the second precondition expression ("and the battery has charged for twelve hours"). In this example, the linker ("and"), the slot (standard temporal order), and the clause combining (combined in the same sentence with the following expression) are all correct. The form, however, is not correct. IMAGENE predicts an agentless passive, whereas the corpus uses the present-perfect tense.

This accuracy leads to an 85.71% accuracy rate for the full example, that is, every lexical and grammatical issue of concern the current study is predicted correctly for 85.71% of the seven action

```
? (compute-stats '(phl-cl3 phl-cl4 phl-cl5 phl-cl6 phl-cl7
                    phl-cl8 phl-p8))

Choice Correctness Statistics:

Linker: 100.0%  total occurrences: 7

Form: 85.71%  total occurrences: 7

Slot: 100.0%  total occurrences: 7

Combining: 100.0%  total occurrences: 7

Combined: 85.71%
(7 100 600/7 100 100)
?
```

Figure 1.6: Sample Run of the Statistics Collection Routines

expressions in the Installed-Phone text. It must be noted that the one incorrect prediction, on the second clause ("and the battery has charged for twelve hours") is just that, a failed prediction, and definitely not a complete failure. A readable text is actually produced by IMAGENE; it is just not exactly what is found in the corpus.

The current study has been based primarily on an analysis of a small subset of the full corpus, namely on the instructions for a set of three cordless telephone manuals. This training set constitutes approximately 35.32% of the corpus. The results of this analysis were then implemented in IMAGENE and applied to the full corpus, providing a detailed characterization of the instructions found in the original telephone manuals, and a quantitative analysis of how well this characterization applies to the other forms of instructions in the corpus.[4]

Figure 1.7 shows the relative accuracy of IMAGENE's predictions for all of the source texts in the corpus. As can be seen, the results for the text in the training set, i.e., the Airfone text and the other phone manuals, are the best overall.[5] The results for the text in the testing set, i.e., the other electronic devices, the manipulative processes, and the creative processes, are also fairly high, around 50%. Thus, the model of instructional text coded in IMAGENE can be applied with some accuracy to other types of instructions besides phone manuals.

1.5 Overview

This introductory chapter has discussed the over-arching goals and methodologies adopted in this study. A note is appropriate here concerning the power and specific focus of the approach. As has been discussed, the current study addresses the problem of determining the precise rhetorical and grammatical forms that are most effective for expressing actions in an instructional context. The corpus-based methodology employed here is well suited for this problem, providing both a principled means for cataloging the rhetorical and grammatical forms that are consistently used in instructional text, and an environment for testing and confirming hypotheses concerning the contextual issues that determine the choice among these forms. The approach is also well suited for the study of other issues

[4]There are a couple of exceptions to this mode of operation which are discussed in Chapter 7.

[5]The results for the non-procedural (or *primitive*) portions of the other phone manuals are not as good because the current study has focussed primarily on highly procedural text. These non-procedural portions of text were included in the other phone manual texts in a pilot study done before the focus on procedural text had been determined. The remainder of the corpus, i.e., all the text besides the other phone manuals text, was collected exclusively from the procedural sections of the respective sources.

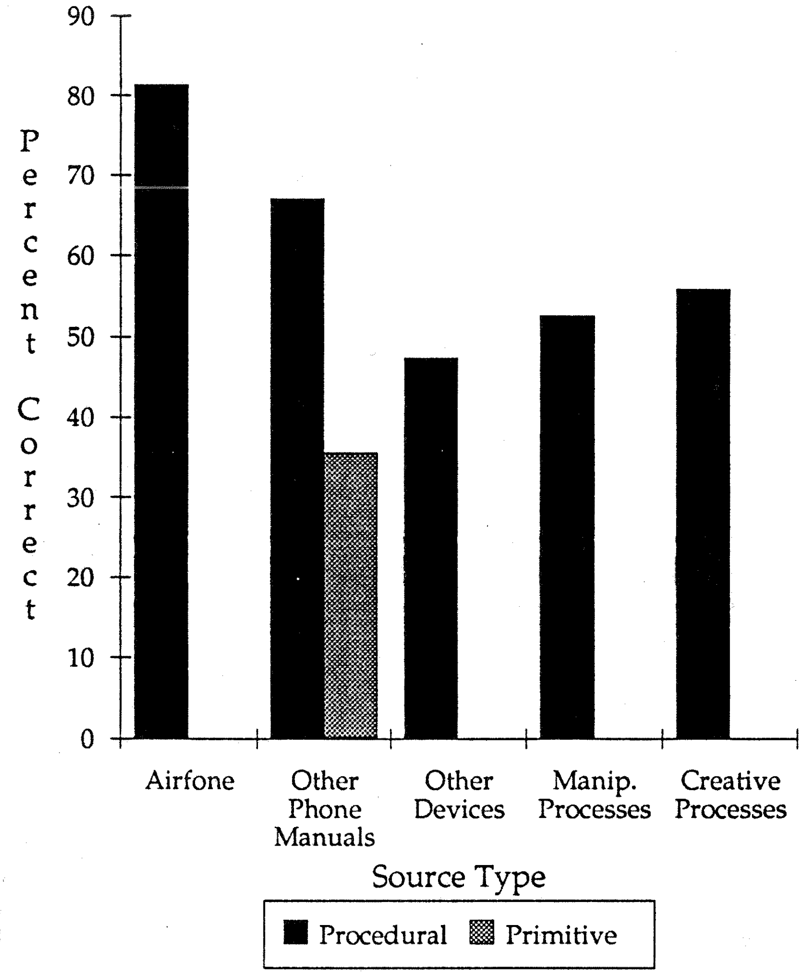

Figure 1.7: The Relative Accuracy of IMAGENE Predictions for the Various Source Texts

not addressed in the current study, including the macro-structure of instructional text and referring expressions.

The issues of procedural planning, user modeling, and content selection, although of unquestionable importance to the broad goal of generating instructions, are not directly accessible to the corpus-based methodology and, thus, will not be specifically addressed here. The current study will make a number of prescriptions for the type of information that these sources should provide to the text planner pursuant to the generation of effective text, but will say nothing about how they should be implemented in order to achieve this.

There are three fundamental contributions of the current study. First and foremost is the proposal and demonstration of a four step approach for managing diversity of expression at the textual level. Concern with this issue is absent in many current studies in Computational Linguistics, particularly those related to instructional text. The approach involves collecting a suitable corpus of text, analyzing that text, implementing the results of the analysis in a text generator, and verifying the output of the generator.

The second contribution is analysis of instructional text itself. There is an increasing interest in instructions and the possibility of automating their production. The results of this study contribute an important analysis of some of the rhetorical and grammatical characteristics of instructional text.

The final contribution is the IMAGENE program. Given the broad use of the Penman generation system, it would be possible to distribute IMAGENE as part of the general Penman environment for those interested in directly using the implementation.

This thesis will begin by reviewing the current work on generation and instructional text in Cognitive Science, then detail the four step process, and conclude with an assessment of results of the current study.

Chapter 2 will review the related work from the fields of Linguistics, Psychology, and Artificial Intelligence. Linguistics has provided much of the theoretical foundation for the current study's characterization of instructional text. It has also contributed to the mechanisms used in the implementation of IMAGENE. Section 2.1 will review these contributions. Section 2.2 will discuss Psychology's interest in improving instructional text and general instructional material. It will then specifically delineate the current study's potential contribution to this work. Finally, Section 2.3 will discuss work in Artificial Intelligence on understanding and generating instructional text, delineating the uniqueness of the current study with respect to this work.

Chapter 3 will detail the corpus study, discussing the text that was included in the corpus and the criteria that were used for its inclusion. It will also explicate how this text was represented in a relational database and give a detailed example of how the database was used to analyze a particular aspect of purpose expressions.

Chapter 4 will provide a detailed discussion of the implementation of IMAGENE. This information will be critical in understanding the discussion of IMAGENE's system networks, given in Chapters 5 and 6. Chapter 5 will specifically address IMAGENE's mechanisms for choosing rhetorical status and Chapter 6 will address its mechanisms for choosing grammatical form. As the functional distinctions that the current study has determined to be important in instructional text generation will be spread throughout these latter two chapters and thus difficult to list, a complete list of the features is given in Appendix D.

The critical foundation for the validity of the entire study is the assessment of how accurately the implemented results match the form of the text in the corpus. This assessment is given in Chapter 7. IMAGENE's predictions concerning the rhetorical and grammatical form of instructional text are matched with the forms found in the corpus. The chapter concludes with a discussion of implications of the current study for the definition of the characteristics of the instructional register.

Chapter 8 reviews the contributions of the current study, concluding with an outline of the prospects for continued work in this area.

Chapter 2

Related Research

Instructional text has been a subject of interest in many of the fields of Cognitive Science, including the fields of Linguistics, Psychology, and Artificial Intelligence. This chapter reviews the work in these fields and assesses its relevance to the current study. The general goal of Linguistics has been to delineate the linguistic characteristics of instructional text, that of Psychology to provide formalisms that aid in the its improvement, and that of Artificial Intelligence to provide mechanisms for its processing.

2.1 Characterizing Instructional Text

This section reviews the relevant work on instructional text within the Linguistics community, specifying where necessary the contribution that the current study makes to the field. There has, unfortunately, been little if any interest in the linguistic features of instructional text. On rare occasions, linguists have used procedural text as a source of issues for study; unfortunately, these studies do not attempt to propose a characterization of the instructional register that would be useful to a text generation project. The fundamental contribution of Linguistics research to the current study has been in the areas of formalism and methodology. Following a brief discussion of the contributions of the Generative-Transformational tradition in Linguistics, this section will discuss the contributions to the current study of the Systemic-Functional and Discourse-Functional schools.

The mass of linguistic literature stemming from formal theories of language, such as those of Noam Chomsky (Chomsky, 1957; Chomsky, 1965; Chomsky, 1981), bear upon the problems of linguistic clause and sentence structure. This work has contributed greatly to the problems of identifying the full range of syntactic forms that are considered "grammatical" and has been most useful in the construction of sentence understanding and generation grammars. Some studies of particular interest have concerned various syntactic constructions that occur frequently in instructional text such as purposes (Bach, 1982; Huettner, 1989; Jones, 1991; Jackendoff, 1990).

The problem with these studies is their even-handed treatment of the full range of "grammatical" possibilities of expression. Choosing among the diverse forms of expression in this range, all of which are "grammatical," is the fundamental problem in text generation. For example, Brown's work on tailoring Government Binding theory to the task of language generation (Brown et al., 1986a; Brown et al., 1986b; Brown, 1987) required significant changes to the "pure" theory in order to address the problem of choice in generation. As he and his colleagues report (Brown et al., 1986a, page 2–3):

> There are two main differences between "pure" GB theory and the use we have made of it. First, GB deals only with strictly linguistic aspects of language (ignoring, for example, real-world context, motivation, etc.) Thus, while it generates sentences, there is no control over *which* sentence will be generated at any given time. On the other hand, a sentence generated by our system is determined by the use of Conceptual Graphs (CGs) which represent certain aspects of the meaning of the sentence to be generated.
>
> A second major difference is that in GB Theory filters are often used to rule out un-grammatical sentences once the sentences are generated, whereas in our approach filters

are transformed into active constraints that prevent ungrammatical sentences from being generated.

As with this work in GB theory, formal linguistics in general has provided few tools which a text generation system could deploy to deal with the problem of choice. The type of analysis that is needed is one where the elements of the communicative context that help determine the choice of form are delineated. These features, including the interpersonal and textual contexts, could then be used in the text generator to choose the appropriate form from among the grammatical alternatives.

There have been a number of alternative approaches to linguistic analysis that are much more useful to the current study, namely in the areas broadly identified as *Functional*. These fields are deeply interested in precisely the same issue of choice that motivate text generation researchers, namely the identification of the contextual factors that motivate linguistic choice. The following sections will discuss two of these fields, Systemic-Functional Linguistics and Discourse-Functional Linguistics, indicating the contributions each has made to the current project.

2.1.1 Systemic-Functional Linguistics

Systemic-Functional Linguistics is based on the work of Michael Halliday, starting with the work on scale and category grammars (Halliday, 1961) and continuing on to systemic-functional grammar (Halliday, 1976; Halliday, 1985). The primary goal of this work has been to identify the features of the "functional" environment that motivate language use. Its roots are in anthropology and sociology (Firth, 1957; Malinowski, 1923; Hjelmslev, 1943) rather than in mathematics and logic (as they were for Generative-Transformational Linguistics).

The fundamental argument of the Systemic Linguists is that language is primarily a tool for expressing meanings in context. Thus, the primary goal of a grammar is to map from meanings or functions to forms. The task of specifying structural forms is only a secondary part of this process. Systemic linguists divide function into three types, called metafunctions: (1) Ideational, (2) Interpersonal, and (3) Textual.

The *Ideational* metafunction represents the logical and experiential information we have about the world. In this sense it is closely related to the traditional definition of the term *semantics*. The *Interpersonal* metafunction represents the relevant factors concerning how people interact in the world using language, including issues such as formality, power, and familiarity. This is closely related to the study of Socio-Linguistics. The *Textual* metafunction represents flow of information in the text, including the traditional notions of theme and rheme (Firbas, 1966). This is closely related to discourse analysis.

This work has been critically important to the IMAGENE project not just because of its interest in the questions of language in use, but more concretely in its contribution of the *system network* formalism. This formalism has been extensively used in the current study and will be presented in some detail in Chapter 4. The fundamental point is that it is a formalism that allows the mapping of features of the functional context, i.e., the ideational, interpersonal, and textual context, directly onto grammatical forms. The system network has been used extensively in computational linguistics for sentence level generation (Mann, 1985; Winograd, 1972; Davey, 1978; Patten, 1988), resulting in several fully implemented sentence generators. It has also been used, although less commonly, at the text or discourse level (Berry, 1981a; Berry, 1981b; Hovy et al., 1992; Fawcett, 1990; Fawcett et al., 1988; Halliday, 1984; Butler, 1985; Hasan, 1978).

One area of considerable interest in the systemic community is register (Halliday, 1978; Halliday and Hasan, 1985; Gregory, 1967; Ure and Ellis, 1974). They define register as the variety of language distinguished according to use, the determining characteristics of which are termed field, tenor, and mode. *Field* pertains to the relevant objects and purposes of the domain, *Tenor* to the addresser/addressee relationship, and *Mode* to the medium. These three determining characteristics closely mirror the tripartite metafunctional division.

The basic problem addressed in the current study, the characterization of language used to instruct, can therefore be seen as an attempt to characterize the instructional register. The current work on register in Systemics, unfortunately, has contributed little to the detailed understanding of the instructional register. For one exception to this, see Young (1985), which includes a short description of some of the characteristics of the field, tenor, and mode of the instructional register and how they affect the form. This work provided broad characterizations such as the fact that imperatives are common and that moodless clauses are used as headings. The work was, however, too general to be of direct use in IMAGENE. The various characteristics were listed, but no principled means of using them in a generator were given.

There are also computational linguistics approaches to register that are strongly motivated by this systemic work (Bateman and Paris, 1989; Paris and Bateman, 1990; Bateman and Paris, 1990), but these as well are concerned with issues that are not specific to instructional text. They have, so far, been largely concerned with the development of an architecture for the investigation of register, rather than on performing a detailed register analysis itself. IMAGENE can be seen as providing the detailed characteristics of the instructional register at the rhetorical level that could be implemented in their architecture.

2.1.2 Discourse-Functional Linguistics

The theoretical framework for linguistic analysis performed in the current study has been taken directly from Discourse-Functional Linguistics.[1] This group of linguists is united primarily by a concern with how cognitive and social pressures shape the range of grammatical phenomena in the world's languages. This concern manifests itself in explanatory accounts of why the grammatical phenomena exist at all, and descriptive accounts of how the grammatical phenomena are used in context. The current study is primarily a descriptive account of how actions are expressed in context. What distinguishes the Discourse-Functionalist school from the Systemic-Functional school is its goal of describing the textual metafunction in terms of the other two metafunctions, the ideational and the interpersonal. There are two fundamental contributions of this work to the current study:

- The focus on naturally occurring text;
- The use of quantitative measures;

Discourse-Functionalists are profoundly interested in using quantitative measures of naturally occurring text as a source of insight into language. No contrived sentences are used in the analysis. This concern with "real" text has dictated that most of the recent studies in this area be based on analyses of corpora of various types. One example of this method is the collection of studies on the "pear" stories, a set of elicited narratives describing the events in a film (Chafe, 1980). Although these narratives were not taken from a totally natural context, the analyses do tend to display a quantitative methodology (see for example, Du Bois' 1980 study linking the trace of identity with the use of articles.)

Other phenomena that have been studied in this school include adverbials in conversation and text (Chafe, 1984; Ford, 1988), dative shift (Thompson, 1988), anaphora (Fox, 1987, see also the introduction to Givón, 1983), relative clauses (Fox and Thompson, 1990), and clause combining (Cumming, 1984). The attraction of the corpus-based approach for the current study is its ability to uncover the full variety of expressional forms that are used in instructional text, not relying on introspective evidence.

Further, these studies have used corpora of various sorts to correlate meanings or functions in discourse with specific, quantitative measures of the text. Givón (1983), for example, uses a set of three measures of topic continuity, called *referential distance*, *potential interference*, and *persistence*, all of which are explicitly defined in terms of quantitative measures of the text. Referential distance, for instance, is defined as the number of clauses from the current referring expression to the most recent reference to the same referent. Although even measures as quantitative as these can be difficult to apply in practice, they do provide a more solid ground for linguistic analysis, not based on more difficult to apply intuitive measures such as "topicality" or "empathy." This is their primary attraction for the current study.

Given that the methodology is to look at quantitative measures of real text, it is not surprising that this type of analysis tends to result in non-exhaustive accounts of the phenomenon in question. The results are statistical in nature, that is they attempt to explain as large a percentage of the corpus as

[1]No definitive reference for this school of Linguistics exists. Some of the major researchers and publications include Givón (1979; 1983; 1989), Thompson (1985; 1987; 1988), and Chafe (1980).

possible. Certainly the goal is to account for every expression, but it must be realized that the features of the functional context are quite vast and often inaccessible to the analyst.

There have been several studies from the Discourse-Functional school have been particularly important for the analysis of instructional text, including Thompson's work on purpose clauses and general subordination in narratives and Cumming's work on nominalizations. They will be discussed in the following sections.

2.1.2.1 Work on subordination

Thompson's analysis of purpose clauses in English (1985) has been particularly helpful in the current study, although, as will be seen in later chapters, her study did not specifically address instructional text. The results of the current study, nonetheless, provide a confirmation of her predictions in this domain.

In a more general study (1987), Thompson addresses the distinction between: ".... material whose primary function is to report the narrative EVENTS of the story, those which are temporally ordered with respect to each other, and material whose primary function is something other than this."

This distinction is based on one originally proposed by Labov (Labov and Waletzky, 1967; Labov, 1972). The general notion is that the former type of events (the narrative events) are to be expressed as main clauses, while the latter type are expressed as subordinates of various types. Thompson proposes and defends a two-part thesis: "a. The vast majority of subordinate clause predicates will not be on the time line. b. Those which are on the time line are doing other discourse work in addition to naming a temporally sequence event."

She goes on to identify various special functional contexts in which narrative events are expressed as subordinate clauses, such as in recapitulations and advance summaries. This closely parallels the current study's analysis of rhetorically demoted action expressions. *Rhetorical demotion*, to be discussed in detail later, is the process by which *imperative events* (the parallel of narrative events in instructional text) are expressed as subordinate clauses of various types.

2.1.2.2 Work on nominalizations

A study by Cumming (1991) concerning nominalizations in various types of text including instructional text, has implications for the current study. Cumming provided a workable definition of nominalizations which was useful in the analysis of the instructional text in the corpus. It included various criteria for identifying a nominalization, such as the existence of a morphologically related verb for the NP, and the use of time expressions. These criteria were useful in the analysis of the corpus for the current study.

Her study, further, explored the implications of the nature of nominalizations for the architecture of text generation systems. She proposed that a generation system should distinguish at least two levels of representation, one which allows the system to reason about events, regardless of their expression as clauses or nominalizations, and another which differentiates between the two. IMAGENE, as will be detailed in a later chapter, contains both levels, the former is the level at which the actions in the plan are considered for expression, primarily the representation of the plan (using a plan representation language). This level determines the nature of the action, the relation the action has to other expressed actions, and the knowledge that the reader has about it, leading to the choice of the appropriate rhetorical status for its expression. The latter is the representation of the rhetorical and grammatical structure of the text (using a text representation language). This representation is required to deal with the grammatical consequences of, in this case, the expression of the propositional arguments of the clause in nominalized forms.

2.1.3 IMAGENE

The fundamental contributions of these fields of Linguistics to the current study are the system network formalism provided by Systemic-Functional Linguistics and the theoretical foundations and quantitative methodology for linguistic analysis provided by Discourse-Functional Linguistics.

IMAGENE's contribution to this linguistic work is the implementation of the results of the analysis. These linguistic studies, to varying degrees, reflect a lack of the specificity required for computational implementations. The current study also extends many of the analyses into the domain of instructional text, an important task given that the variation of language between registers is so pronounced (Biber, 1988).

2.2 Improving Instructional Text

This section reviews the relevant work on instructions within the Educational and Psychological communities, specifying where necessary the contribution that the current study makes to the field. There has been a considerable amount of interest in these communities in the improvement of "instructional" text. The vast majority of this work, however, has either directly concerned general instructional material, or has lumped instructional text and general instructional material together. Because this work is nonetheless related to the current study, this section will review it, showing what it has contributed to the current study and conversely what the current study can contribute to it.

2.2.1 General Writing Heuristics

Textbooks on general writing style (e.g., Strunk Jr. and White, 1979; Zinsser, 1985) have very little to say concerning the precise choice of rhetorical and grammatical form. They typically contain a list of general heuristics such as Strunk and White's "Omit needless words" and "Do not overwrite." They are based largely on introspective evidence concerning "good" writing.

These introspective studies exhibit the same problems that most of the introspective studies of the late 1800's (see Gardner 1987 for a discussion of introspectionism) and the current studies of expert behavior exhibit, that is that the writers really don't know what they are doing. The lists of heuristics produced by such efforts are ad hoc at best, relating long lists of uncorrelated rules of thumb that no doubt have some validity and practical application, but are difficult to use in specific contexts (Britton, 1986; Hartley et al., 1980; Klare, 1963). Klare (1963) found very little agreement among writing manuals concerning the specific set of heuristics to include and how they should be worded.

The same observations are true of writing manuals directed specifically at technical writing or even instructional text writing (Gopen and Swan, 1990; Weiss, 1988; Weiss, 1991; Holtz, 1988). Not only is there little agreement among writing manuals concerning the prescriptions, but the prescriptions themselves are quite vague. Commands like "Adapt style to reader" and "Use concrete language" are typical. Because of the vagueness of the prescriptions given in these lists of writing heuristics, they are basically unusable in an automated text generation system, which requires detailed prescriptive algorithms for making rhetorical and grammatical choices in instructional text.

The fundamental problem with producing lists such as these is that language is so diverse. How is one to produce a set of heuristics, applicable to a wide range of communicative purposes, that are both simple and specific? This difficulty has led some to propose that only an officially sanctioned subset of English be used for technical documentation (AECMA, 1988) or perhaps that legislation should be adopted requiring the use of "plain language" (Steinberg, 1991). This would reduce the amount of complexity that must be dealt with at language production time. These approaches have produced much more specific prescriptions for generation than the general writing heuristics, but have dealt largely with lexical rather than rhetorical issues. Further, one must ask if the real problem has been addressed at all. The approach could certainly produce manuals that are internally consistent, which must be helpful in some way, but the problem of deciding which rhetorical and grammatical forms to include in the specification at all remains unaddressed. Consistency for its own sake will not necessarily lead to effective instructional text.

2.2.2 Psychological Work on Instructional Text

There have been a number of psychological studies of instructional text (Dixon et al., 1988; Dixon, 1987b; Dixon, 1987a; Dixon, 1982; LeFevre and Dixon, 1986; Jou and Harris, 1990; Wright and Wilcox, 1978).

The attraction of working on imperative instructional text to some psychologists is that the results of reading need not be discerned with recall studies, but rather by directly observing the subject's performance of the task specified. This avoids the questions concerning the validity of recall tests as a measure of understanding (Sawyer, 1991; Wright and Wilcox, 1978). Various sorts of recall tests may not actually address the question of understanding at all. The performance criterion, on the other hand, because it measures the ability to perform the specified task based on reading the instructions, more closely matches the real world application of instructional text, and thus provides a more useful measure of readability.

The basic approach of this work is to test, using the techniques of experimental psychology, the relative efficacy of various forms of imperative instruction. This involves the choice of a set of grammatical forms to test and the construction of an experimental setting in which to measure the reading speed and execution accuracy of the instructions. For example, Dixon (1988) tested the efficacy of commands like:

(14a) Turn the left knob to set the top meter to 20.
(14b) To set the top meter to 20 turn the left knob.

The results indicated that example (14a) was better, lending support to the current study's emphasis on the importance of grammatical variations of this sort. Unfortunately, the results said nothing of the large number of other forms of expression that are commonly used in instructions to express this type of information, such as:

(15a) Set the top meter to 20 by turning the left knob.
(15b) For setting the top meter to 20, turn the left knob
(15c) Turn the left knob so that the top meter is set to 20.
(15d) Turn the left knob until the top meter is set to 20.

The current study's findings indicate that the form in example (15d) is more commonly used in contexts that could include this sentence. These psychological methods are designed to test the relative efficacy of a set of expressional forms, but do not provide an effective means for determining the appropriate forms to test. Given that we believe that the furnace text, for instance, is poorly written, what forms should be tested in order to improve it? The author developed a list of alternative forms for this example early in the IMAGENE project, none of which turned out to be commonly employed in the corpus, leading one to suspect our ability to invent relevant forms to test.

There has also been work exclusively on instructional text conducted with respect to instructions for filling out forms. This work will be discussed in the next section.

2.2.3 Psychological Work on General Instructional Material

This section discusses the considerable body of work on general instructional text which has gone on at least since Thorndike (1921). Good reviews and critiques of this work can be found in Sawyer (1991), Britton (1986), and in Pearson (1984), a collection of reading research work which addresses the general issue of readability from various perspectives. This section will not discuss this work in detail as it doesn't address the detailed issues of linguistic choice that the current study addresses. Rather, it will review the elements of this work that are relevant for the current study, including how this work addresses the problem noted above with psychological approaches in general, i.e., that of providing a principled means of determining the appropriate forms to test in the experiment.

2.2.3.1 Early readability measures

The earliest work on this subject concerned sentence-level measures only (Thorndike, 1921; Flesch, 1948), including issues such as word frequency, syntactic complexity, and the active versus passive distinction. These measures have had limited use as quick screening devices for identifying the general reading level of a text and perhaps as tools for getting a preliminary idea of the complexity of an instructional text. They are, however, severely limited in their ability to access the quality or effectiveness of an instructional text (Klare, 1984; Duffy, 1985). Further, their focus on sentence-level issues makes them largely unadaptable to the current study.

2.2.3.2 General text structure measures

The text structure work takes into account the importance of the structure or cohesiveness of the text, rather than just looking at the sentence itself. The basic work here generally proposes some psycholinguistic model of the text comprehension process and proceeds to manipulate the texts based on the predictions that the model makes concerning the predicted ease of comprehension of the text. There are a vast number of such models to choose from (Kintsch and van Dijk, 1978; Kintsch, 1974; Kintsch and Vipond, 1979; van Dijk and Kintsch, 1983; Frederiksen, 1972; Frederiksen, 1986; Meyer, 1975; Meyer and Rice, 1982; de Beaugrande, 1980).

These models have been heavily used in the psychological community to provide some handle on what aspects of the text must be manipulated in order to produce better text (Britton et al., 1990; Britton and Gülgöz, 1991; Wagenaar et al., 1987). Of considerable interest to some of these studies has been the issue of adjunct aids and typographical layout (Hartley, 1985; Hartley, 1986; Waller, 1979), an issue that the current study is not concerned with.

In general, this work addresses general semantic and pragmatic connections between *propositions* found in the text, but has not been overly concerned with the precise grammatical forms used to signal rhetorical structure. The precise grammatical nature of the proposition (text order, rhetorical and grammatical form) in Kintsch's work (Kintsch and van Dijk, 1978; Kintsch, 1988), for example, is in question. A prepositional phrase with a nominalization might or might not be considered a proposition, but even if it were its unique grammatical form would not be distinguished in any way. Even the construction-integration model, which works bottom-up, makes these same simplifications.

2.2.3.3 Revision analysis

Another fruitful area of work, often closely related to the work in the previous section, is the work on revision analysis (Britton, 1986; Britton et al., 1989; Duffy et al., 1989; Graves et al., 1988; Graves, 1991). The basic approach here is to take some written instructional material and have various revisions made, testing each one with respect to readability. The better texts are then analyzed to determine what it is about them that makes them better. Not surprisingly, this work has had difficulty finding factors that are specific enough to be used in an automated text generation system. The expert revisers do not always improve the texts and when they do, they don't have a concrete idea about what they did to improve it and why it works (Sawyer, 1991).

2.2.3.4 Readers' comprehension analysis

The final aspect of the text comprehension work on general instructional text is the use of evidence of reader difficulty to identify the specific spans of text that must be improved. This evidence has been taken from errors in filling out forms (Holland and Rose, 1981; Rose and Cox Jr., 1980; Charrow et al., 1980) and from verbal protocols (Flower et al., 1980; Brown and Clement, 1987). Again, this is very interesting work, but no principled means of determining the alternate forms of expression is given, rather, the focus is on providing a mechanism for determining ineffective portions of text.

2.2.4 IMAGENE

The root problem with applying these approaches in the current study is that they do not pay enough attention to determining the best detailed forms of expression to be tested and the contextual motivations for using them. Aside from the revision studies, very little of this work has looked at the text itself in order to determine ways to improve instructional text.[2] They have tended to make sweeping, decontextualized prescriptions concerning issues such as textual order and form, or have abstracted away from the grammatical form altogether. This makes their results difficult to apply directly in IMAGENE.

[2] Donin has done some work (1992) on eliciting real instructions but it focused primarily on finding the errors in the text elicited from non-professional instructional text writers, rather than using the elicited text as a source of ideas for improving instructional text.

The most applicable work here seems to be the revision studies, they, at least, have looked at real text revisions produced outside of a particular psychological theory, a methodology also adopted by Meteer (1990) in the context of text generation. The fundamental problem with this approach to elicitation of precise grammatical forms is summed up in a quote from Britton (1989, page 238):

> When we read the text pairs that were most improved, we get the distinct impression that there are other important text changes present in them, but that we are not yet able to specify them explicitly or articulate them in words. That we cannot articulate them is not particularly surprising when we remember that other classes of facts about our native natural language are often extremely difficult to articulate explicitly. Consider the case of grammar: A freshman daily makes correct use of grammatical rules that even the most knowledgable linguist cannot yet formulate explicitly. Because grammar is only one of several rule systems that govern text, the explicit rules for constructing good text may be several orders of magnitude more difficult to formulate explicitly than the rules for grammar.

The current study's contribution to this work is its attempt to get at just these rules. It proposes a principled way to look at real texts (either revisions or the originals) and discern what makes them effective by looking for consistent patterns of expression in context. The results of this effort can easily be seen as a linguistic model of effective instructions, one which could serve as a source of predictions concerning the most effective rhetorical and grammatical forms for instructions. These predictions could then be used to map out the contexts and expressional forms that should be tested using the psycholinguistic methodology discussed in this section.

2.3 Processing Instructional Text

This section reviews the relevant work on instructional text within the Artificial Intelligence community, specifying its relationship to the current study. This work can be broken down along two scales, the granularity of language that is of interest, ranging from sentence-level work to discourse-level work, and the orientation of the work, alternating between understanding work and generation work. This section begins with sections discussing the general work on language understanding, sublanguage analysis, discourse processing, and language generation as they pertain to the current study. While these bodies of work may not specifically address instructional text, they do contribute to the issues and methods employed in the IMAGENE project. A section will then discuss generation projects that specifically address instructional text. Finally, the aspects of this work that IMAGENE makes use of will be discussed.

2.3.1 Language Understanding

The earliest work in language processing focussed on understanding, dealing largely with sentences (good reviews can be found in Winograd, 1983; Allen, 1987; Grishman, 1986). This work has contributed greatly to the understanding of the linguistic competence needed to parse and understand sentence-level spans of natural language and correspondingly to the work on the linguistic competence needed to generate natural language sentences.

Some of this work, specifically the work on understanding instructions, is of great interest to the current study. This work, as is typical of understanding research in general, has paid a fair amount of attention to the expressional form of instructions; the systems must, after all, be capable of parsing and understanding the forms used in instructions. This concern has lead the researchers to catalogue the various expressional forms found in instructional text corpora, such as those used for purposes (Di Eugenio, 1992), repetitions (Rock, 1992), free adjuncts of various types (Webber and Di Eugenio, 1990), and other general instructions (Badler et al., 1991; Alterman et al., 1991). Although the results of this work have been very helpful to the IMAGENE project, their primary concern has been to determine the nature of the procedural relationships and how they are to be represented. The effect of the issues of textual and interpersonal context on expressional form are largely ignored. Unfortunately, the forms of expression used in examples such as the Furnace text and the Installed-Phone text, as will be seen later, cannot be explained without reference to precisely these issues.

2.3.2 Sublanguage Analysis

The extreme difficulty of accounting for all of natural language has lead some understanding researchers to consider the specification the characteristics of more tightly defined text types. This work, called *Sublanguage Analysis* (Harris, 1968; Grishman and Kittredge, 1986), has been applied mostly in understanding work (Sager, 1981; Sager, 1986), but also to generation (Kittredge et al., 1986) and machine translation (Lehrberger, 1982; Kittredge, 1982; Kittredge, 1983).

Kittredge (1983, page 112), for example, has given an overview of what he terms the "the interesting similarities between technical manuals from very different semantic areas" by discussing aviation hydraulics manuals and recipes. This interesting similarity has been noted in the current study as well and will be discussed in some detail in Chapter 7. His characterization of these two types of instructional text include issues at the macro-structural level and the lexico-grammatical level. He notes, for example, that at the macro-structural level there tends to be tripartite division including a descriptive introduction, a list of components or ingredients, and a procedural section. At the lexico-grammatical level, he notes that procedural text usually deletes definite articles and repeated definite object noun phrases. These characteristics are certainly of interest in automating the generation of instructions, but do not give enough details to support their automated generation.

More detail on this issue is provided by Lehrberger (1982) in the same volume. He discusses, in some detail, the sublanguage used in aircraft maintenance manuals and how its characteristics affect the automated translation of this type of manual. The characterization of the sublanguage is quite specific, including restrictions concerning the lexicon, syntax, a host of semantic issues, and text macro-structure. The primary observation that will be made in this section is that the analysis is restrictive in nature, not predictive.

At the lexical level, for example, Lehrberger observes that the corpus only contains 4876 lexical items, and does not include certain syntactic constructions such as questions, tag questions, exclamations, and simple past tense. He notes, however, that aside from these restrictions, the full range of syntactic forms are used, such as the passive, restrictive and non-restrictive relative clauses. This type of restrictive analysis is primarily aimed at building a parser that can handle all of the forms necessary for the sublanguage. Such accounts are of limited use in text generation. Text generation requires predictive accounts, that is accounts that determine which of the full range of syntactic forms is appropriate for a particular context. It is not good enough to observe that imperatives are common in instructions, we must have some explicit, predictive model of when to use them and why.

Another example of this is Lehrberger's account of definite articles. Although this has not been a concern in the current study, it is illustrative of the issue at hand. He observes that definite articles are often but not always deleted. He reports (page 89) "It does not seem to be the case that in some contexts the definite article is always omitted while in others it is not. We can only say that it may be omitted and very often is." He has evidently asked the appropriate question, whether there is some contextual or syntactic feature or set of features that would predict the use or deletion of the definite article. He gave up, however, when no completely accurate predictor could be found. The approach adopted in the current study is to find statistically significant correlations between functions and forms, regardless of whether they correctly predict 100% of the examples in the corpus or not.

In summary, the sublanguage work reviewed here has provided solid observations concerning the form of instructional text, a number of which have been useful in the current study. The primary problem is that the characterizations are restrictive in nature rather than predictive.

2.3.3 Discourse Processing

The natural next step beyond sentence-level processing is discourse-level or text-level processing. As in Psychology, the Artificial Intelligence community is greatly interested in determining the appropriate scheme for representing and reasoning about discourse. The various schemes proposed for this include

Rhetorical Structure Theory, discussed in some detail in Chapter 1, and a number of other formalisms and variants. This section will discuss some of these other methods, particularly with respect to RST.

2.3.3.1 TEXT

TEXT is a landmark generation system developed by McKeown (McKeown, 1982; McKeown, 1985) which used a high-level schema to guide the generation process. It has given rise to a number of similar methods applied in other domains (Paris, 1988; Paris, 1987; McCoy, 1985; Rösner, 1987). COMET (McKeown et al., 1990), an instructional text generator that will be discussed below, employed this same technique.

In her original work, McKeown was primarily concerned with the problem of content selection. She studied a number of expository texts and identified a set of macro-structures which were commonly used in particular discourse contexts. These macro-structures, which she called schemas, were used to determine the content of an explanation and to organize this data into a high-level rhetorical structure. The schema types she identified were the Identification schema, the Constituency schema, and the Compare and Contrast schema. They were implemented as Augmented Transition Networks (Woods, 1973; Bates, 1978).

The problem with applying the results of the TEXT project to the current study is that, first of all, McKeown did not deal with instructional text. Secondly, and more importantly, she did not address the diversity of possible forms of expression for each rhetorical relation handled by her system. The fundamental contribution of her work concerned the use of discourse schemas and focus constraints on content selection. Little was ever said about the diverse forms of expression that might be used to express any part of the resulting text. Consider the following output produced by TEXT using the identification schema (McKeown, 1985, page 221):

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

The point of interest here is the repetition of the grammatical form "An <object> is an <object-superclass>." It is most likely that this repetition of forms does not represent the diversity of the forms used in the texts she analyzed, although this is difficult to judge as no explicit measurement of the match between the corpus and the produced text was ever given. It is this diversity and the resulting match between the predictions of the system and the real data that are the core of the current study.

2.3.3.2 Sanders and Knott

Both Sanders (1992) and Knott (1992) are interested in deriving a more principled set of rhetorical relations for use in discourse analysis. Sanders employs a psychological methodology, while Knott employs a linguistic one. Each will be discussed in this section.

Sanders et al (1992; 1993) attempt to produce a list of features of coherence, at a lower level of abstraction than the relations in RST, that can be specified in groups to define the nature of inter-clausal relations. This is an interesting idea, but it was unclear at this stage of development what this approach could buy in the current study. In the current study, the forms of expression of the procedural relationships found in instructional text are of interest, breaking each of the useful rhetorical relationships down into psychologically motivated sub-components is not particularly helpful.

Further, one critical issue in instructional text is temporal sequence, an issue which they dismiss as not being a "basic categorizing principle" (Sanders et al., 1992, pages 27–29). They point out that temporal relations, unlike the other components of rhetorical relations they discuss, are largely determined by the nature of the individual spans of text, and their order may not be altered with out changing the meaning of the passage. Certainly, temporal relations are not identical with other relations, but they are a distinct procedural phenomenon that must be differentiated from other relations. Thus, their specification must be separate from the specification of other relations for the current study.

Knott and Dale's linguistic approach (1992) is more closely related to the current study. It employs a form to function study of cue phrases in academic English text in an attempt to derive a more principled set of rhetorical relations. It stands in interesting contrast to the function to form approach employed in the current study. The relations used in the current study were procedurally motivated, using relations such as purpose, precondition, result, and sequence, which have a distinct status in procedural terms. This allowed the current study to determine all of the lexical and grammatical forms in which various procedural relations are expressed, including relations that are not lexically or grammatically marked, rather than looking specifically at cue phrases. Knott and Dale's work, although not directly contributing to the current study, stands as an interesting point of comparison.

2.3.3.3 Grosz and Sidner

Grosz and Sidner (1986) have proposed a model that incorporates elements of the text structure, intentional structure, and attentional structure. They see themselves as proposing a definite alternative to RST-like approaches, focusing largely on the intentions involved in discourse, subjugating the rhetorical relations between spans of text to these intentions. This approach has been helpful in their work with interactive discourse and in text where intentions are complex and varied.

The instructional text that is of concern in the current study has not been interactive and does not tend to display the varied intentions that constitute the bread and butter of Grosz and Sidner's approach. Rather, the current study is interested in explicitly representing the rhetorical relations so that a predictive mapping from relation to grammatical form can be found. Thus, the subjugation of rhetorical relations to intentions does not match the needs of the IMAGENE project.

Further, IMAGENE is concerned with the grammatical instantiations of inter-clausal relations and even clause/phrase linking. The Discourse Segments of Grosz and Sidner's work tend to be larger, intention oriented units, that are of less use to the current study. RST provides a more suitable granularity of analysis.

An interesting combination of the intentional and relational representations is proposed by Moore and Pollack (1992). They argue that a distinctions should be made between *intentional* relations and *informational* relations, and that they should be allowed to coexist in the discourse representation. This is a very nice synthesis of the two approaches, but would not enhance the current study because of its primary focus on the informational, procedural relations.

2.3.4 Language Generation

Natural language generation systems surfaced as simple components of more elaborate understanding systems. It was felt at the time that the generation task could be hacked fairly easily with canned text and template based systems. As expressed by Grishman (Grishman, 1986, page 159):

> ...., language generation has generally taken second place to language analysis in computational linguistics research. This imbalance reflects a basic property of language, namely, that there are many ways of saying the same thing. In order for a natural language interface to be fluent, it should be able to accept most possible paraphrases of the information or commands the user wishes to transmit. On the other hand, it will suffice to generate one form of each message the system wishes to convey to the user.

This characterizes the general view of generation research in the 60's and 70's. By the early 80's, however, researchers realized that canned text and template-based systems were grossly inadequate to deal with complex contexts in generation. This realization is at the root of the current study. The "many ways of saying the same thing" referred to by Grishman are not all appropriate in the same context. The task of the current study is to determine the specific form of expression of the "same thing" that is appropriate for the various relevant contexts in instructional text.

It is not the purpose of the current section to review the entire field of language generation, this has been done elsewhere (McKeown and Swartout, 1988; Dale et al., 1990; Bateman and Hovy, 1991). It will briefly review some of the relevant work on sentence-level generation systems and then discuss a number of directly related text-level generation systems. Although none of these systems directly addresses instructional text, they are, nonetheless relevant to the current study because of their varied approaches to the problem of handling diverse forms of expression. The text generators specifically dealing with instructional text will be discussed in the next section.

2.3.4.1 Sentence-level generators

There are a number of sentence-level generation systems that have been developed. Among the more comprehensive generators are Penman, discussed in Chapter 1, the various instantiations of Mumble (McDonald, 1983; McDonald and Pustejevsky, 1985; McDonald, 1980; Meteer et al., 1987), which pays particular attention to some of the psycholinguistic aspects of generation, FUF (Elhadad, 1992; Elhadad, 1991), which employs a Functional Unification Grammar (FUG), IDAS (Reiter and Mellish, 1992), which uses KL-ONE-style classification to perform realization, and KING (Jacobs, 1985; Jacobs, 1987), which uses a phrasal approach. These sentence-level systems typically do not address the issue of choosing between alternate grammatical forms of expression[3]. As was discussed in Chapter 1, Penman suited the needs of the current study better than any of the others.

Given Penman, a broad-coverage text generation system, the problem of managing diversity of expression in text generation can be more effectively addressed. McKeown's TEXT system, particularly relevant to IMAGENE, was discussed in the previous section. The following sections discuss other relevant text generation systems.

2.3.4.2 Danlos

An early study of the linguistic issues in text generation was done by Danlos (1984; 1987). She focussed on the domain of two sentence expressions of direct causal relationships, those with an action and a result of that action as in "John shot Mary. He killed her." She identified four issues that were relevant in the generation of the discourse structure of this type of expression and the possible choices for each one:

- Ordering of the information (Act-Result or Result-Act);
- Clause combining (Separate sentences or one sentence);
- Form of the action (active, passive, agentless passive);
- Form of the result (active, passive, agentless passive).

She first noted that of the 36 possible combinations of these choices, only 15 were acceptable. These 15 acceptable combinations were coded in what she called a *discourse grammar*. One set of choices that she considered unacceptable, for example, was the expression of an agentless passive and a passive in separate sentences such as "Mary was shot. She was killed by John." She rejected this pair because it didn't seem as interpretable as, for example, "Mary was killed by John. She was shot."

In addition to the choice of discourse structure, she also showed how the properties of various lexical items affected the choice of discourse structures. After an in-depth introspective analysis of the relative acceptability of various combinations of these choices, she concluded that no strict ordering of the choices would be acceptable in the general case (1984, page 503):

> ...., we have shown that decisions about lexical choice, order of the information, segmentation into sentence and syntactic constructions are all dependent on one another. This result is fundamental in generation since it has an immediate consequence: ordering these decisions amounts to giving them an order of priority.

Her solution to this problem was to prescribe an architecture which separately determined the sub-set of the 15 acceptable discourse structures that were selected by: (1) syntactic concerns, (2) semantic concerns, and (3) stylistic concerns. These sub-sets were called L_1, L_2, and L_3 respectively. The

[3] For a notable exception see Green's work on integrating style into Penman 1992; 1993). This work is based on the theory of stylistics developed by DiMarco (1990).

intersection of these constrained sub-sets was used to determine the correct discourse structure, thus allowing the decisions to be made in no particular order. If this set was empty, there was an elaborate algorithm for relaxing certain parts of the analysis.

The current study agrees with her basic assessment of the unorderability of the choices. IMAGENE uses the system network formalism to express all of the choices in the order appropriate for the specific aspect of instructional text that is being addressed at that point, independently of the order used for other aspects. It also agrees with her general methodology of studying the linguistic constraints of a specific genre as a way of providing the text generator with a subset of forms that are appropriate for expressing the information from that genre. The current study, however, has improved on two aspects of Danlos's approach: (1) it addresses the problem of diverse forms of expression in context, and (2) it bases its choices on the results of a detailed corpus study.

Danlos did not present a full solution to the problem of choosing among diverse forms of expression. She discusses constraints on expression but gives no indication of how to decide upon the choices if these constraints leave more than one choice. She admits that when the intersection of L_1, L_2, and L_3, she does little or nothing (1987, page 98): "When this intersection is not empty, there is no problem. If it contains several elements, either one of these elements is picked out at random, or the generator produces a text for each element with the final choice being left up to the user."

Her intent is that the syntactic, the semantic, and particularly the stylistic constraints will make the choice between various forms of expression. Unfortunately, her set of constraints does not take the full communicative context into account. They appear to do a good job of dealing with syntactic, semantic, and some stylistic issues, but leave out certain interpersonal and textual issues that will be seen, in later chapters, to be significant in the generation of instructional text.

The second issue is that she doesn't base her prescribed constraints on the results of a corpus analysis. All her examples appear to be contrived, and the corresponding analysis questionable. For example, consider the choice between the following forms (1987, page 97):

(16a) The Pope was wounded yesterday in Paris. Anarchists blew up his house.

(16b) Anarchists wounded the Pope yesterday in Paris. They blew up his house.

(16c) Anarchists blew up the house of the Pope who was wounded.

She asserts that example (16a) seems most satisfactory and concludes that "This indicates that a terrorist crime against a famous political figure is better expressed with the result coming first in a separate sentence, and if possible in the passive." (1987, page 97). This is certainly a good hypothesis to start with, but without empirical justification it is little more than a guess. In her defense, she does occasionally reference linguistic research justifying her claim, but this is not sufficient to completely address the problem.

2.3.4.3 PAULINE

A study done by Hovy, as part of his thesis work, resulted in the construction of PAULINE, a text generator that generated text constrained by various elements of the pragmatic context (Hovy, 1988a; Hovy, 1990). The system was capable of producing an impressive range of expressional forms, based on a list of pragmatic features of the communicative environment including information about the conversational atmosphere, the speaker, the hearer, the relationship between the two, and the interpersonal communicative goals of the speaker. As an example of the range of issues Hovy dealt with, consider the following three different versions of the same story, all generated by PAULINE (Hovy, 1988a, pages 7,8). The first text is a relatively neutral presentation of the facts:

> On 20 February, Carter and Kennedy were the candidates in a primary in Michigan. Carter lost to Kennedy by 1335 votes. At present, Kennedy has better chance of getting the nomination than he had before. Carter is also closer to getting the nomination than he was before. Both Carter and Kennedy want to get the nomination.

In the following text, PAULINE is told, among other things, that the speaker is pro-Carter: "I am pleased to inform you that Carter has improved his chances of winning the nomination. At the present time, Carter has many more delegates than he had in the past; Also, Carter has many more than Kennedy has."

In the last text, PAULINE is told that the allotted time for the communication is very short: "Well, so Carter lost the primary to Kennedy by 1335 votes."

The attempt to produce such a large range of expressional forms and to motivate the production of each one via detailed elements of the pragmatic context required a considerable amount of analysis of sample texts. Very little, however, is said about how this analysis was performed. In his analysis of the characteristics of formal text (Hovy, 1990; Hovy, 1987), Hovy presents a sample analysis of a particular text, describing the style in terms of the decisions a generator must make. Lexical selection, for example, is a decision that must be made by the generator, so the lexical selections of the text are considered as to their effect on formality. He reports (1990, page 172) that "A number of texts, ranging from politicians' speeches and writing to discussions with friends, were analyzed in the manner above." The results of the analysis are exploited by PAULINE, making what sounds like formal text. One is left wondering, however, what texts were in the corpus and precisely how well the analysis matches those texts. A similar approach appears to have been taken to the effect of opinion on expressional form (Hovy, 1986).

The results of the PAULINE project are an impressive array of generalizations about language in spoken and written contexts, based on an analysis of a broad range of texts. The IMAGENE project is similar in nature, except that both the text type, instructional text, and the generator decision of interest, expressing rhetorical relations, are much more focussed. The results of the study are therefore highly specific to instructional text and correspondingly more detailed. The problem of determining where the rhetorical relations are in an instructional text and then cataloging the range of forms of the contexts in which they appear is a more concrete task than looking for the expressions of formality, wherever they may appear. Further, the actual predictive performance of IMAGENE is statistically analyzed with respect to the corpus of texts.

2.3.4.4 The ISI text planner

This section discusses a particularly relevant application of Penman's systemic generation to the generation of text or rhetorical structures. The system, as described by Hovy (Hovy, 1988b; Hovy, 1989), uses the rhetorical relations as defined in RST (Mann and Thompson, 1989) as plan operators in a hierarchical planner in the style of Sacerdoti (Sacerdoti, 1977). This planner produces a rhetorical plan which is passed to Penman for generation. The system is designed to address narratives in the domain of naval operations and apparently is used in two other domains as well.

The RST planner itself required the formalization of the RST rhetorical relations with intentional predicates, modeled after Cohen and Levesque (Cohen and Levesque, 1985). These predicates were used to indicate constraints on the nucleus and satellite of the relation, as well as the intended effects of the relation as a whole. Given this annotation, the planning system starts with a database of facts about the domain, represented in NIKL (Kaczmarek et al., 1986), and takes as input a primary communicative goal. It then proceeds to plan a hierarchical structure of RST relations designed to meet the goal. The resulting text displays a coherent rhetorical organization based on how original goal and the database interact with the precise formalization of the RST operators.

The point of interest for this study is the resulting text realization. The planner itself makes no specifications with respect to how the chosen rhetorical relations are to be expressed. This task is left to Penman which has a rudimentary implementation of the rhetorical relations. In the case of a purpose, for example, Penman will produce a non-fronted "in order to" infinitive clause, as in the following output of the ISI planner (Hovy, 1988b, page 167): "Knox is en route in order to rendezvous with CTG 070.10."

This planner/generator is a good example of the approach which predetermines a set list of grammatical forms that are sufficient to disambiguate the rhetorical relations that are chosen for implementation. The text is not particularly bad or incomprehensible, but still does not reflect the variation of form found in real text.

2.3.4.5 Spokesman

Meteer proposed an architecture for addressing what she called the problem of *expressibility* in text planning (Meteer, 1991; Meteer, 1990). Her fundamental thesis is that an abstract linguistic representation is needed in the text planner. This representation should perform two important tasks: (1) It should ensure that the text planner never specifies a text plan that is not "expressible" and (2) that the text planner should have access to the full "expressiveness" of the realization component. She proposes a well defined representational formalism, *Text Structure*, to address these issues.

Her Text Structure represents the aspects of both the rhetorical structure and the sentence structure that were deemed relevant in a study of textual revisions made by professional editors on technical papers. The representation was just abstract enough to allow the planner convenient access to the linguistic tools provided by the text realizer and the constraints on their use, and just detailed enough to allow access to full range of grammatical forms that were available.

As the current project is using a Text Representation Language (called TRL), which is largely based on RST, to serve the function of Meteer's Text Structure (TS), it is helpful to review her critique of RST (Meteer, 1991). She identifies two problems with RST, as it is used in the ISI text planner discussed above.

First, she notes that TS deals with elements at a lower level of abstraction than the clause level, whereas RST deals only with clause level units and above. This critique is mostly correct for the ISI text planner's use of RST, but not for RST in general. Although the vast majority of work with RST has not looked below the clause level, this restriction is more cultural than definitional. Mann and Thompson (1987a) define the lowest level of rhetorical analysis, called the text span, as "an uninterrupted linear interval of text" which could include phrases or even smaller constituents. The admission of lower-level spans is not common in RST work, but it has not been strictly prohibited (Matthiessen and Thompson, 1987). IMAGENE freely allows rhetorical relations to be expressed in sub-clausal units such as prepositional phrases with nominalized complements provided the sub-clausal forms are actually used productively in the corpus.

Secondly, she notes that RST is required to posit a rhetorical relation between to two elements of the text plan before they may be expressed. Her example centers around the following expression produced by the ISI text planner (Hovy, 1989): "Knox, which is C4, is en route to Sasebo. It is at 18N 79E, heading SSW. It will arrive on 4/24. It will load for 4 days."

The point she makes is that although the spans "It is at 18N 79E" and "heading SSW" are two properties of the Knox that are not really related to each other, the ISI planner posits an elaboration relation between them. This critique is, again, true enough for the implementation of RST in the ISI text planner, but does not necessarily apply to RST in general. RST provides a joint schema for joining elements that do not display nucleus/satellite relations. IMAGENE is capable of using this schema when necessary.

Meteer's fundamental concern is to provide the text planner with the following information (Meteer, 1991, page 300): "The text planner must know (1) what realizations are available to an element that is, what resources are available for it, (2) the constraints on the composition of those resources, and (3) what has been committed to so far in the utterance that may constrain the choice of resource."

Her concern, as can be seen here, is primarily with constraints on text planning (see also Huettner et al., 1987). Item (1) refers to the various forms that a TS node could take, while items (2) and (3) refer to constraints that affect the acceptability of the form. For example, she allows her text planner to produce either verbal or nominalized expressions of an event, as in the following examples, but provides facilities for disallowing certain unacceptable forms such as the one in example (17b) (Meteer, 1991, page 300):

(17a) Michael quickly decided to go to the beach.

(17b) *Michael importantly decided to go to the beach.

This is an extremely important task; one which IMAGENE has addressed by allowing its system network to produce only those forms that were found in the corpus. The second task of concern to IMAGENE, that of choosing between the acceptable alternative forms of expression, is largely ignored by Meteer. She doesn't address what to do if, after using her constraints to weed out the unacceptable forms of expression, there are a number of remaining acceptable forms (Meteer, 1990, page 171): "If more than on choice (alternative) in a class is acceptable, then a simple mechanism, such as taking the first choice or the most specific choice, is used to select one of them."

This issue of choice is the key concern of the current study.

2.3.5 Generating Instructional Text

One would expect the choice of expressional form to be the fundamental concern of the work on generating instructions, but this is not always the case. Although generation researchers have been greatly interested in the issue of referring expressions (Reiter, 1990; Dale, 1990), this level of concern has not extended to the rhetorical level. Instructional text generation systems (Dale, 1990; Dale, 1992; McKeown et al., 1990; Mellish and Evans, 1989; Reiter et al., 1992; Rösner and Stede, 1992b), dealing with other issues, have tended to hard code choices at the rhetorical level. This approach is at variance with the practice in actual instructional manuals, where the actions being specified are systematically expressed within any one of a number of rhetorical relations, depending upon the functional context, and, similarly, each rhetorical relation is expressed in any one of a number of grammatical forms, again, depending upon context.

This section will discuss a number of the most important text generation systems which are particularly concerned with instructional text.

2.3.5.1 Mellish and Evans

A prime example of the generation of instructions from plans can be found in the system built by Mellish and Evans (Mellish, 1988; Mellish and Evans, 1989). Their system is the first attempt to actually build an instructional text generator using the output of an planner as input. The planner that they used was Tate's NONLIN (Tate, 1976). Their intent was to produce a system against which future work in this area could be compared. In this spirit, IMAGENE can be seen as an attempt to address one particular simplification that they made in their work. The simplification in question is the small range of rhetorical and grammatical forms that they used in their generation.

Their system used the planner output as a preliminary rhetorical structure for the text. Because this often produced hard to understand or monotonous text, they included a *message optimization* phase which specified rules for removing or modifying certain elements of the plan structure that were known to produce poor text. This greatly improved the text. They noted, however, that even after this, some of the text tended to be "repetitive and quite hard to follow" (Mellish, 1988, page 245). In their defense, some of the plans they looked at were quite complex and correspondingly difficult to express. This difficulty has lead the current study to look at real instructional texts to find useful principles of expression. Mellish and Evans' study does not appear to have taken advantage of such a study (1988, page 243):

> In the absence of a formal and detailed psychological theory of discourse comprehension, researchers in natural language generation are reduced more or less to using their intuitions about whether one way phrasing something is "easier to understand" than other. We have regretfully had to follow the same course in designing and evaluating our own system.

The approach advocated in the current study attempts to address this problem of expression by basing the decisions made in the generator on the results of a detailed corpus study of instructional text and by verifying its operation on that corpus.

2.3.5.2 Epicure

Dale addressed the problem of referring expressions using an architecture similar to Mellish's. EPICURE, the result of this study, formed the basis of Dale's thesis (Dale, 1990; Dale, 1992; Dale, 1988). He did not specifically address the expression of rhetorical relations in the texts, but his approach is still instructive.

The critique here is similar to the one given of Mellish and Evans' system in the previous section; EPICURE's output, at least at the rhetorical level is not based in any measurable way on a corpus of real text. Dale's primary purpose was to look at referring expressions, and although it is unclear if he ever took detailed look at a corpus of recipes, he does at least make reference to such a study, performed by Hobbs (1978). This reference and his concern for the specific issue of referring expressions leads his work to produce what appear to be very satisfying results with respect to that issue. One still must wonder, however, how true to the real practice of writing recipes his results are. IMAGENE, nonetheless, can be seen as a similar approach to the rhetorical expression of procedural relationships.

One other issue of interest with respect to Dale's work is the characteristics of recipes as opposed to other forms of instructions. One of the interesting results of the current study is that the predictions concerning the rhetorical structure of instructions in the context of telephone manuals appear to extend reasonably well to the structure of recipes. This is surprising because recipes are typically produced and consumed by very different sorts of people than are other forms of technical documentation. This issue will be discussed in detail in Chapter 7, but it should be noted here that the conclusions of the current study do not pertain to referring expressions. Evidently, referring expressions are one of the issues, not addressed in the current study, that distinguish recipes from instructions for other forms of non-creative processes. Dale's work is an important study of what causes these differences.

2.3.5.3 COMET

Another major system designed to produce instructional text is COMET (McKeown et al., 1990). COMET is an application of the basic schema-type approach to text structuring, originally used by McKeown in her thesis work (McKeown, 1985), to the domain of interactive instruction concerning army radio technical manuals. One of the primary interests of this project is to coordinate graphics and instructional text meaningfully (Feiner and McKeown, 1990). WIP, a system developed by Wahlster et al (1993), shares this concern, but does not generate the range instructional text of interest to the current study. As was the case with TEXT, COMET identifies the appropriate response schema for the question and proceeds to run the schema, accessing a database of facts about the device.

The schemas make use of preconditions, side-effects, and sub-steps, very similar to the procedural terms of analysis used in IMAGENE. Once these terms are assembled into a rhetorical structure, text is produced using FUF (Elhadad, 1991). Here is an example of such text (McKeown et al., 1990, page 109):

> Set the channel knob to position 1. Set the MODE knob to SCX. Set the FCTN knob to LD. Now enter the frequency: first, press the FREQ button. This causes the display to show an arbitrary number. Next, in order to clear the display, press the CLR button. Next, enter the new frequency using the number buttons. Next, record this number in order to check it later. Finally, press Sto ENT. This causes the display to blink.

First, notice that action sequences are coded using "First, ... Next, ... Finally, ..." and that purposes are coded using either the form in "Now enter the frequency: first, press the FREQ button. ...." or the one in "in order to clear the display". It is unclear what the procedural difference between these purposes is. Results are expressed as in "This causes the ...."

Because so few examples of the text were included in the reports on COMET, it is impossible to say with certainty that these are the only forms that are produced. Clearly, however, the issue of variety of expression depending upon functional context was not a primary concern of the project.[4]

This system did contain a serious attempt at a corpus study, based on Smadja's colocational analysis (Smadja, 1989b; Smadja, 1989a; Smadja and McKeown, 1991). This is good in that it addresses the real text, but the methods are too low-level to address the issues that are of interest in the current study. The primary concern appears to be with discovering and encoding two word collocations for use in lexical selection. This is the sort of problem well suited to a statistical study of a large corpus. The analysis of rhetorical relations, which cannot reliably be identified by surface syntactic characteristics, is quite another matter, requiring a knowledge intensive study of a smaller corpus of texts.

[4]The underlying generation system, FUF (Elhadad, 1992), however, has been used in several more linguistically motivated studies (Elhadad and Robin, 1992; Robin, 1992).

2.3.5.4 TECHDOC

TECHDOC (Rösner and Stede, 1992b) is Rösner and Stede's system for generating multilingual instruction manuals. It's general approach is very similar to the methods of the other instructional text generation systems discussed in this section as well as IMAGENE, namely it takes a non-linguistic plan structure as input and produces instructional text. The point interest for the current study is its concern with the use of RST, particularly in the context of multiple languages.

The use of multi-lingual text generation from language independent representations is a promising alternative to machine translation (Kittredge and Polguère, 1991) as it helps avoid many of the problems of text understanding that arise in translation. It still, however, faces the task of managing the diverse forms of expression used in written language. Rösner and Stede (1992a) address this problem with by giving examples of how RST can be used to address the variations in style found in expressions of the same procedural information in multiple languages.

One issue that surfaces immediately in their analysis is the fact that one language's clause is another language's phrase. As a result, they conclude that the inclusion of sub-clausal spans of text in rhetorical relations is important. This observation was made in the current study as well (Vander Linden et al., 1992b), although not in the multilingual context. It was made here because of the use of phrasal forms and clausal forms to express certain procedural relations even within a single language.

Another point of agreement with the current study is on the usefulness of RST as a flexible tool for organizing instructional text, particularly at the local level. Rösner and Stede, as in the current study, make extensive use of RST constructs, and freely make modifications to the inventory of relations provided for analysis as the situation dictates. This approach emphasizes the fact that discourse analyses may be made a differing levels of abstraction, requiring different inventories of relations. The claim made in RST that a certain small subset of relations was sufficient to address all rhetorical relations in discourse (Mann and Thompson, 1989) may be true at a gross level, but studies of specific forms used to express specific functions may require a custom tailored inventory of relations. TECHDOC, for example, includes relations for precondition expressions and "until" clauses. IMAGENE, as well, modifies the inventory of relations. This modification is discussed in Chapter 3.

The fundamental contribution of the current study to the TECHDOC project is the corpus study, at least for English, of the detailed lexical and grammatical forms used to express various procedural relationships found in instructional text. TECHDOC uses a set of forms of expression for the various relations it addresses that is slightly more diverse that is found in the other systems discussed in this section, but still is not complete. TECHDOC expresses purpose, for example, in one of the following three ways:

- A fronted "to" infinitive clause, as in "To X, do Y."
- A non-fronted "in order to" clause, as in "Do Y in order to X."
- A separate sentence as in "Do Y1, Y2, ... As a result, X."

These forms were presumably gleaned from a detailed reading of the automobile repair manuals that were used as a model for the generation process in the study. The method of differentiating the contexts in which these forms would be used is based on structural constraints and some explicitly coded "preferences." TECHDOC also has some ability to deal with mutually interacting constraints, particularly with respect to clause combining.

This list of forms is more extensive than is commonly the case in instructional text generation systems, but still does not represent the diversity of forms of expression found in the current study. This is perhaps because Rösner and Stede focussed on a corpus of automobile manuals, perhaps a very small one, which did not include the full range of purpose forms found in the current study. Neither do they provide any empirical justification for the selections made my TECHDOC. These latter two issues, the use of diverse forms of expression and the empirical justification constitute the fundamental concerns of the current study.

2.3.6 IMAGENE

IMAGENE's contribution to the field of natural language processing, particularly to the text generation community, is the proposal and demonstration of a detailed corpus-based analysis of the forms of expression used in instructional text to convey procedural relationships in the process structure. The emphasis on quantitative analysis of a corpus of the relevant type of text is crucial. Further, the implementation of the results of this analysis in IMAGENE can serve as a guideline for future generation efforts.

2.4 Summary

This chapter has reviewed the work in Linguistics, Psychology, and Artificial Intelligence that has a bearing on the current study. Linguistics was seen to have contributed much of the theoretical foundation for the work, as well as a number of the most important analytical methodologies and linguistic formalisms. Psychology was seen as a potential beneficiary of the results of the current study, as well as contributor of a number of possible hypotheses to be tested in the corpus analysis. Artificial Intelligence was seen to be the contributor of the major computational mechanisms used in the current study. The remaining chapters will discuss how all these contributions play out in the corpus analysis and the implementation of IMAGENE.

Chapter 3

The Corpus

The range of grammatical coverage offered by current tactical text generators allows considerable flexibility in the forms that may be chosen to express a given concept. This flexibility, however, gives rise to the problem of choice in generation. Which of the many different forms that express the same basic idea should be used to express this concept in this communicative setting? Nowhere is this problem more evident than in text-level generation, where the choice problem exists at the text level as well as at the sentence level.

A typical approach to this problem is to let our intuitions do the talking, that is, continue to adjust the text generator until the output sounds good. The resulting text is good, but only to a first approximation. Discerning the nature of effective expression requires that we move beyond our intuitions and look at real text. The forms of expression that are used in real text are typically the ones found most useful in expressing a concept. This basic approach conflates "what is" with "what ought to be", that is it assumes that the forms that are consistently used in a genre are the ones which are the most effective. Employing this corpus-based method requires that a set of examples of the text type of interest be gathered and carefully studied, looking for consistent patterns of use. This approach lends a credible reality check to the text generated by text generation system. Mapping functions to forms in a large corpus, however, is a daunting task, requiring some level of computational support before it can be attempted.

An effective approach for computational support that has come out of Discourse-Functional Linguistics is to encode a small text corpus in a relational-style database, encoding the various useful elements of the communicative context. To date, these systems have received little if any attention in the literature.[1] This needs to change. If some standard of representation could be established, the arduous task of encoding information on a particular text type in the first place could be performed once and the finished product distributed for the benefit of all. Clearly a definitive standard of representation is unattainable due to the variety of possible research questions, but the relational database model provides considerable flexibility for taking a baseline representation and tailoring it to suit specific research goals. This chapter specifies the system used for the current study, and may be used as a baseline for other research using this approach.

The type of corpus analysis advocated here and for which the database tools have been built differs from the large scale text corpora work done on data available from the ACL data collection initiative (DCI) (Liberman and Marcus, 1992; Church and Mercer, 1993) and from the even larger scale statistical analyses. This thesis advocates the detailed analysis of a relatively small corpus where the analyst has total control of the analysis process, as advocated by Cumming (1990). As will be seen in examples of the analysis for the current study, the determination of functional factors leading to lexical and grammatical choice requires explicit representation of varied functional features of the communicative setting. Before such an analysis has been done, the lexico-grammatical features that code these functional features cannot be reliably determined, thus automated parsers cannot be used to make the determinations required for the current study. The parsers would necessarily have to hard code answers to the very questions the study is attempting to address.

[1] For a notable exception to this see DuBois' work on representing interactive discourse and intonation units in a relational database (Du Bois and Schuetze-Coburn, 1993).

The current study, therefore, has been based on the collection and detailed analysis of a corpus of a broad range of instructional texts. This chapter discusses the contents of this corpus, the method by which it has been represented, and concludes with an example of how the corpus was used to address a particular function to form mapping issue.

3.1 The Nature of the Corpus

The general rule with respect to the size of corpora is that the bigger they are the better, that is, the more grammatical phenomena they exhibit, the more comprehensive an analysis must be to deal with them. This desire for size is counter-balanced by the reality that the theoretical tools to handle a corpus of all of English are beyond the state of the art, nor do we have the capability to collect such a corpus. This is true of syntactic studies and particularly true of the knowledge intensive, in-depth analysis performed for IMAGENE. Thus, the corpus of instructional text created for the current study is not as large as those created for other corpus-based studies in Computational Linguistics.

The corpus is made up of approximately 1000 clauses of instructional text (6000 words) taken from manuals and instructions concerning a variety of devices and processes, taken from 17 different sources. The attempt has been to include samples of text from sources covering a wide range of instructional text, made up of excerpts from the following sources:[2]

1. Three telephone manuals, two for cordless telephones and one for the GTE airfone (Code-a-phone, 1989; Excursion, 1989; Airfone, 1991).

2. Instructions for other consumer electronic devices including a television (Sharp, 1986) and a clock radio (Panasonic, 1987).

3. Excerpts from two auto-repair manuals (Reader's Digest, 1981; Volkswagon, 1987).

4. Excerpts from one first-aid manual (Rosenberg, 1985).

5. Two recipes (Ying, 1992; Knox, 1978).

6. Excerpts from computer software instructions (Macintosh, 1988).

7. Operation manuals for a child's toy (Bubble plane, 1991) and a garbage disposal (Waste King, 1979).

8. Excerpts from craft manuals for pottery (Gillberg, 1992) making jigsaw puzzles (Williams, 1991) home fix-up and repair (Reader's Digest, 1973) and on using a hammer (Livers, 1992).

The text in the three telephone manuals (item 1) served as a starting point for the current study. An analysis was performed on them alone (Vander Linden et al., 1992b; Vander Linden et al., 1992a), resulting in system networks very similar to those described in this thesis. The corpus was then expanded to include the other types of instructions listed above (items 2 through 8). There are several fundamental reasons for expanding the corpus in this way:

1. To define the limits of the instructional text register defined by the current analysis. It was unclear to what extent the results of the original analysis applied to other types of instructional text. For example, recipes are produced in very different communicative contexts than are, say, manuals for consumer electronic devices.

2. A larger corpus provides more examples of a few grammatical constructions that occurred infrequently in the phone manuals. Such constructions include "until" and "so that" clauses and expressions of concurrency.

---

[2] The code names used for each of these source texts are included in appendix A. These codes have been used throughout the analysis and the thesis to refer to particular source texts.

3. Some commonly studied lexico-grammatical forms did not occur at all in the telephone corpus and were deemed important enough to seek out. Such forms include "so that" clauses.

The current study's goal has been to perform a deep analysis of the telephone manuals, and apply this analysis to other forms of instructional text. Thus, the first reason is fundamental in this thesis. Chapter 7 will discuss the accuracy of the analysis for the various text types. The texts selected for the corpus were intended to be representative of the range of popular instructional text and were, thus, selected from typical instruction manuals that would be found in any household. This selection process, however, was not totally random. There were several issues that guided the inclusion of text in the corpus. First, only English texts were selected. The current study makes no explicit multi-lingual claims.

Secondly, the analysis excluded instructions that are not well written such as those obviously written by non-native English speakers. Here is an example of the first paragraph of an instruction booklet for an abacus (the errors in this sample are as they appeared in the original text):

```
                              ABACUS

                          Small and Handy
              but a remarkably effient Chinese Calculator

The Abacus, a sort of rectangular board with many beads-somewhat
similar to what children use at primary school to learncounting.
Similar calculators exist in several other countries, including
Russia as well as Japan, but with differentiating variations.
```

.... and a little later:

```
How to manipulate your fingers.

In pushing the counters, there are three ways.

1) Use only the forefinger.
2) Using the thum and forefinger.
3) Using the thumd, forefinger and middle finger,
The third one is the method with the thumb.
```

These types of obvious semantic, grammatical, and lexical errors would adversely affect the results of the analysis had they been included in the corpus.

The third issue concerns the problem of mixing text and pictures. Certain manuals have a preponderance of pictures, using text only as an annotation. The current study doesn't address the issue of mixing graphics and text, so these types of texts were not included in the corpus. There is a considerable amount of interest in precisely this problem in the computational linguistics literature (Arens and Hovy, 1990a; Arens and Hovy, 1990b; Feiner and McKeown, 1990; Wahlster et al., 1993). These texts were excluded because of the probable effect of the interaction of graphics and text on the lexico-grammatical forms that are used by the writers. Many of the texts in the corpus included a diagram of the device which affected the referring expressions that were used, but no other significant use of diagrams was made.

Finally, general instructional material, as defined in Chapter 1, was also excluded. This text is descriptive rather than prescriptive and is thus beyond the scope of this analysis. UNIX "man" pages, for example, are of this type. They describe what the command and its options will do. No mention is made of what readers must do; readers are assumed to be able figure that out on their own based on the frequent example given.

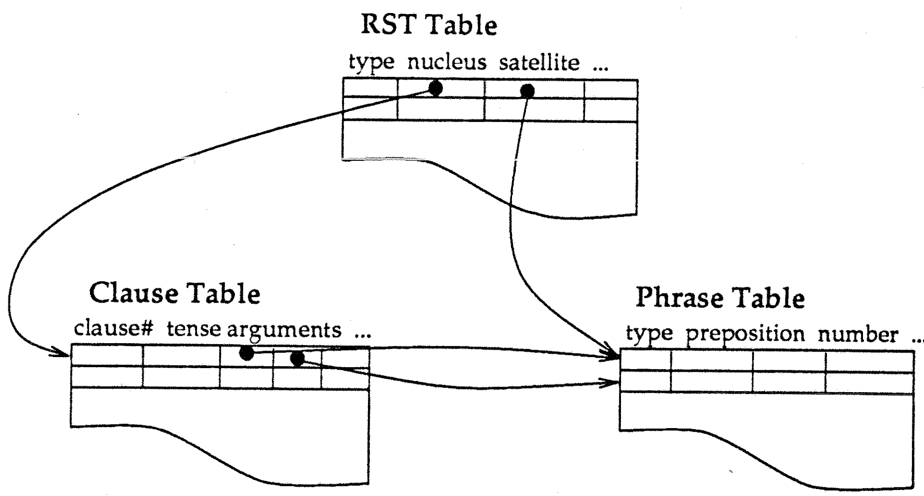

Figure 3.1: The relations between the three levels

3.2 The Structure of the Corpus

The database built for the current study represents information on three different levels, the rhetorical, clausal, and phrasal levels. Figure 3.1 shows the relations that are maintained between the three levels. The Clause table maintains a record for each clause in the database which, among other information, references records in the Phrase table, which detail the clause's direct arguments. The Phrase table also contains NP modifiers such as possessors and prepositional phrases. The RST table contains a record for each node in an RST analysis of the text, containing fields referring to the nucleus and the satellite of the relation. The database system described here and the full instructional text corpus is implemented in FoxBase+/Mac (Fox, 1991).

The following sections detail the nature of each of the fields in the tables, specifying in many cases the valid range of values. In many cases pre-specifying such a range of values is important for effective querying of the database, because the queries must assume a consistent naming scheme. A complete listing of the acceptable values for all the restricted fields in the database can be found in appendix B.

3.2.1 The Clause

The Clause table stores both syntactic and semantic information concerning each clause in the text. One record is stored for each clause. This section details the contents and use of each field. The primary key for the table is the combination of SOURCE and INDEX. A quick reference to all the fields in the Clause table can be found in Table 3.1. A more detailed discussion of each field and its possible values follows directly.

SOURCE is a 3 character string coding the particular source that the clause is from. Including it allows INDEX to be an index to clauses, starting from 1 for each source. This allows the text from the various sources to be removed and edited without affecting the clause numberings for the other sources.

INDEX is the number of the particular clause. This number must allow a fractional component in order to allow the addition of new clauses in a database that has already been built. It is fairly common to come back to a clause and decide that it is really two separate clauses.

PAGE is simply the page number in the source text on which the clause in question starts. It is helpful in finding the clause in the original text and potentially also in identifying page breaks in the text.

LINKER is the lexical item used in the clause to link with another clause in another record. For coordinating clause linkages, the linker is stored with the second clause as the first lexical item in the

Table 3.1: Quick Reference of Clause Table Field Types

Field Name	*Field Type*	*Field Size*	*Description*
SOURCE	Character	003	The source text of the clause
INDEX	Number	008	The clause number within the source text
PAGE	Number	005	The page where the example can be found
LINKER	Character	012	Any clause linker used for this clause
ADVERB	Character	012	Any adverb used
TENSE	Character	003	The Tense of the clause
MODAL	Character	010	Any modal used
HAVE_EN	Character	001	Marks the Have-En form
BE_ING	Character	001	Same for Be-Ing
BE_EN	Character	001	Same for Be-En
VERBSTEM	Character	012	The stem of the main verb
POLARITY	Character	001	The polarity of the predicate
SEMCLASS	Character	006	The semantic class of the predication
REFERENCE	Character	012	The referenced predication
LOOKBACK	Number	005	The computed lookback distance
NOUNHEAD	Number	005	The head NP for relative clauses
CLAUSEHEAD	Number	005	The head clause for hypo- and parataxis
ROLE	Character	006	The semantic role
SLOT	Character	001	The slot of the clause
AGENT	Character	001	The action's agent (reader or non-reader)
ARG_1	Number	005	The Phrase number of the subject
ARG_2	Number	005	The Phrase number of the direct object
ARG_3	Number	005	The Phrase number of the indirect object
TAG	Character	001	A utility field
TEXT	Character	254	The raw text of the clause.
ABSNUM	Number	005	Another utility field

raw text, and as its value for LINKER. Subordinating linkers are stored with the subordinate clause in the same way.

ADVERB simply lists any adverb that is employed in the text. If there are more than one, the database currently lists them in order in this field until the space runs out for the field.

TENSE represents the tense used by the clause. The range of possible values includes simple past, present, and future, "to" infinitive, and imperative. The full list is given in appendix B in table B.2. The value "-" is used when there are incomplete sentences in the source that must be represented somewhere. We don't represent these incomplete sentences as noun phrases or prepositional phrases because they are not grammatically associated with any predication or NP, and we want to be able to determine the order of the text from the Clause table alone.

MODAL represents any modal verbal elements used in the clause.

HAVE_EN, BE_ING, and BE_EN are fields having values of "y" or "n". They code the aspect and voice systems of English. HAVE_EN indicates perfective aspect, BE_ING, progressive aspect, and BE_EN indicates the passive. These three auxiliaries can occur in any combination in the English verb sequence and are thus represented separately.

VERBSTEM represents the base form of the main verb used in the clause.

POLARITY indicates the polarity of the clause. The range of possible values is "A" for affirmative, "N" for negative, and "?" for unknown cases.

SEMCLASS represents the semantic class of the predication. The range of possible values includes material action, verbal action, mental action, and relationals, and is represented in appendix B in Table B.3.

REFERENCE and LOOKBACK are used for referential distance computations. REFERENCE contains a unique string identifying the referent of the predication. In the Clause table, this referent is some event mentioned elsewhere in the text. LOOKBACK is a utility field in which a utility program that computes referential distances will load its result.

NOUNHEAD is used in the Clause table entry for a relative clause to reference the noun phrase being modified. This is an index into the Phrase table. Notice that both the SOURCE and NOUNHEAD values are needed to find the correct Phrase entry. It is also used for other modifying participial clauses and complements.

CLAUSEHEAD is used to represent any kind of coordination or subordination of clauses. Subordinate clauses refer to their head clauses using CLAUSEHEAD and the second clause in a coordinate construction refers to the first in the same way.

ROLE indicates the syntactic role played by a dependent clause. The range of possible values includes adverbial, complement, and conjunction, and is listed in appendix B in table B.4. The notation "-" is used for all clauses that are not relative or conjoined.

SLOT indicates the position of the relative or subordinate clause relative to its head and is left empty for other types of clauses. The range of possible values is "B" for before, "A" for after, and " " for unknown.

ARG_1, ARG_2, and ARG_3 contain the indexes of entries in the Phrase table corresponding to the subject, direct object, and indirect object respectively. If no such syntactic element exists, the field is left blank. Note again that the SOURCE field is needed to find the correct entry in the Phrase table. Note also, that other oblique prepositional phrases are not directly referenced in the head clause record. The Phrase records for such obliques contain references back to the head clause. This is done because there may be any number of them.

TAG contains some arbitrary character indicating any problems that need to addressed later. It does not represent any syntactic or semantic information concerning the clause.

TEXT contains the raw text of the clause directly from the source. It will contain the textual representations of all the related noun phrases and prepositional phrases, which will also appear in the TEXT field of the Phrase table.

ABSNUM is used as a utility field to record a linear count for the clauses of each manual. The INDEX field could be used were it not for the fact that, occasionally, text spans originally identified as single clauses, are broken into two clauses using fractional INDEX numbers, and, likewise, text spans originally identified as multiple clauses, are consolidated into a single clause, thus skipping an INDEX value.

Source: ph1
Index: 15.00
Page: 2

The Clause Table

Linker:	none	Reference:	
Adverb:		Lookback:	0
Tense:	imp	Nounhead:	0.00
Modal:		Clausehead:	0.00
Have_en:	n	Role:	-
Be_ing:	n	Slot:	
Be_en:	n	Arg_1:	0.00
Verbstem:	move	Arg_2:	52.00
Polarity:	A	Arg_3:	0.00
Semclass:	MAT	Tag:	
		Absnum:	16

Text: move the OFF/STBY/TALK [8] Switch to STBY.

Figure 3.2: An example record from the Clause table

Figure 3.2 is an example record from the Clause table for an instructional text database. In this example, a simple imperative mood clause with the verb stem "move" is represented. It is from the source "ph1" indicating the first of the phone manuals. Notice that the ARG_2 slot indicates that the phrase record number 52 is the direct object. This record will be discussed later and is represented in Figure 3.3.

3.2.2 Noun and Prepositional Phrases

The Phrase table stores both syntactic and semantic information concerning noun phrases (NPs) and prepositional phrases (PPs) in the database, one record being stored for each. The table represents NPs and PPs that serve as phrases to predications by referencing the head clause and represents possessors and modifying PPs by referencing the head NP. This section details the contents and use of each field. The primary key for the table is the combination of SOURCE and INDEX. A quick reference to all the fields in the Clause table can be found in Table 3.2 while a more detailed discussion of each field and its possible values follows directly.

SOURCE, as in the Clause table, is a 3 character string coding the particular manual that the phrase is from. It allows INDEX to be an index to phrases in each text that is included in the database, starting from 1 for each source. This allows the text from the various sources to be removed and edited without affecting the phrase numberings for the other sources.

INDEX, as in the Clause table, is the number of the particular phrase. Taken together, SOURCE and INDEX form the primary key on the Phrase table. Although this number allows a fractional component as in the Clause table, it is not used. The numbering of the phrases does not correspond in any way with the numbering of the clauses in which they are included. Thus, new phrases, added to the table at later times, are simply added at the end of the table. The linear order of the source text must, therefore, be determined from the Clause table, not the Phrase table.

SEMCLASS represents the semantic class of the referent of the phrase. The range of possible values

Table 3.2: Quick Reference of Phrase Table Field Types

Field Name	*Field Type*	*Field Size*	*Description*
SOURCE	Character	003	The source text of the phrase
INDEX	Number	008	The phrase number within the source text
SEMCLASS	Character	006	The semantic class of the phrase
TYPE	Character	006	The type of the embedded noun phrase
NUMBER	Character	001	The number of the embedded NP
PREP	Character	012	The preposition used (if any)
DET	Character	005	Any determiner used
QUANT	Character	006	Any quantifiers used
ADJ	Character	012	Any adjectives used
REL_PP	Character	001	Marker for relative clauses or PP modifiers
REFERENCE	Character	014	The referent of the noun phrase
LOOKBACK	Number	005	The distance looked back by the reference
CLAUSE_HD	Number	005	The clause to which this is an phrase
NP_HD	Number	005	The noun head adjectival NPs and PPs
ROLE	Character	006	The role of the phrase
SLOT	Character	001	The slot of the phrase
TAG	Character	001	A field used to mark questions or problems
TEXT	Character	100	The raw text of the phrase

includes human and inanimate, and is represented in appendix B in table B.6.

TYPE specifies the type of the embedded noun phrase, giving both syntactic and semantic information. The range of possible values includes lexical NP, personal name, and personal pronoun and is given in appendix B in Table B.7.

NUMBER gives the number of the noun phrase embedded in the phrase. The range of possible values is singular, plural, mass, and no number.

PREP, DET, QUANT, ADJ indicate the lexical preposition, determiner, quantifier, and adjective(s) used in the phrase, if any. Currently, if these fields are filled the remaining information is not stored, except in the raw text field.

REL_PP indicates whether the phrase has a relative clause or modifying prepositional phrase. The range of possible modifiers is given in appendix B in Table B.9.

REFERENCE and LOOKBACK are used as in the Clause table. REFERENCE contains a unique string identifying the referent of the predication. This list of possible referenced objects must be built for each source type. LOOKBACK is a utility field in which a utility program that computes referential distances will load its result.

CLAUSE_HD is the index number of the clause which contains this phrase.

NP_HD is the index of the phrase which the current phrase modifies. This is used in the case where the phrase is a possessor or an adjectival prepositional phrase. It is also used to indicate the first member of a conjoined or apposed NP in the record for the following ones.

ROLE gives the case role of the phrase. The range of possible values includes subject, object, and indirect object, and is given in appendix B in Table B.10.

SLOT indicates the position of the phrase relative to its head. The range of possible values is " ", "B" for before the predicate, and "A" for after the predicate. " " is only used for incomplete sentences containing no predicate.

TAG is used as in the Clause table. Some arbitrary character is put in TAG to indicate any problems that need to addressed later. It does not represent any syntactic or semantic information concerning the phrase.

TEXT, as in the Clause table, contains the raw text of the phrase directly from the source. The text here will be repeated in the Clause table in the record for the clause head.

Figures 3.3 and 3.4 are example records from the Phrase table for an instructional text database. The

The Phrase Table

Source:	ph1
Index:	52.00

Semclass:	INAN	Reference:	off/stby/talk
Type:	LEX	Lookback:	11
Number:	S	Clause_hd:	15.00
Prep:		Np_hd:	0.00
Det:	the	Role:	O
Quant:		Slot:	A
Adj:		Tag:	
Rel_pp:			
Text:	the OFF/STBY/TALK [8] Switch		

Figure 3.3: An example NP record from the Phrase table

first example shows how a noun phrase is represented. It is denoted as a singular inanimate object expressed as a noun phrase with a lexical head. The REL_PP field is blank because it does not apply. CLAUSE_HD is a reference back to the Clause table record discussed above and shown in figure 3.2. The second example shows how a prepositional phrase is represented. Here we have the preposition field filled with the preposition "to." This record, as in the previous example, refers the clause record discussed earlier as it is a separate phrase to that clause.

3.2.3 Rhetorical Structure of the Text

The RST table stores information concerning the rhetorical structure of the database. It is an implementation of Mann and Thompson's Rhetorical Structure Theory (Mann and Thompson, 1989). One record is stored for each node of the RST tree for the given text.

This section details the contents and use of each field. The primary key for the RST table is the combination of SOURCE, SYNTAX_IDX, and SYNTAX_CD. A quick reference to all the fields in the Clause table can be found in Table 3.3. A more detailed discussion of each field and its possible values follows directly.

SOURCE is used as in the Clause and Phrase tables. The source text is indicated in the SOURCE field.

SYNTAX_IDX, CHILD, and NUCLEUS: The RST table uses a single record type to represent both parent child relationships (called JOINT schemas in RST) and nucleus-satellite relationships. SYNTAX_IDX is the index of the clause or phrase corresponding to parent or satellite respectively. CHILD and NUCLEUS are the indexes of either the child or the nucleus involved. Only one of these is filled in for any record.

SYNTAX_CD, CHILD_CD, and NUCLEUS_CD: Because we allow both clauses and phrases to enter into rhetorical relations, an additional field, SYNTAX_CD is included to indicate which table the SYNTAX_IDX refers to, the Clause or the Phrase tables. Additionally, SYNTAX_CD can indicate that the node is a node internal to the RST tree which is associated with neither an phrase or a clause.

Source: ph1

Index: 53.00

The Phrase Table

Semclass:	ABS	Reference:	stby
Type:	LEX	Lookback:	-1
Number:	S	Clause_hd:	15.00
Prep:	to	Np_hd:	0.00
Det:		Role:	Loc
Quant:		Slot:	A
Adj:		Tag:	
Rel_pp:			

Text: to STBY

Figure 3.4: An example PP record from the Phrase table

Table 3.3: Quick reference of RST table field types

Field Name	*Field Type*	*Field Size*	*Description*
SOURCE	Character	003	The source text of the phrase
SYNTAX_IDX	Number	008	The current clause or phrase
SYNTAX_CD	Character	001	The type of that entry
CHILD	Number	008	The index of the child of the source
CHILD_CD	Character	001	The type of the child
RELATION	Character	012	The type of RST relation
NUCLEUS	Number	008	The nucleus of the relation
NUCLEUS_CD	Character	001	The type of the nucleus

Similarly, the child and nucleus indexes require a code. The range of possible values for SYNTAX_CD, CHILD_CD, and NUCLEUS_CD allows the representation of internal, leaf clause, and leaf phrase.

RELATION indicates the type of RST relation that pertains between the nucleus and satellite or the parent and the child. The range of possible values are taken from Mann and Thompson's work, but can be augmented by more specific relations that prove useful in the particular domain of interest. Those values that have proved useful for the instructional text type includes the various relations taken from RST and modified for instructional text. They include precondition, purpose, and result. The full list is given in appendix B in Table B.13.

An example of the representation is probably the best way to see how the fields in this table are used. We will show a sample analysis and representation of the installed-phone text, repeated here:

Installed-Phone Text:

> When the 7010 is installed and the battery has charged for twelve hours, move the OFF/-STBY/TALK [8] Switch to STBY. The 7010 is now ready to use. Fully extend the base antenna [12]. Extend the handset antenna [1] for telephone conversations.

To begin with we segment the text into clauses as follows:

1. When the 7010 is installed
2. and the battery has charged for twelve hours,
3. move the OFF/STBY/TALK [8] Switch to STBY.
4. The 7010 is now ready to use.
5. Fully extend the base antenna [12].
6. Extend the handset antenna [1] for telephone conversations.

The RST structure is for the installed-phone text (copied from Chapter 1) is presented in figure 3.5. Note that it includes a reference to each of the text spans given above and also a reference to the phrase "for telephone conversations" which is labeled as P1[3] in the figure. The representation in the RST table for this tree is given in table 3.4. The source field simply indicates that the RST fields refer to text from the first phone manual, coded as "ph1" (Code-a-phone, 1989), so it has been left out of the diagram. **Syntax_Cd** and **Syntax_Idx** give the code and index of the text span associated with the record. The remainder of the fields allow the specification of nucleus-satellite relations and joint schemas.

Nucleus-Satellite relations are represented by allowing the satellite to refer to the nucleus. This allows any number of satellites to refer to a nucleus. During the analysis for the current study, this feature was used to describe texts (as was also the case in Mann and Thompson 1989), but IMAGENE, as will be seen in Chapter 4, only produces a single satellite for each nucleus, subordinating further relations in the RST hierarchy. This was done to simplify the generation process. The type of relation is indicated in the **Relation** field. For example, the span C4 is a satellite of the span I3 of type result. Thus, the record for node C4 refers to I3 and indicates the relation type, not the other way around. I3, in turn, refers to its child node, C3.

Joint schemas are represented by specifying only the first child in the parent node, and specifying the sequence of children nodes using the **Nucleus** and **Nucleus_Code** fields. Thus, these two fields are used for two purposes, depending upon the relation type. In multi-nuclear relations (i.e., sequence and joint), the nucleus fields refer to the sequence of children, in single-nuclear relations, they refer to the nucleus. Because of the representation scheme, internal nodes are the only nodes with children. Clauses and phrases are always at the root. For example, the span I4 has a precondition relation with it nucleus C3 and a parent-child relationship with spans C1 and C2. This is represented in the database

[3]The notation "P1" indicates that this is the first record in the Phrase table for this particular source text. "C1" indicates the first record in the Clause table for this source. These codes are represented in the RST table using the index and code field pairs, one for the central node of the record, one for the first child of that node, and one for the nucleus of that node, called SYNTAX_IDX and SYNTAX_CD, CHILD and CHILD_CD, NUCLEUS_IDX and NUCLEUS_CD respectively.

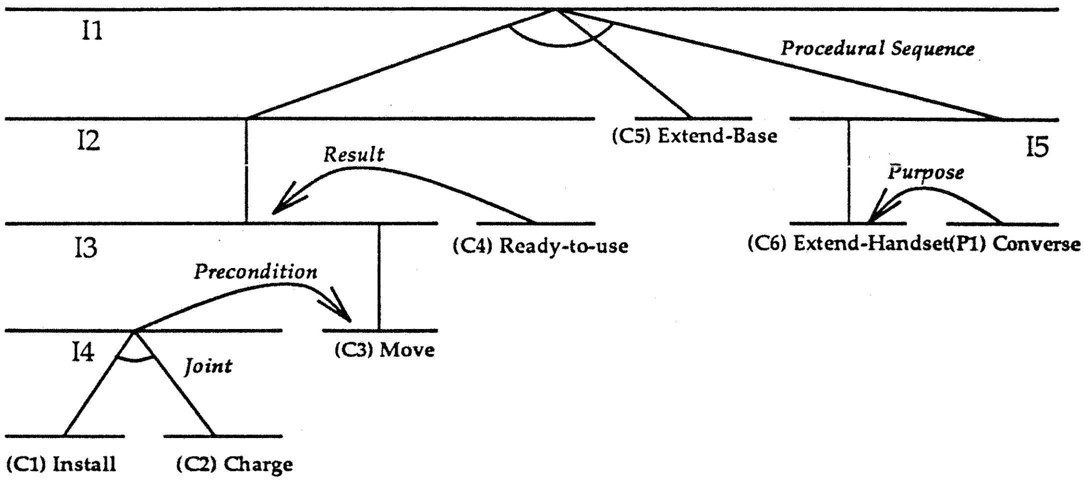

Figure 3.5: The RST analysis for the Installed-Phone Text

Table 3.4: The RST table representation of the example RST structure

Syntax_Cd	Syntax_Idx	Child_Cd	Child	Relation	Nucleus_Cd	Nucleus
C	1			Joint	C	2
C	2			Joint		
C	3					
C	4			Result	I	3
C	5			Sequence	I	5
C	6					
P	1			Purpose	C	6
I	1	I	2			
I	2	I	3	Sequence	C	5
I	3	C	3			
I	4	C	1	Precond	C	3
I	5	C	6	Sequence		

by specifying the first of its children, C1, as a child node, and specifying node C3 as its nucleus with a precondition relation. The precise nature of the parent-child relationship is indicated by the child node's relation-type. In this case node C1 indicates that it is a sequence node with a trailing node C2. The final child node of I4 is still marked as a sequence node, but has not value in the nucleus fields.

3.3 Using the Corpus

The whole point of designing and implementing such a corpus is to provide facility for doing functional research on instructional text, particularly on discourse structures. The database represents a number of the possible functional explanations of various discourse phenomena, the problem, now, is to find them. This section gives an approximation of how the entire corpus would be used to determine the appropriate slot for purpose expressions.[4]

The first task is to query the database for all possible purpose expressions. This is done, at least in the FOX implementation of the database with a simple join statement over the RST table and either the

[4]As has been noted only the training portion of the corpus was actually used to perform this analysis.

Source	Index	Slot	Text
aph	2.00	B	1. To release handset
aph	19.00	B	To place additional calls
aph	25.00	A	to end billing cycle.
aph	26.00		Replace handset to seatback
aph	29.00	A	to lock.
aph	39.00		remove phone
aph	43.00	A	to place calls.
aph	46.00	A	To place call
aph	48.00	B	To end call
aph	50.00	B	To place additional calls
aph	52.00		5. Return handset to wall unit from which it was taken.
ar1	8.00	A	to catch the fluid.
ar1	12.00	A	so that the fluid drains out.
ar1	14.00	A	so that it does not bend.
ar2	1.00	B	To repair a cut in a vinyl roof
ar2	5.00	A	so that no cement oozes onto the surface of the vinyl.
ar2	9.00	B	To eliminate a bubble in a vinyl top
ar3	1.00	B	1. To relieve steam pressure
ar3	18.00	A	to remove it.
ar3	23.00	A	to adjust mixture as soon as possible.
bpl	5.00	B	TO MAKE BUBBLES
bpl	13.00	A	to make big bubbles.
bpl	14.00	B	To make more bubbles
bpl	19.00	A	to begin
bpl	20.00		TO CLEAN
...	...	...	...

Figure 3.6: The First 25 Purpose Clauses

Clause or the Phrase tables. This operation was so common that it was implemented as a general utility for finding all the expressions of a certain rhetorical relation. The core of the retrieval is the following relational join command:

```
Join with RST to TEMP for (rst->relation = relation).and.
                          (source = rst->source).and.
                          (((index = rst->syntax_idx).and.
                            (rst->syntax_cd = code)).or.
                           ((rst->syntax_cd = "I").and.
                            (index = rst->child).and.
                            (rst->child_cd = code)))
```

The point of the command is to return all of the expressions that appear as satellites in nucleus-satellite relations with simple or complex nuclei. Either the clause or phrase expressions are returned, depending upon whether the clause or the phrase table has been selected as the current table and on whether the value of code is set to either "C" or "P". When this command is executed with the current corpus, it returns 104 purpose clauses, the first 25 of which are shown in Figure 3.6 and the 22 purpose phrases shown in Figure 3.7.

With respect to the determination of the slot for the purpose expressions, we can see from this data that 28.6% (36) of the purpose expressions are fronted, 47.6% (60) are not fronted, and 23.8% (30) are unmarked (because they are not subordinate clauses). Of the 30 examples that were not marked, 10 were fronted "by" purposes, and another 15 were other forms of fronted expressions, bringing the combined totals for clauses and phrases to 48.4% (61) fronted and 51.6% (65) non-fronted. These values

Source	Index	Slot	Text
aph	48.00	A	for instructions
aph	70.00	A	for instructions
bpl	2.00	A	FOR FUTURE USE
bpl	3.00	B	For future reference
fa2	1.00	A	for seizures in children with high fever
jp1	91.00	A	For a special gift
jp2	15.00	A	For a harder puzzle
ph1	5.00	A	for desk installation
ph1	6.00	A	for more information
ph1	8.00	A	for telephone conversations
ph1	13.00	A	for computer transactions.
ph1	15.00	A	for computer services or custom calling services that ...
ph1	16.00	A	for charging the batteries.
ph1	20.50	A	with telephone services like Call Waiting or Call Transfer.
ph1	26.00	B	For frequently busy numbers
ph1	29.00	A	for paging.
ph1	31.00	A	for recharging.
ph1	60.00		For pulse or rotary dial systems.
ph1	62.00		For touchtone systems.
tv1	47.00	A	for each available VHF channel.
tv1	57.00	A	for each available channel
tv1	59.00	B	For picture adjustment of BRIGHTNESS

Figure 3.7: Purpose Phrases

closely match the percentages of initial and final purpose clauses found by Thompson for procedural texts (1985). Although she reports just over 18% initial purpose clauses for English text in general, she reports 49% initial purpose clauses for a book of recipes and 34% for an auto-repair manual.

Initial hypotheses concerning this distribution can be derived from two sources, either an informal perusal of the output of these commands or from a knowledge of the functional linguistic literature. Using the existing functional literature is the best way to start, provided that the particular issue in question has already been addressed. The informal perusal is a daunting task, and was attempted only in the absence of other research. In the case of purpose expression slot, Thompson's study is a good source. As seen in Chapter 2, her corpus included some procedural texts, and indicated that one common feature of fronted purpose clauses is that their scope is global, that is, there is more than one expressed proposition that is directly related to the fulfillment of the purpose. This provides a good starting hypothesis for determining the slot of purposes, namely that global purpose clauses are fronted.

This hypothesis is relatively easy to confirm or disconfirm in the corpus. The global purpose expressions can be obtained by executing the following join command on the table of purpose clauses:

```
Join with RST to test1 for (rst->relation = "purpose").and.
                           (source = rst->source).and.
                           (rst->nucleus\_cd = "I").and.
                           (index = rst->syntax_idx)
```

This command causes the database to look through the purpose expressions table created above, finding those purpose expressions that have a complex nucleus node (indicated by the "I" code). The command results in the purpose clauses shown in figure 3.8 and the purpose phrases shown in figure 3.9.

On close inspection of these two tables we find that 50.0% (19) of the global purposes (38) are explicitly marked as fronted, 34.2% (12) of them are unmarked, but global nonetheless (as noted above,

Source	Index	Slot	Text
aph	2.00	B	1. To release handset
aph	19.00	B	To place additional calls
aph	26.00		Replace handset to seatback
aph	52.00		5. Return handset to wall unit from which it was taken.
ar2	1.00	B	To repair a cut in a vinyl roof
ar2	9.00	B	To eliminate a bubble in a vinyl top
ar3	1.00	B	1. To relieve steam pressure
bpl	5.00	B	TO MAKE BUBBLES
bpl	19.00	A	to begin
bpl	20.00		TO CLEAN
bpl	26.00	B	For proper storage
bpl	30.00		FOR BETTER BUBBLE BLOWING
bpl	33.00	B	2. To improve bubble blowing performance of the plane:
crl	16.00		The correct time can usually be obtained
crl	41.00		The radio or radio/buzzer alarm can be set to any ...
fa1	9.00	B	To prevent shock
fa3	4.00		turn off the electric current
fa4	1.00	B	To control bleeding in the mouth
hr	16.00	A	so that it billows black smoke for a while.
jp1	10.00		the puzzle can be bisected
jp1	25.00	A	so you don't end up with one abnormally small or ...
jp1	28.00	A	to avoid losing them.
jp1	33.00		check the back of the pieces for burrs
jp2	9.00	A	so that the small burr that does occur will not signal ...
ms1	13.00		So you always know where you were - and where ...
ph1	34.00	A	to return to PULSE dialing mode.
ph1	67.50	B	1. To end a previous call
ph1	83.00	B	To avoid
ph2	6.00	B	To get the most from your battery
ph2	9.00		To drain or discharge the battery:
ph2	17.00		To recharge the battery:
ph2	79.00		Your call may be ended in any of the following manners:
tv1	4.00	B	To turn on the power
tv1	23.00	B	To compensate for this
tv1	26.00	B	To preset fine tuning of VHF channel
wp	2.00	B	To operate the 'Waste King Pulverator
wp	10.00	B	To stop the 'Waste King Pulverator':

Figure 3.8: Global Purpose Clauses

Source	Index	Slot	Text
jp2	15.00	B	For a harder puzzle

Figure 3.9: Global Purpose Phrases

non-subordinate clause purposes such as "by" purposes), and 15.8% (7) are not fronted. Of these non-fronted cases, 4 are actually not global, being retrieved here because they took part in some other form of complex nucleus-satellite RST schema, just not one that was global. The remaining 3 cases (7.9%) are legitimate counter-examples to the hypothesis. The small percentage of counter examples to the hypothesis provides good evidence that it would be a good one to use in the system. Furthermore, it was discovered in later analysis that the form of some of the counter-examples was motivated by the use of a "so that" expression, which is used only in the final position.

Thompson's paper goes into a fair amount of detail concerning the precise nature of the discourse function being served by the fronted purpose expression. In the current study, the higher-level feature of scope has been sufficient to achieve a good coverage of the text. The current study has tended to use the simplest functional feature that explains a good percentage of the corpus.

The next question to be asked is if this is the only explanation of fronted purpose expressions, or if there are other fronted purpose expressions that are not global. This can be addressed by querying all the fronted purpose clauses in the corpus using the following command:

```
Join with RST to test1 for (rst->relation = "purpose").and.
                           (source = rst->source).and.
                           (index = rst->syntax_idx).and.
                           (slot = "B")
```

This command yields the fronted purpose clauses shown in Figure 3.10 and a similar command yields the purpose phrases shown in Figure 3.11.

After removing the global purposes, it is clear that these tables contain a number of records that are not global. This leads to the conclusion that there are other factors at work in the question of when to front a purpose expression. The iterative process of hypothesis generation and testing can then be conducted on these other cases in a similar manner. Full implementation of the analysis of purpose expression slot is given in Chapter 6.

Concerning the amount of iteration that is appropriate in this process, the obvious goal of the process is to identify features of the communicative context that divide the action expressions in the corpus into groups of expressions with homogeneous forms, at least with respect to the lexical and grammatical elements of form that are of interest the study at hand. This, however, is not always possible. The analyst may well be confronted with a situation where a set of heterogeneous forms exist for which there is no apparent feature that correlates with the differences in form. This is not surprising, given that some of the features of the communicative context are inaccessible to the analyst, who has only the text to work from.

To these two obvious stopping points, another must be added. When analyzing the training portion of the corpus, a balance must be drawn between explaining as many of the examples as possible on the one hand, and preparing the system network to deal with the examples in the testing corpus on the other. Clearly, adding systems that address single examples in the testing corpus will probably do more harm than good in the testing corpus. Thus, a general heuristic was followed in the current study, repeating the analysis cycle until further analysis served to only explain groups of around 2 or 3 examples or less, at which time the iteration was discontinued.[5]

3.4 Summary

This chapter has given a considerable amount of detail concerning the nature of the instructional text in the corpus, and the structures with which it is represented. It concluded with an example of the iterative process by which the functional features of the communicative context most important to the determination of expressional form were derived. The results of this study serve as the basis for the features encoded in the IMAGENE text generation system, described in Chapters 4 through 6. The corpus

[5]Occasionally, systems which addressed 2 or fewer examples were included if they appeared to be important observations that would scale up. In the current study, a few systems were included for this reason, but then removed later if it was clear that they didn't scale.

Source	Index	Slot	Text
aph	2.00	B	1. To release handset
aph	19.00	B	To place additional calls
aph	48.00	B	To end call
aph	48.00	B	To end call
aph	50.00	B	To place additional calls:
ar2	1.00	B	To repair a cut in a vinyl roof
ar2	9.00	B	To eliminate a bubble in a vinyl top
ar3	1.00	B	1. To relieve steam pressure
bpl	5.00	B	TO MAKE BUBBLES
bpl	14.00	B	To make more bubbles
bpl	26.00	B	For proper storage
bpl	33.00	B	2. To improve bubble blowing performance of the plane:
cr1	2.00	B	To set the correct time on the Clock Display
cr1	10.00	B	* To forward the displayed time hour (minute) by hour...
cr1	28.00	B	5. To turn the radio off
cr1	33.00	B	To obtain the best FM reception,
cr1	38.00	B	To improve the reception of weak stations
cr1	54.00	B	* To confirm the alarm time
cr1	62.00	B	5. To shut off the radio alarm before automatic shut-off
cr1	71.00	B	5. To shut off the radio/buzzer alarm before automatic shut-off
fa1	9.00	B	To prevent shock
fa4	1.00	B	To control bleeding in the mouth
ph1	67.50	B	1. To end a previous call
ph1	83.00	B	To avoid
ph2	6.00	B	To get the most from your battery
ph2	32.00	B	In order to prevent
ph2	35.00	B	To insure the long life of your battery
tv1	2.00	B	For switching the TV ON or OFF
tv1	4.00	B	To turn on the power
tv1	17.00	B	NOTE: To adjust preset fine tuning
tv1	23.00	B	To compensate for this
tv1	26.00	B	To preset fine tuning of VHF channel,
wp	2.00	B	To operate the 'Waste King Pulverator'
wp	10.00	B	To stop the 'Waste King Pulverator':

Figure 3.10: Fronted Purpose Clauses

Source	Index	Slot	Text
bpl	3.00	B	For future reference.
ph1	26.00	B	For frequently busy numbers,
tv1	59.00	B	For picture adjustment of BRIGHTNESS

Figure 3.11: Fronted Purpose Phrases

itself is the rule against which the predictions of IMAGENE's text grammar are evaluated, as reported in Chapter 7.

Chapter 4

IMAGENE

This chapter provides the technical specifications for IMAGENE that are necessary for understanding the specific details of the system networks to be discussed in Chapters 5 and 6. It discusses the input to IMAGENE, represented as a *Process Representation Language* (PRL) structure, the gradual construction and manipulation of the textual representation of the rhetorical and grammatical structure of the text, represented as a *Text Representation Language* (TRL) structure, and finally, how the TRL structure is translated into *Sentence Planning Language* (SPL) statements for generation by Penman.

After discussing a few preliminaries concerning the implementation and notation, this chapter will give an overview of IMAGENE's architecture and proceed to discuss each part in some detail. To make the discussion more concrete, the Remove-Phone text, repeated here, will be used as a running example:

Remove-Phone Text:

> When instructed (approx. 10 sec.) remove phone by firmly grasping top of handset and pulling out. Return to seat to place calls.

4.1 Implementation and Notation

IMAGENE has been implemented in Macintosh Common Lisp (MCL, 1989) using many of the Lisp tools available in the MacPenman Distribution from the University of Southern California's Information Sciences Institute, including Penman itself (Mann, 1985; Penman, 1989) and Loom (Brill, 1992; LOOM, 1991). Specifically, TRL and PRL have been implemented in Loom, while the SPL command language and the system network implementation are provided directly in MacPenman.

Because the system network serves as a notation for the fundamental results of the current study, a note is appropriate here concerning the notation with which they will be represented in this thesis. System networks are basically decision networks read from left to right, employing the four basic notations shown in Fig. 4.1. The standard notation for a simple system is shown in Figure 4.1A, where the name of the system is given on the left and available output features are given on the right. Only one of the of the features may be chosen, based upon a possibly complex set of inquiries concerning the functional context of the utterance. The structure of the inquiries will not be shown. Figure 4.1B shows a notation indicating that **System-A** and **System-B** are to be entered in parallel. Figure 4.1C and 4.1D show complex entry conditions for single systems. The former indicates an OR-gate which is passed when either X or Y is true (or both), and the latter an AND-gate which is passed only when X and Y are both true. A more detailed discussion of these formalisms has been given by Winograd (1983) and Hudson (1971). They are very similar to the Artificial Intelligence construct of the Production System (Post, 1943) which has also been used in text generation (Kukich, 1983; Kukich, 1988).

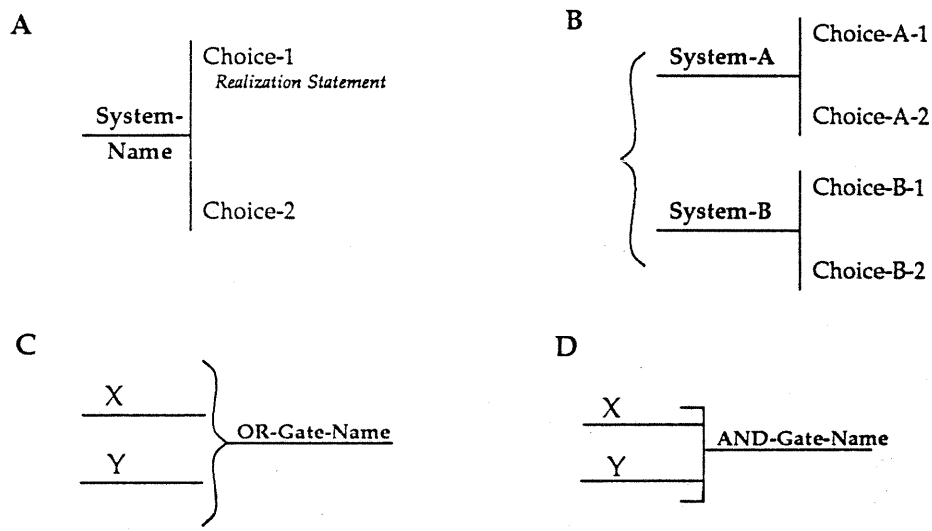

Figure 4.1: Notational Forms for System Networks

4.2 Overview of the Architecture

Chapter 2 discussed the relation of the methodology used in the current study with other related work. The relation of IMAGENE's architecture to other text generation architectures, however, was not covered in any detail. This section will place IMAGENE in the taxonomy of generation architectures developed by Bateman and Hovy (Bateman and Hovy, 1991) in order to introduce its basic components and to illustrate its contribution to the field.

Bateman and Hovy have identified three parallel dimensions of description along which any architecture may be related to other work. These dimensions are:

1. *Generator Task* — To what general questions does the generation system address itself? There are three alternatives usually listed here, all of which must be dealt with in some way, but typically only one is addressed in any detail:
   (a) Grammatical Realization
   (b) Content Selection
   (c) Motivation for Decisions
2. *Method of Generation* — How does the system accomplish these tasks? Bateman and Hovy identify four basic approaches:
   (a) Canned Items
   (b) Templates
   (c) Cascaded items
   (d) Features
3. *Register* — How does the generation system address the three elements of communicative context defined in systemic linguistics? These aspects, discussed in Chapter 2, are listed here:
   (a) Field
   (b) Tenor
   (c) Mode

The basic task to which IMAGENE addresses itself is the task of motivating decisions in the communicative context of instructions. The problem of grammatical realization is left to Penman. The problem of content selection is left to the application program, assumed to be a procedural planner in the current study. In Bateman and Hovy's terms, this use of external means to determine the content and basic structure of the text is called a "canned" paragraph structure. It is commonly used in cases where the text displays considerable conformity to a non-linguistic structure of some sort (Kittredge and Polguère, 1991). This is similar in spirit with the Paris and McKeown's use of the structure of a knowledgebase to structure descriptions of complex objects (1987).

The procedural input to IMAGENE is assumed have been built by a procedural planner, allowing the current study to address the equally important problem of expressing the plan in text. This allows IMAGENE to fit into the broader context of an architecture that includes a procedural planner and a user model which determine the content of the instructional text. This architecture is very much in line with the proposal of using multi-lingual generation as an alternative to machine translation (Rösner and Stede, 1992b; Kittredge and Polguère, 1991).

IMAGENE employs a feature-based mechanism for text generation very similar to the one Penman employs for sentence generation. The precise nature of the features relevant for the construction of instructional texts is the subject of Chapters 5 and 6.

The study of register is at the heart of the current study. The entire methodology is designed to gain insight into the characteristics of the instructional register, and to code these insights in an automated generation system. IMAGENE deals with any of the aspects of register, termed field, tenor, and mode, that covary with the expressional forms used in instructional text.

IMAGENE's architecture is shown in Figure 4.2. There are two fundamental inputs to IMAGENE: (1) the structure of the process being described, called the *Process Structure*, and (2) the responses to a set of *text-level inquiries*, analogous to Penman's sentence-level inquiries. The Process Structure will be discussed in some detail in a later section.

The inquiries are made by a single system network, where each system potentially constructs or manipulates a particular aspect of the text structure being built. There are three basic tasks accomplished by this network:

- Constructing the basic hierarchical structure upon which the text will be based;
- Determining both the rhetorical status and the grammatical form of the action expressions;
- Combining the various clauses and phrases into sentences.

The text structure being built by these networks is a single, evolving entity represented in the text representation language (TRL) and manipulated by a set of *realization statements*. This section will discuss the process structure, the text representation, how it is manipulated by the realization statements associated with the systems in the network, and the SPL builder.

4.3 The Process Representation Language

As mentioned above, one of the inputs to IMAGENE is the *Process Structure*, a representation of the process to be expressed. The current study takes a traditional AI view of the representation of processes and actions that is similar to the one taken by Mellish and Evans (1989), as based on STRIPS-style planning (Sacerdoti, 1973). It assumes that processes can be represented with discrete action nodes arranged in a hierarchy, where actions at the higher levels represent general actions and those at the lower levels detail the sub-processes involved. These plans have also been extended to deal with concurrent actions, another aspect of processes important for the current study (Sacerdoti, 1977; Tate, 1976; Allen, 1984).

The current study takes the actions from a plan, one which would result from a procedural planner, and determines how to express them depending upon the communicative context. Some actions will be expressed as sequential commands, while others may be expressed as preconditions, results or purposes. The preconditions and postconditions that are attached to the plan operators that allow the planner to reason about the actions do not affect this process. They can be used as a resource to identify

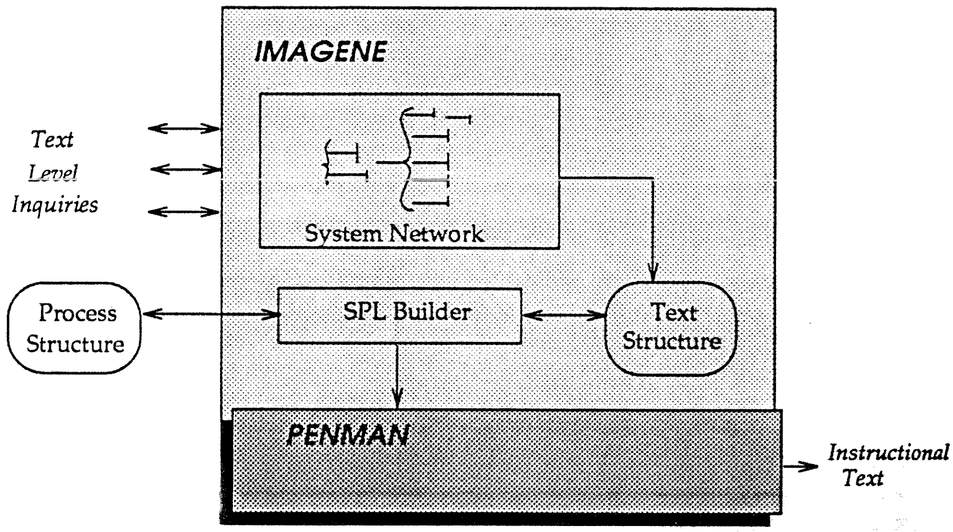

Figure 4.2: The Architecture of IMAGENE.

the relevant state that results from an action, but not to determine the rhetorical status of an action. The rhetorical status is determined from the features of the action itself and from other features of the larger communicative context.

The Process Representation Language (PRL) provides a method of representing processes that takes all of the factors just mentioned into account. It allows the representation of actions in a hierarchy and provides facility for representing concurrency. IMAGENE's system networks do not actually access the PRL structure during generation, rather they determine the relevant features of the process through the execution of a number of textual inquiries. This makes the network and the generation process independent of the particular plan representation method used. The process structure does, however, provide one important source of information to the SPL Builder, that of the linguistic forms of the various processes and objects represented in the process being expressed. The problem of choosing among the various lexical items and case structures that might be appropriate for the expression of a particular action is an interesting one, but beyond the scope of the current study. The current study focuses on expressing the procedural relationships between actions using the rhetorical tools provided by English.

The attributes, represented as slots, that are represented for each of the actions in the network are as follows:

Action-Type — The particular lexical item in the Loom represented domain model for instructional text that represents the action being performed. The point of the current study was to determine, among other things, the morphological form of the verb, not the lexical item itself.

Actor — The PRL entity which represents the actor or agent of the current action. This an important deviation from standard STRIPS-style planning, which assumes a single agent in the world. Instructional text frequently expresses actions performed by other agents as effective actions rather than as states. Such non-reader agents can include the device itself or service personnel. It was, thus, important to represent these as actions, rather than as simple pre-conditions or post-conditions attached to reader actions. Figure 4.3, for example, which will be discussed shortly, includes the **Instruct-Action** which is an action by a non-reader agent.

Actee — The PRL entity which represents the object acted upon by the actor.

Destination — The PRL entity which represents the destination of a moving action.

Duration — The natural number representing the number of duration units that an action is to take.

Duration-Units — The lexical item in the domain model for instructional text that represents the units in which the duration is coded.

Instrument — The PRL entity which represents the instrument used in the action.

The PRL representation for each action references the objects that fill the case-roles for that action. These objects are represented as separate PRL entities. The attributes that are represented for each of them are as follows:

Object-Type — The particular lexical item in the domain model represents the object being referred to.

Object-Number — Whether the object is singular or plural.

Object-Status — Whether the object is associated with the device being described or with another entity. This distinction is important for the model of referring expressions with produces "the" for device objects and "a" or "an" for non-device entities. This treatment of referring expressions is a simplified mechanism that is not based on a corpus study of referring expressions.

Object-Location — The PRL entity representing the location of the object being referred to, if it is important to express this.

Object-Possessor — The PRL entity representing the owner of the object being referred to, if it is important to express this.

Object-Property — The PRL entity representing other properties of the object being referred to, if it is important to express them.

These attributes for actions and objects are a largely ad hoc list of features that were useful in addressing elements of the expression of actions that were beyond the scope of the current study, issues such as the particular verb used to express an action, or the particular noun to use for units of duration. There are, no doubt, other attributes that would have to be represented to build a full treatment of instructional text, but these have sufficed for the purposes of the current study. This simplification allowed the current study to focus on the issue of expressing procedural relations using rhetorical relations in instructional text.

Another important simplification employed by the current study is that no planner was implemented and integrated into the system. The PRL structures used in IMAGENE are built by hand. This considerable simplification has allowed the current study to focus on determining the features of the communicative context that dictate forms of expression rather than on planning.

The process structure built for the Remove-Phone example is shown graphically in Figure 4.3 and textually in figures 4.4 and 4.5. It represents a single procedural root action, called ***prl-root***, which has three sub-actions, **instruct-action**, **remove-action**, and **place-action**. **Remove-action** is further divided into two sub-actions, **grasp-action** and **pull-action**. **Place-action** has a single sub-action, **return-action**. Notice that the lexical items for all of the actions and their arguments are explicitly given in this structure. IMAGENE's task is to take them and produce rhetorical, lexical, and grammatical forms appropriate for expressing the relations between these actions. Notice also that Figure 4.3 only represents the actions and not the objects. The actions are represented in Figure 4.4 and the objects in Figure 4.5.

4.4 Text Representation and Manipulation

The purpose of the systems in the network is basically to build a linguistic structure. Each system is capable of adding to or changing the representation of the structure using *realization statements*. This process requires two important elements: (1) a representation for the linguistic structure (in the current study, the rhetorical and grammatical structure of the text), and (2) a set of realization statements that build and manipulate this structure in appropriate ways. This section will discuss both of these elements.

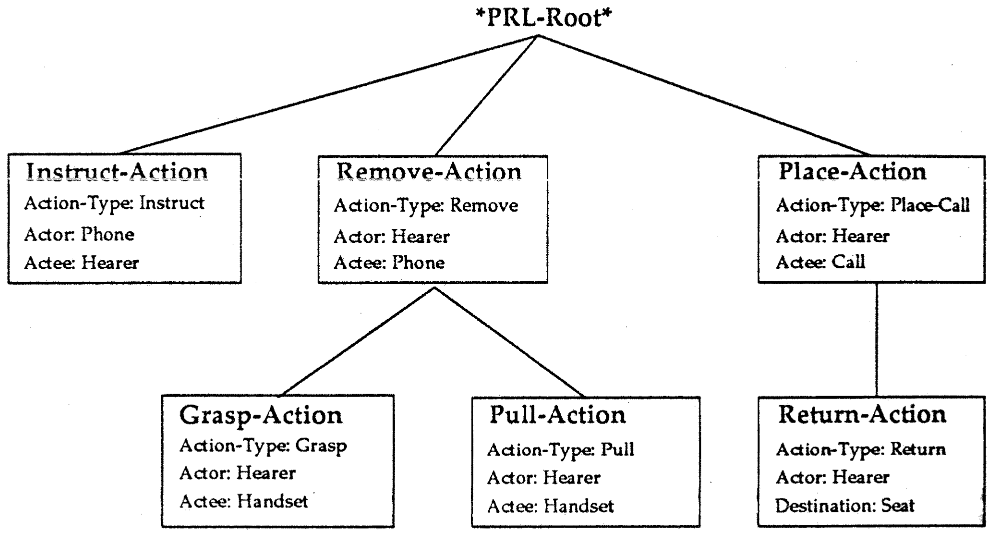

Figure 4.3: A Graphical View of Remove-Phone PRL Structure

4.4.1 The Text Representation Language

The Text Representation Language (TRL) has been designed to represent all of the major rhetorical, lexical, and grammatical forms found in the corpus study of instructional text, as well as their textual order and clause combining. The rhetorical structure is represented hierarchically, as an RST tree. The RST relations and schemas that were most useful for instructional text are: (1) Procedural sequence, represented using joint RST schemas; (2) Precondition; (3) Purpose; and (4) Result. These rhetorical relations are strongly based on the procedural relationships commonly expressed in instructional text.

These relations are not all taken directly from Mann and Thompson's list of relations, but have been specifically tailored for the task at hand, being based on the procedural relationships found in typical plan representations. Precondition is a consolidation of what Mann and Thompson call Volitional Cause, Non-Volitional Cause, and Condition. Result is a consolidation of Volitional and Non-Volitional Result. The distinctions made by Mann and Thompson are not ignored; rather, IMAGENE uses the system networks to make much more fine-grained distinctions for the purpose of expression.

Each of the expressible nodes of the RST tree are further marked with the following lexical and grammatical features:

- Form (imperative, "to" infinitive (TNF), nominalization, participle, goal-metonymy, passive, action, relational, and is-required);
- Linker (then, for, when, and, if, only-if, after, make-sure, and with);
- Tense (present, past, future, and present-perfect);
- Actor (which, that);
- Modal (should).

Not all of these features are required for every node. The list would certainly not be complete for all forms of text, but has supported the needs of the domain of instructional text addressed in the current study.

TRL also includes a separate "trace" of the specified order of expression of the actions, represented by a linear path of pointers, starting with the root of the structure and proceeding through each expressible node to the end. Each pointer is marked as either *new-sentence* or *continue-sentence*, specifying which clauses and phrases are combined into a single sentence, and which are to be separated.

```
(tell (:about *PRL-Root* Action
              (subaction instruct-action)
              (subaction remove-action)
              (subaction place-action)))

(tell (:about Instruct-Action Action
              (action-type it::Instruct)
              (actor phone)
              (actee hearer)))

(tell (:about Remove-Action Action
              (subaction grasp-action)
              (subaction pull-action)
              (action-type it::Remove)
              (actor hearer)
              (actee phone)))

(tell (:about Grasp-Action Action
              (action-type it::Grasp)
              (actor hearer)
              (actee top)))

(tell (:about Pull-Action Action
              (action-type it::Pull)
              (actor hearer)
              (actee handset)))

(tell (:about Place-Action Action
              (subaction return-action)
              (action-type it::Place-Call)
              (actor hearer)
              (actee call)))

(tell (:about Return-Action Action
              (action-type it::return-movement)
              (actor hearer)
              (destination seat)))
```

Figure 4.4: The PRL Action Commands for the Remove-Phone Text

```
(tell (:about Phone Object
              (object-type it::phone)
              (object-status device)))

(tell (:about Hearer Object
              (object-type it::hearer)))

(tell (:about Top Object
              (object-type it::top)
              (object-status device)
              (object-possessor handset)))

(tell (:about Handset Object
              (object-type it::handset)
              (object-status device)))

(tell (:about Call Object
              (object-type it::call-noun)))

(tell (:about Seat Object
              (object-type it::seat)
              (owner hearer)))
```

Figure 4.5: The PRL Object Commands for the Remove-Phone Text

Now, using these rhetorical lexical, grammatical, and clause combining elements, the TRL has the capability of representing the detailed form of expression determined by the system network. The complete representation produced by IMAGENE for the remove-phone example will be shown at various stages of development in the next section.

4.4.2 The Realization Statements

IMAGENE uses specially designed text-based realization statements to construct the TRL structure as a side effect of the traversal of the system networks. The fundamental primitive operations involved in building a TRL structure are given here, along with the IMAGENE realization statements used to implement them:

- Inserting nodes (**iterative-insert, insert, copy**)
- Ordering nodes (**order, reorder, insert-order, combine**)
- Building RST structure between nodes (**structure, unlink**)
- Grammatical marking of the nodes (**mark, iterative-mark**)

The TRL structure starts with a single system defined root node, called ***Text Root***, and is incremented, using **Insert**, **Iterative-Insert**, and **Copy**, as deemed appropriate by the various systems in the three networks. **Insert** inserts a single node into the TRL, but doesn't link it in any way with the others. **Iterative-insert** iteratively inserts various types of children for a node and structures them as an RST joint schema. **Copy** makes a copy of a node in the network, including all RST or sentence order links to or from the node and all of its grammatical markings.

The ordering statements manipulate the sentence trace. **Order** is used to order two previously unordered nodes while **Reorder** modifies the order of two previous ordered nodes. **Insert-Order** allows a new node to be ordered between two previously ordered nodes.

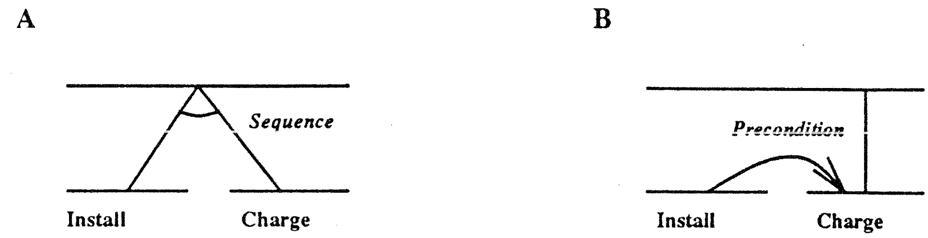

Figure 4.6: A Structural View of Rhetorical Demotion

Structuring is the process of building up the rhetorical structure in the TRL. IMAGENE must both add rhetorical links, using **Structure**, and remove them, using **Unlink**. This provides a powerful capability for both building and manipulating the structure. One common manipulation, which can be taken as an example of the TRL modification process, is that of rhetorically demoting an action from imperative status (as seen in figure 4.6A) to precondition status (as seen in Figure 4.6B). This is accomplished by first adding the **precondition** link (using **Structure**) and then removing the joint sequence link (using **Unlink**).

Finally, **Mark** is used to set the appropriate lexical and grammatical markings for each expressible node in the TRL. It can set any of the grammatical markings listed in the previous section on the TRL.

IMAGENE's realization statements have a significantly different flavor than their counterparts in the Nigel grammar developed for Penman (Nigel, 1988). Nigel's sentence-level realization statements work with a single, pre-specified list of features of the sentence, called *grammatical functions*, such as ACTOR, PROCESS, GOAL, and THEME. The appropriate list of these features was determined by a linguistic analysis of English sentence constructions. At the text-level, there is no definitive list of what would be called *text functions*. Rather, IMAGENE's realization statements allow the insertion of subscripted elements which can correspond to lists of sequential commands, multiple preconditions, etc. In the cases where Penman must deal with similar lists of arbitrary numbers of elements, such as with compound NP's, it uses recursion to avoid name-space problems. Given the frequency with which this phenomenon occurs at the text level, IMAGENE was forced to employ a more general mechanism.

Closely related to this subscripting of text functions is the issue of referring to the appropriate function in the realization statements. In Nigel, the appropriate, pre-defined function is simply referenced by name in the grammar. At the text level, it is not always clear which subscripted element is being referred to in the realization statement, rather, the element being references is defined structurally as, say, the immediately following node or the parent node. To allow this kind of specification, IMAGENE includes a number of predefined operators which refer to text functions based on the local structure of the TRL, rather than by global name. These operators include:

- Previous-node
- Following-node
- Nucleus-node
- Satellite-node
- Parent-node
- Child-node

These operators are interpreted before the realization statements execute, allowing the realization statements to make charges not only to the current node of interest, called ONUS, but also to node related to it such as its parent or its nucleus, without actually knowing the particular subscript of that node.

4.5 The System Network

The function of the systems in IMAGENE is to establish the list of features of the communicative environment that are relevant in choosing the form of expression of procedural relationships. These features are expressed in a discrimination network implemented as a system network. The attraction of the system network formalism for the current study was the ease with which the results of the corpus analysis could be coded in it. The level of analysis performed using the corpus techniques detailed in Chapter 3 can very easily be coded in terms of a Penman-style system network.

As noted above, there are three basic sub-tasks that this network performs, the task of building the initial rhetorical structure of the expression, that of rhetorically analyzing each of the actions, and finally that of combining the resulting clauses into sentences. These three tasks are generally executed in this order, but the facility, provided by the system network formalism to modify this general order in appropriate cases, has been exploited in a number of cases. The specific instances of this reordering are discussed in Chapters 5 and 6.

The navigation of these systems is controlled by the results of the text-level inquiries, discussed above, that determine the precise nature of the communicative context. Currently, these inquiries have not been implemented, that is the data structures and code necessary to respond to them automatically have not been designed. Rather, the inquiries are answered manually, allowing the current study to focus on the lexical and grammatical consequences of the results of the inquiries. The result of navigating through these systems is the construction of the appropriate Text Structure. This construction is accomplished by the execution of a number of realization statements which are executed as a side effect of running the system networks.

This section discusses the three basic tasks of the system network shown in Figure 4.2 and the nature of the inquiries and realization statements performed by each one pursuant to building the TRL structure. The aim is not to give a detailed analysis of the systems in each of the sub-networks; there are nearly 70 systems over all, each of which will be discussed in some detail in Chapters 5 and 6. Rather the nature of the architecture and how it accomplishes its task will be given.

4.5.1 The Structure Systems

The purpose of the structure building systems is to produce the basic hierarchical structure upon which the text will be based. These systems occur early in the traversal of IMAGENE's system network and are shown in Figure 4.7. The basic form of the resulting text structure will follow that of the process structure (called "PRL structure" in figure 4.2), but needs to be sensitive to the nature of the device and the assumed knowledge of the reader. Were the structure systems to be run in isolation, their output for remove-phone text would be similar to the structures shown in figures 4.9 and 4.8. Figure 4.9 shows the TRL commands for the simulated structure, and Figure 4.8 provides a graphical picture of this structure. Note that because the IMAGENE system network is executed as a whole, this structure alone would never exist at any point in the execution of the network. It is, rather, the hypothetical output of the structure systems if they were executed in isolation. It is intended to illustrate the function of the structure systems. This section will now discuss the elements of the functional context that affect the execution of the structure systems, and particularly how they affected the generation of the remove-phone text.

4.5.1.1 The Nature of the Device.

Simple procedures that will be performed once give rise to procedural sequences of imperatives in the text. This is not surprising. These procedures display what this thesis terms a *procedural* structure. Very few instruction manuals, however, exhibit this form at the highest level. Typical devices, being more complex, provide large sets of primitives that may be used in various combinations to achieve various goals. This type of process gives rise to a *primitive* organization, where the primitives provided by the device are presented in various orders (including alphabetical order, structural order, and order of importance). This characterization is similar to the one proposed by Nida (1984). As the structure tools network proceeds through the actions in the process structure in depth-first fashion, it builds up a preliminary rhetorical structure that contains either procedural or primitive lists, based on the nature

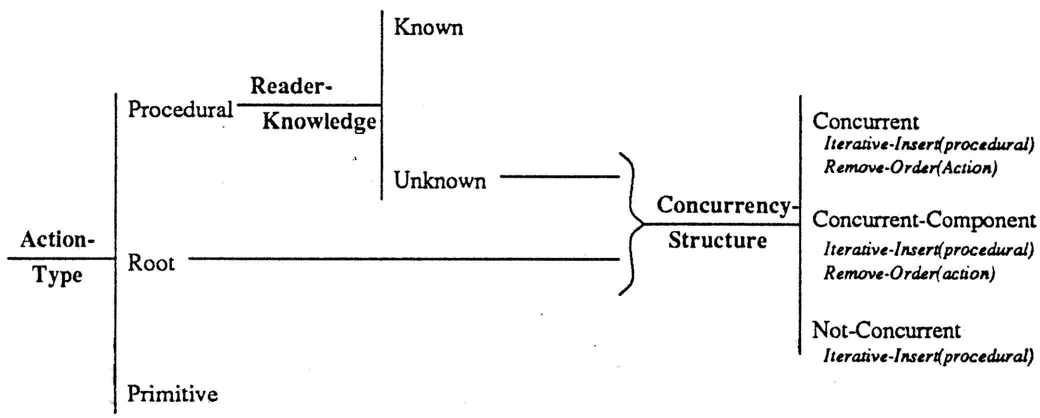

Figure 4.7: The Structure Systems

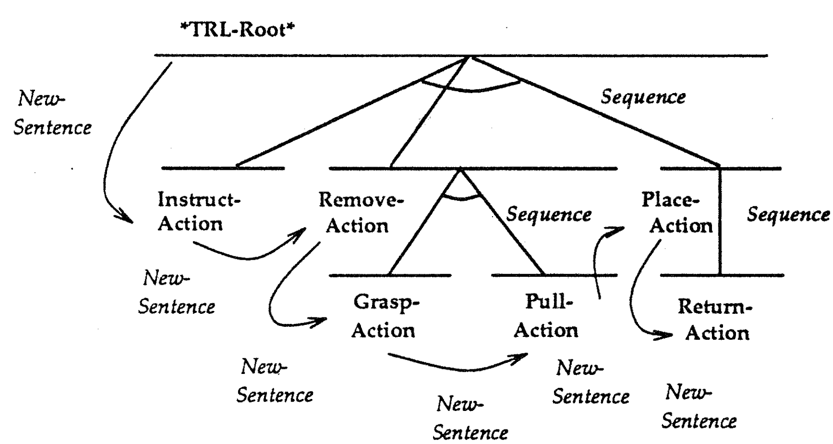

Figure 4.8: A Graphical View of the Hypothetical Output of the Structure Systems for the Remove-Phone Text

```
(tell (:about grasp-action span
       (new-sentence pull-out-action)))

(tell (:about instruct-action span
       (new-sentence remove-action)))

(tell (:about pull-out-action span
       (new-sentence place-action)))

(tell (:about return-action span))

(tell (:about ig-text span
       (new-sentence instruct-action)
       (procedural-subspan remove-action)
       (procedural-subspan instruct-action)
       (procedural-subspan place-action)))

(tell (:about remove-action span
       (new-sentence grasp-action)
       (procedural-subspan pull-out-action)
       (procedural-subspan grasp-action)))

(tell (:about place-action span
       (new-sentence return-action)
       (procedural-subspan return-action)))
```

Figure 4.9: The Hypothetical TRL commands Output by the Structure Tools Network for the Remove-Phone Text

of the particular actions involved. This distinction is made in the **Action-Type** system shown in figure 4.7. A further option in this system is the *root* option, which is chosen only for the root node of the entire rhetorical structure of the text. This node receives special attention because it is never expressed.

The current study has not addressed macro-structure in much detail. The structure tools sub-network is the only part of IMAGENE that deals with it, and it does so only in a high-level way. All of the examples dealt with in the thesis concern smaller segments of no more than a dozen actions, most of which exhibit project structures. The remove-phone text, for example, includes only procedural sequences and thus results in the *Sequence* relations in figure 4.8). Concern with macro-structure is common in schema-type generation systems (McKeown, 1985). For another interesting study of macro-structure in another textual domain see Mooney's work on the macro-structure of congressional reports (1990; 1991).

4.5.1.2 The Assumed Knowledge of the Reader.

The process structure produced by a procedural planner will inevitably include detailed descriptions of actions that the reader may already know, and thus need not read about. The structure systems, therefore, inquire about the reader's knowledge of a particular procedure, pruning that part of the process structure if the reader already is familiar with it. If, in the running example, IMAGENE had determined that the reader already knew how to remove the phone, it would have produced:

(18) When you are instructed, remove the phone.

This knowledge on the part of the reader would be highly unlikely in the case of the Airfone, a device which is largely novel to the reader. Thus, in the Remove-Phone example, the detail concerning how the phone is removed is retained in the structure tools output, see Figure 4.8. This distinction is made by the **Reader-Knowledge** system. In order to respond to this inquiry, the generation environment would have to have an implemented user model capable of making this determination.

The **Concurrency-Structure** system ascertains the type of procedural node that is involved. For all of the procedural types, an arbitrarily long list of children is added using the *Iterative-Add* realization statement. After the entire system network is traversed for the current action, the structure systems are re-entered once more for each of the added children. A Concurrent node indicates that the children are to be performed concurrently, a concurrent-component node represents sequential actions in the context of concurrency. Both of these concurrent node types are immediately removed from the sentence order using the *Remove-Order* realization statement. A specific example of how these nodes are used is the Furnace text, which will be discussed in detail in Chapter 8.

4.5.2 The Action Systems

The action systems have been the focus of the majority of the current study's linguistic investigations (Vander Linden et al., 1992b; Vander Linden, 1992). They determine both the rhetorical status of the expression of that action and the precise lexical and grammatical form of its expression. The general layout of this sub-network is shown in figure 4.10. The network is executed from left to right, the earliest systems performing rhetorical status selection and passing control off to the appropriate rhetorical relation sub-network for the detailed grammatical marking. The systems in the network are not completely stratified in this way, but this is illustrative of the general flow of control.

The corpus study has identified a number of distinctions that are important in the determination of the appropriate form of the expression for the action; these distinctions are implemented as individual systems in this network.

In the running example, the hypothetical output of the structure tools (see Figure 4.8) would include six actions: instructing, removing, grasping, pulling, placing, and returning. Again, because the structure systems do not run in isolation, this output is only hypothetical, but does serve to illustrate the types of structures produced by the action systems. This section will discuss the execution of the action systems for each of these actions.

The rhetorical status selection sub-network of the action systems determines that the instructing action ("When instructed") should be expressed as a precondition because it is an action performed

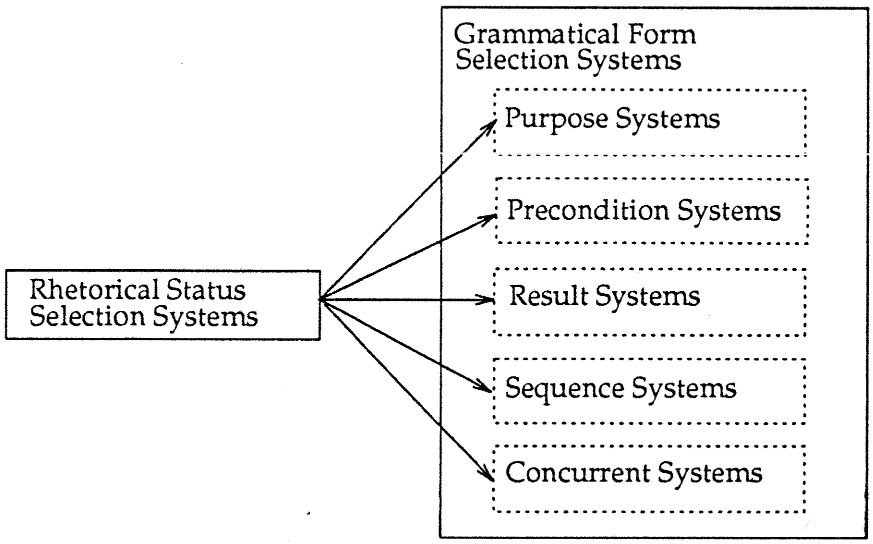

Figure 4.10: A High-Level View of the Action Systems.

by a non-reader agent and also the culmination of a monitoring action by the reader. Control in the network is thus passed to the precondition portion of the action systems for detailed grammatical marking. Because the action is a possibly complex process performed by a non-reader agent, the agentless passive form is chosen. The linker "when" is chosen because the action is not conditional, that is, because the action will be performed under normal circumstances. This precondition is fronted because the action that it relates to is not the topic of the local text, where local topic is defined as the single component that a paragraph is dedicated to. This type of paragraph is fairly common in manuals with primitive organization but was not used in the context of the remove-phone text.

The removing action ("remove phone") becomes a purpose because it is a higher level action in the process structure, commonly termed a *generator* action (Goldman, 1970), and is passed to the purpose sub-network. Typically, generator relationships are said to hold between single actions such as "Turn on the light by flipping the switch." There is no reason, however, to make this a hard restriction. A generation relation can just as easily hold between a high-level action (in this case, removing) and a lower level sequence of actions (in this case, grasping and pulling), a fact observed by DiEugenio (1992). The removing expression is fronted because it pertains to more than one subaction. It is expressed as a "by" prepositional form rather than a "to" infinitive or nominalization because of the existence of a condition on the purpose ("when you are instructed"). In the absence of this condition, a fronted "to" infinitive would have been produced.

The grasping and pulling out actions ("by grasping top of handset and pulling out") become conjoined gerunds in the propositional phrase. Were it not for the grammatical constraint imposed by the "by" prepositional phrase, they would have been expressed as imperative actions. They are expressed in order because there is no specific functional reason to do otherwise, such as would be the case if the first was considered procedurally obvious.

The placing and returning actions ("Return to seat to place calls") are another purpose, sub-action pair. In this case, however, the purpose is expressed as a non-fronted "to" infinitive. There are a complex set of reasons for this which will be discussed in detail in Chapter 6. For the purposes of the current chapter, it is important to note that this purpose is expressed differently from the removing purpose ("remove phone") because of differences in the communicative context.

The hypothetical final TRL output of the action systems is just as will be shown in Figure 4.11 except that the sentence trace links may not be set properly, a task left to the sentence systems.

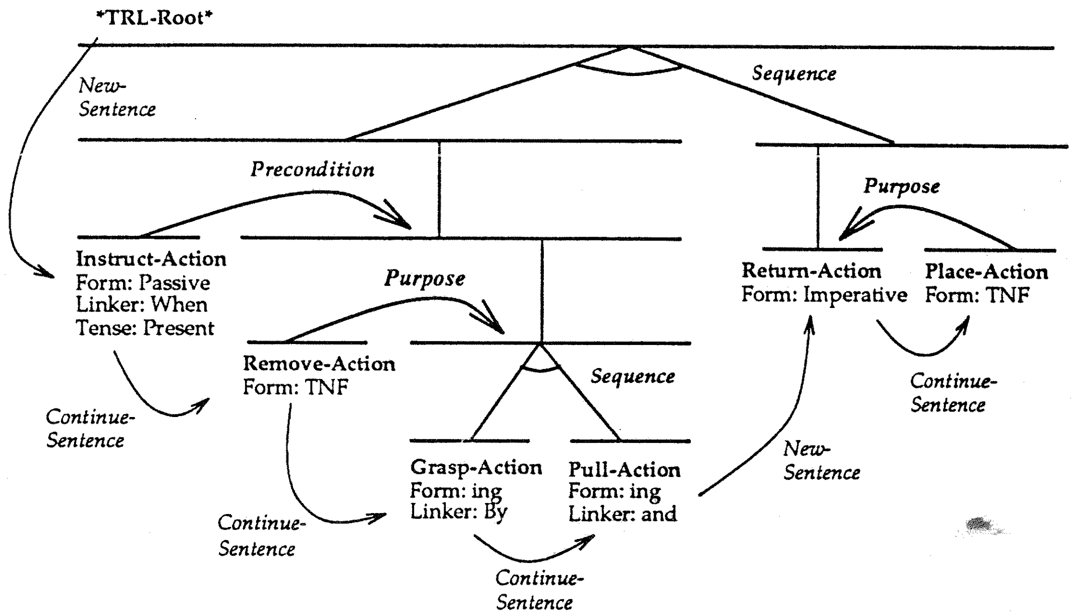

Figure 4.11: A Graphical View of the TRL Command Output by IMAGENE for the Remove-Phone example.

4.5.3 Clause Combining

The task of clause combining is largely completed as part of the action systems, but there are some extra systems included at the end of the network traversal that exclusively deal with clause combining. These systems will be detailed in Chapter 6. They determine whether or not the current action should be combined in a single sentence with its nucleus node or its following node, and if so, what linker should be used.

In the running example, the precondition, purpose, and action nodes are already combined in a single sentence by virtue of the fact that they form a single action sequence with a precondition and a purpose. One issue that is left for specialized sentence systems is the choice of the appropriate linker for the grasping and pulling actions. This choice is based, as will be discussed in Chapter 6, on the possible existence of unwanted concurrency between the two actions involved. In the Remove-Phone text, no unwanted concurrency between grasping and pulling is possible, so IMAGENE chooses "and".

The resulting Text Structure produced by the three sub-networks for the remove-phone example is given in figures 4.11 and 4.12. The bold faced lines represent the rhetorical structure and the light faced lines represent the sentence trace. All expressible nodes have the appropriate lexical and grammatical features set as well. The linker feature, seen in the **pull** node, is included with the conjoined phrase or clause. The root of the TRL structure and two internal nodes will not be expressed in the resulting text, and thus neither have grammatical features and nor are included in the sentence trace. Notice also that the RST-type rhetorical relations are specified in the structure. These were marked following the rhetorical status selection performed by the action tools sub-network. These markings are used by the SPL translation module to aid in the production of the appropriate SPL command for the Text Structure.

4.6 The SPL Builder

The final component of IMAGENE's architecture is the SPL builder (see figure 4.2). SPL is Penman's Sentence Planning Language, which IMAGENE uses to command Penman to produce the sentences specified in the final Text Structure. SPL is an extensive and flexible sentence specification language

```
(tell (:about pull-action trl:span
       (trl:linker 'and) (trl:form 'trl::ing)
       (trl:new-sentence return-action)))

(tell (:about remove-action trl:span
       (trl:continue-sentence grasp-action)
       (trl:form 'trl::imperative)))

(tell (:about superordinate-purpose156 trl:span
       (trl:procedural-subspan internal-purpose156)
       (trl:precondition instruct-action)))

(tell (:about internal-precond154 trl:span
       (trl:procedural-subspan superordinate-purpose156)))

(tell (:about return-action trl:span (trl:form 'trl::imperative)
       (trl:continue-sentence place-action)
       (trl:purpose place-action)))

(tell (:about place-action trl:span (trl:form 'trl::tnf)))

(tell (:about instruct-action trl:span
       (trl:linker 'when) (trl:tense 'trl::present)
       (trl:form 'trl::stative-passive)
       (trl:continue-sentence remove-action)))

(tell (:about ig-text trl:span
       (trl:new-sentence instruct-action)
       (trl:procedural-subspan superordinate-purpose157)
       (trl:procedural-subspan internal-precond154)))

(tell (:about grasp-action trl:span
       (trl:continue-sentence pull-action)
       (trl:form 'trl::ing) (trl:linker 'by)))

(tell (:about internal-purpose156 trl:span
       (trl:purpose remove-action)
       (trl:procedural-subspan grasp-action)
       (trl:procedural-subspan pull-action)))

(tell (:about superordinate-purpose157 trl:span
       (trl:procedural-subspan return-action)))
```

Figure 4.12: The TRL commands Output by IMAGENE for the Remove-Phone Text

that is structurally analogous to the TRL structures just discussed. It allows the recursive decomposition of sentence specifications just like the TRL does, and thus the task of translating from the TRL to the SPL is a relatively straight-forward problem of recursive descent. A more detailed specification of this language can be found in the full Penman documentation (Penman, 1989).

The SPL Builder uses the rhetorical and grammatical information from the TRL representation of the Text Structure, augmented by the lexical information in the PRL representation of the Process Structure, to build a sequence of commands which are passed individually to Penman. For example, given the Text Structure for the Remove-Phone text, shown in figure 4.11, and the Process Structure shown in Figure 4.3, IMAGENE constructs the two SPL commands shown in figures 4.13 and 4.14. It is important to note here that the issues of referring expressions, and the case structure of the clauses and phrases in the example are predetermined in the process structure (see Figure 4.3). IMAGENE has prescribed the rhetorical status and the lexical and grammatical marking at the rhetorical level.

The SPL Builder starts at the top of the Text Structure and works its way down, producing the roughly analogous SPL commands as it goes. It makes the requirement that only one satellite be specified for each nucleus.[1] Thus, for the Remove-Phone's Text Structure, it first produces the necessary SPL command for the precondition ("When you are instructed"). This is realized in SPL using a `concurrent` type command (called a "term" in SPL Penman, 1989), see figures 4.13 and 4.14. This term specifies a `:domain`, a `:range`, and a `:theme`. The domain is a term specifying the remainder of the text structure, the range is the precondition itself, specified as a present tense dispositive-material-action, and the theme indicates that the range (variable `r427`) is to be fronted.

The SPL builder then recurses down in the text structure to the purpose expression, specified in Figure 4.13 as a another dispositive-material-action (`remove`) with an `:enablement` term. The enablement term results in a "by" preposition. The term for the enablement specifies the `:domain` (grasping) and the `:range` (pulling). Note that the `:actee` of the specification for the pulling expression is a simple variable name (`p461`), referring to the full specification of the handset given above in the text. This is a result of the simple pronominalization mechanism that IMAGENE uses. It uses the full NP specified in the process structure for the first reference to the object, and they uses pronouns for the remainder of the text. Much more complete and accurate models of referring expressions can be found (e.g., Dale, 1992). A similar process is then executed for the second SPL command, passing it to Penman in a similar manner.

4.7 Penman

Penman, as discussed in detail in earlier chapters, is a sentence-level generator for English based on the Systemic-Functional view of language. In a manner that inspired the approach taken in IMAGENE, Penman traverses a complex system network which effects the generation of sentence structures. In fact, the same system network mechanism that was implemented for Penman has been used in IMAGENE. This ability to use the same mechanism or approach is based on the fact that building texts is very much like building sentences in that it is a process of querying the relevant features of the functional environment and effecting realization at the appropriate level of expressional abstraction.

Penman provides two fundamental interfaces for surface realization, the SPL command interface discussed in the previous section and the raw inquiry interface.[2] To exercise complete control over the Nigel grammar (Penman's grammatical component Nigel, 1988), the inquiry interface would have to be used, but this is quite a tedious process. IMAGENE would have to pre-specify a great number of responses for various inquiries to use this interface, and the extra effort would not buy much for the current study. It would be helpful in some issues of lexical selection and rhetorical selection because Nigel provides some facilities for doing these things. IMAGENE, however, either explicitly addresses the issue with its own system networks, as is the case with linkers and clause combining, or provides

[1]Occasionally, RST analyses in the literature have included multiple satellites. This practice was used in some cases in the analysis for the current study as well, but is never used in the construction of the TRL in IMAGENE.

[2]There is one other concept-level interface, making use of Penman's *SPL Constructor* routines, but it doesn't appear to be as well developed as the other two. IMAGENE has basically produced its own SPL constructor interface, implemented in the SPL Builder and tailored to the unique requirements of the instructional domain.

```
(s114 / concurrent
      :domain (d114 / dispositive-material-action
          :lex remove
          :actor (hearer / person
               :lex hearer)
          :actee (a116 / decomposable-object
             :lex phone
             :determiner the)
          :speechact imperative
          :enablement (e131 / conjunction
                  :event-q complex-event
                  :domain (d131 / dispositive-material-action
                      :lex grasp
                      :actor (hearer / person
                           :lex hearer)
                      :actee (a132 / decomposable-object
                         :lex top
                         :determiner the
                         :part-of (p148 / decomposable-object
                              :lex handset
                              :determiner the))
                      :contrastive-extension-q simple-extending
                      :typical-participant-q nottypicalclass)
                  :range (r154 / dispositive-material-action
                     :lex pull
                     :actor (hearer / person
                          :lex hearer)
                     :actee p148
                     :contrastive-extension-q simple-extending
                     :typical-participant-q nottypicalclass)))
      :range (r114 / dispositive-material-action
         :lex instruct
         :actee (hearer / person
              :lex hearer)
         :tense present)
      :theme r114)
```

Figure 4.13: The SPL Command for the First Sentence in the Running Example

```
(s184 / rst-purpose
      :domain (d184 / nondirected-action
          :lex return-movement
          :actor (hearer / person
               :lex hearer)
          :speechact imperative
          :destination (d195 / decomposable-object
                    :lex seat))
      :range (r184 / dispositive-material-action
         :lex place-call
         :actor (hearer / person
              :lex hearer)
         :actee (a202 / decomposable-object
            :lex call-noun))
      :theme d184)
```

Figure 4.14: The SPL Command for the Second sentence in the Running Example

simplified solutions to the issue, as is the case with lexical selection for verbs and their arguments. Thus, the higher-level SPL command interface is used by IMAGENE.

Each new domain of interest requires that Penman be made to produce a new set of lexical and grammatical forms. Penman offers facilities for building a domain specific list of concepts and associated lexical items, which was used in the current study to extend Penman's lexicon to the domain of instructional text for telephone manuals.[3] Nearly all the examples dealt with by IMAGENE are expressed in terms of the objects and actions in the telephone domain. This simplification was necessary as it is quite beyond the scope of this project to create a lexicon for all of the devices and domains represented in the corpus.

Penman also had to be extended grammatically in a couple of cases. Although Penman does not provide specific utilities for this extension, it was still possible to manually expand the grammar to allow the generation of all the important expressional forms found in the instructional text corpus. As an example of one of the forms that Penman doesn't generate, consider the Installed-Phone text, repeated here:

Installed-Phone Text:

> When the 7010 is installed and the battery has charged for twelve hours, move the OFF/-STBY/TALK [8] Switch to STBY. The 7010 is now ready to use. Fully extend the base antenna [12]. Extend the handset antenna [1] for telephone conversations.

The compound, hypotactic expression "When the 7010 is installed and the battery has charged for twelve hours, . . ." is not supported currently in Penman. This form is very common in instruction manuals when expressing multiple preconditions, and thus needed to be added to Penman's competence. The modification required relatively simple extensions to a couple of Nigel's systems. A similar change was required to handle the compound construction in the Remove-Phone text, "by grasping the top of the handset and pulling it".

IMAGENE's version of the Remove-Phone text, resulting from Penman's execution of the SPL commands given in figures 4.13 and 4.14, is given here:

(19) *When you are instructed, remove the phone by grasping the top of the handset and pulling it. Return to a seat to place a call.*

[3]Special exception to this was made for some isolated examples taken from text outside the telephone domain, such as the Furnace text.

4.8 Examples of IMAGENE's Output

This section includes examples of IMAGENE's output and is intended to demonstrate its coverage of the range of lexical and grammatical forms found in the corpus. How these forms are produced is the topic of Chapters 5 and 6, but this section will give a short description of each form as an introduction.

Given the choice to express an action, rhetorically, as a purpose, IMAGENE is capable of producing seven grammatical forms for its expression, most of which can either be fronted or not fronted. Here are the various forms, as generated by IMAGENE:

(20a) *To end a call, hold down the FLASH button for two seconds, then release it.*

(20b) *Follow steps in the illustration for desk installation.*

(20c) *Use the OFF position for charging the batteries.*

(20d) *Use the REDIAL for frequently busy numbers.*

(20e) *When you are instructed, remove the phone by grasping the top of the handset and pulling it.*

(20f) *Remove the phone. Grasp the top of the handset, and pull it.*

(20g) *Tilt the pan so that the fluid drains out.*

If the purpose is global in scope, it is usually expressed as a fronted "to" infinitive, as in example (20a) (*"To end a call"*). If the purpose is local, IMAGENE specifies a "for" prepositional phrase, as in example (20b) (*"for desk installation"*), provided the nominalized form of the verb exists (as specified in the lexicon). The "for" gerund form, shown in example (20c) (*"for charging the batteries"*), is used in cases where there is some problem with the "to" infinitive expression. In this case, the action of using the OFF position is not sufficient to achieve the purpose of charging the battery. In cases such as (20d) (*"for frequently busy numbers"*), where the purpose is based on some semantically simple action such as "deal with" or "handle", a "for" prepositional phrase with the direct object (or goal) of the action as the complement is used. This form is called *goal metonymy* because the goal of the predication is being used to metonymically refer to the entire action. The "by" purpose is used when there is some condition placed on the execution of the purpose. In example (20e) (*"remove the phone"*), the condition that the reader wait until instructed is expressed. If a "by" purpose expression gets too long, then an *adjoined* purpose form is used, as in example (20f). The final purpose form, the "so that" purpose, is shown in example (20g) (*"so that the fluid drains out"*), and is used in cases where the reader is expected to have the purpose of getting some non-reader agent to behave in a volitional way. In this case, the reader is expected to have the purpose of getting the fluid to drain out of the pan. This example is not taken from the telephone domain because this form does not occur in the phone manuals.

Given the choice to express an action, rhetorically, as a precondition, IMAGENE is capable of producing four grammatical forms for its expression, all of which can either be fronted or not fronted and also linked with various linkers. Here are the forms, as generated by IMAGENE:

(21a) *If light flashes, insert credit card.*

(21b) *The BATTERY LOW INDICATOR will light when the battery is low.*

(21c) *When the phone is installed, and the battery is charged, move the OFF/STBY/TALK switch to the STBY position.*

(21d) *Return the OFF/STBY/TALK switch to the STBY position after your call.*

A present tense active form of the action involved may be used as shown in example (21a) (*"If light flashes"*). A relational clause is used in example (21b) (*"when the battery is low"*) to hide the temporally prolonged and possibly temporally remote action of the battery loosing its charge. IMAGENE maintains, in its lexicon, semantic links between actions (drain) and the appropriate predicate adjectives (low). Preconditions typically precede the actions they pertain to, but in this case the battery low indicator is the topic of the local context so its expression is fronted (its form will be discussed in the next section). Example (21c) contains an agentless passive precondition. The agentless passive (*"When the phone is installed and the battery is charged"*) is used for most reader actions that become preconditions. Both of these actions are examples of previously mentioned actions which become preconditions. Finally, example (21d) shows the rhetorically demoted action discussed above.

There are two types of results that IMAGENE supports. Non-reader actions that are not the result of an explicit command to monitor a particular device state are typically expressed as results. IMAGENE expresses this type of result as a future tense clause. The second type of result is not based on an action in the process structure at all, but rather, is a span added by the system networks to signal a state resulting from an expressed action. In a sense, this latter case is one where an action is expressed both in action form, as an imperative, and as a state form. Here are examples of these forms:

(22a) *The BATTERY LOW INDICATOR will light when the battery is low.*

(22b) *When the phone is installed, and the battery is charged, move the OFF/STBY/TALK switch to the STBY position. The phone is now ready to use.*

The preconditions that occur in these examples have already been discussed. The issue here is the form of the results. In example (22a) uses the future tense action form (*"The BATTERY LOW INDICATOR will light"*) which typically follows the action it refers to, and is not combined in the same sentence. Because the battery draining action is performed by a non-reader, however, the action was demoted to precondition status and was combined with the result expression.

The second result, in example (22b) (*"The phone is now ready to use"*), is added by IMAGENE's system networks to mark the preparedness of the device for normal use; this feature typically co-occurs with the repeated mention of installation actions already discussed in the context of the preconditions (see example (21c)).

Any action that does not fit into the categories discussed above is marked as an imperative command. These commands are combined into clauses by the sentence tools system network using "and" when the concurrency that could be implied is impossible or inconsequential, as in example (20f), or "then" when there is possible unwanted concurrency, as in example (20a).

In summary, IMAGENE is capable of producing a wide range of lexical and grammatical forms for expressing actions. Precisely how this range of forms are produced is the subject of Chapters 5 and 6, but this section will conclude with a discussion of just how broad that range is. Currently, IMAGENE is capable of producing each of the 7 forms shown in Example (20) in either the initial or the final position, yielding 14 possible purpose forms. Similarly, it is capable of producing the first 3 precondition forms shown in Example (21) in either slot and with any of 4 linkers. It can also produce the last form, shown in Example (21d), and another imperative form which will be discussed in Chapter 6. This yields 27 possible precondition forms. IMAGENE also produces the 2 result forms shown in Example (22), as well as 6 concurrent forms and 6 sequential forms which will also be discussed in Chapter 6. This results in a total of 55 possible forms for any action expression.

4.9 Summary

This chapter opened with a characterization of the architecture of IMAGENE in terms of Bateman and Hovy's taxonomy of generation architectures. This served to place the issues and methods of the current study in the broader context of text generation research. The chapter then detailed the basic architectural structure of IMAGENE, working through the example of the Remove-Phone text to make the discussion more concrete. It dealt primarily with the details of the mechanism used to perform the generation, including the representation languages, the realization statements, the general architecture of the system networks, and the SPL builder. The concluding section gave a sense of the range of lexical and grammatical forms that IMAGENE is capable of generating. These details provide the background necessary to understand the detailed exposition of the system networks themselves which will be given in Chapters 5 and 6.

Chapter 5

Rhetorical Status Selection

The action systems perform both the rhetorical status selection for each action and the grammatical form selection. This chapter will discuss the systems dedicated to rhetorical status selection, leaving the grammatical form selection systems to Chapter 6. The break between the groups of systems that perform these two tasks is not a clean one. They are all part of the same network and there are occasions when the systems discussed in this chapter determine grammatical form and, conversely, when the systems discussed in the grammatical form selection chapter determine the rhetorical status. The distinction, nonetheless, is interesting theoretically as well as useful pedagogically and will be retained in this thesis.

The task of rhetorical status selection, as was mentioned in Chapter 2, has much in common with the work in Discourse-Functional Linguistics concerning the functional motivations for "subordinated" clauses (Thompson, 1987; Thompson, 1985; Matthiessen and Thompson, 1987). This work has sought to identify various features of the functional environment that motivate the use of subordinated expressions of various kinds.

The analysis done for the current study has yielded a detailed list of the functional criteria for rhetorically demoting an action expression to subordinate clause or even to phrase status. The basic task of the systems devoted to this process of rhetorical status selection is to route control during the traversal of the full system network to the sub-network that is appropriate for each action. This routing task is depicted in Figure 5.1, repeated from chapter 4.

The list of functional criteria that is used by IMAGENE to make this decision is formalized in the subset of the full action systems sub-network shown in Figure 5.2[1] which shows a list of systems whose execution determines the set of features of the functional environment that have proven useful in determining the rhetorical status of an action expression. This choice of the rhetorical status of the action expression is the first selection made in the course of executing the action systems, determining which of the grammatical form selection sub-networks of the action systems is to be entered. These grammatical form selection sub-networks are discussed in Chapter 6.

This chapter begins with a discussion of the nature of the functional features involved in determining rhetorical status (shown in Figure 5.2), specifying how each one can be derived from information available to the generation environment. It is important that these features be derivable from the information generally available to the text generator at the time of text planning in order for them to be useful in planning the text. A more concise list of these features, along with those important for grammatical form selection, is given in Appendix D. This chapter concludes with a discussion of the primary rhetorical relations used in the current study, namely purposes, preconditions, results, concurrency and sequence, explaining the various combinations of functional features that lead to the expression of each one.

[1]This chapter and the following one on the grammatical form selection sub-networks, will include a number of graphical depictions of system networks. These networks will not always include all of the systems that were necessary for IMAGENE, but will include the most pertinent of them. A few systems, such as those needed to deal with RST scoping issues, were considered too low-level to be of must interest to the general discussion.

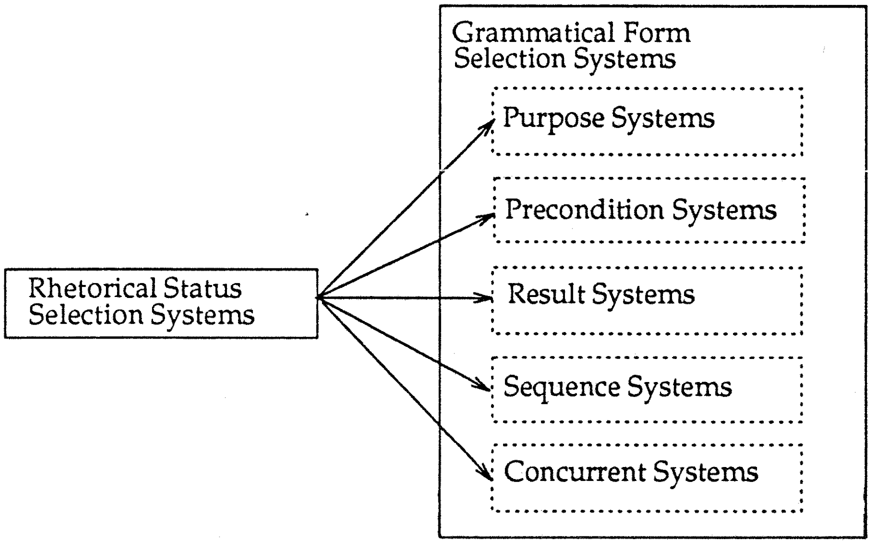

Figure 5.1: A High-Level View of the Action Systems.

5.1 Features for Rhetorical Status Selection

Before discussing the specific features characteristic of each possible rhetorical status, it is helpful, first, to present the terminology and concepts associated with each of the fundamental distinctions. There are five distinctions, shown in Figure 5.2, which determine the appropriate rhetorical status of each action expression, none of which is ordered in priority with respect to the others. They are as follows:

- **Action-Actor**
- **Temporal-Orientation**
- **Procedural-Obviousness**
- **Mention-Status**
- **Logical-Type**

To make the discussion of these features more concrete, the Installed-Phone text will be used as a running example in this section. All five of the functional distinctions will be discussed in the context of this example, but other examples will also be cited as necessary. The original text for this example is repeated here, and its input process structure is repeated in Figure 5.3.

Installed-Phone Text:

> When the 7010 is installed and the battery has charged for twelve hours, move the OFF/-STBY/TALK [8] Switch to STBY. The 7010 is now ready to use. Fully extend the base antenna [12]. Extend the handset antenna [1] for telephone conversations.

5.1.1 The Actor of an Action

The **Action-Actor** system determines the actor of a particular action. Because instruction manuals are largely directed at a particular agent, namely the reader, the only distinction that is relevant is between the reader and any other non-reader agent. This information is readily found in planner output, or at

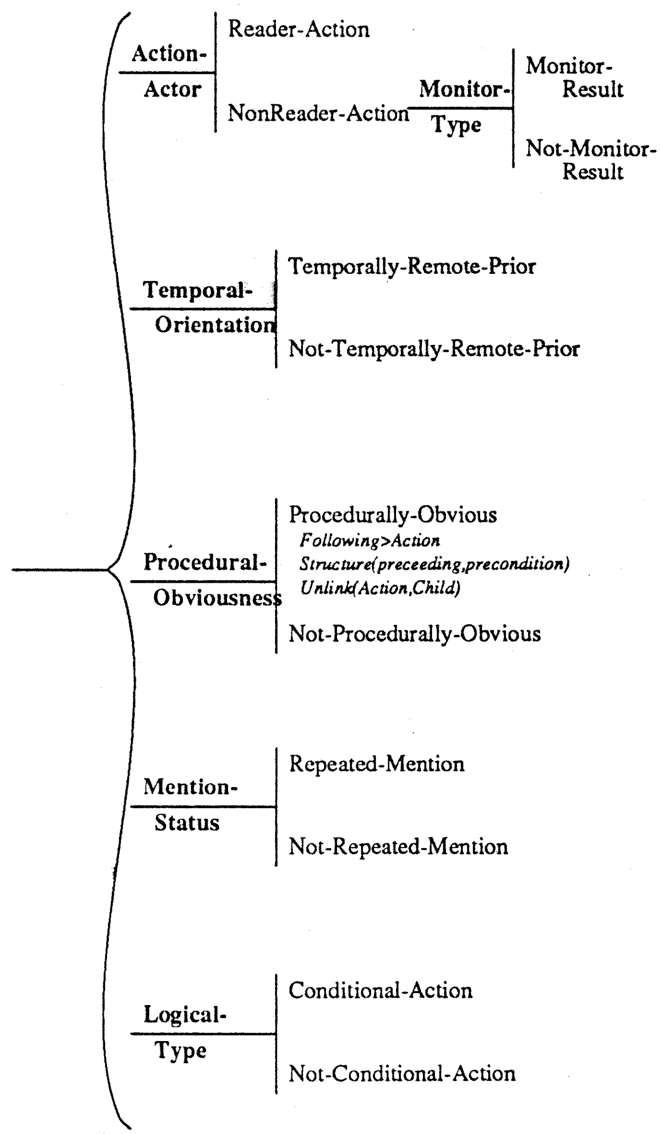

Figure 5.2: The Rhetorical Status Selection Sub-Network of the Action Systems

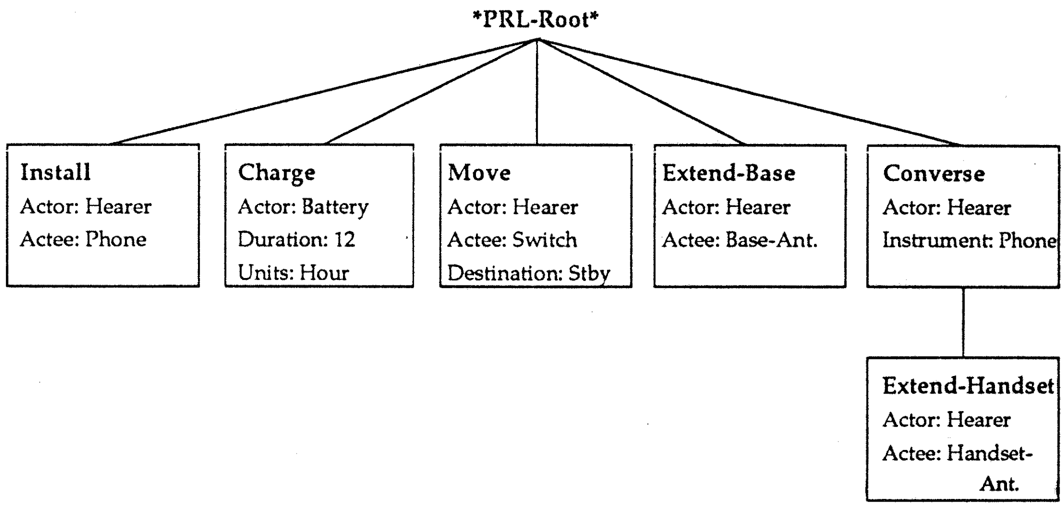

Figure 5.3: The Process Structure for the Installed-Phone Text

least can easily be added to the plan operators. **Monitor-Type** further divides the non-reader actions into those actions that are being specifically monitored by the reader and those that are not. This becomes relevant in the distinction between results and preconditions. The agent in all the actions in the Installed-Phone process is the reader, but this is not always the case. The Remove-Phone text, for example, includes a non-reader action:

Remove-Phone Text:

> When instructed (approx. 10 sec.) remove phone by firmly grasping top of handset and pulling out. Return to seat to place calls.

As will be discussed later in this chapter, the action of the phone instructing the reader to do something ("when instructed") is being performed, not by the reader, but by the phone itself. Such actions, obviously, are never expressed as imperative commands, but rather as preconditions.

5.1.2 The Temporal Orientation of an Action

The **Temporal-Orientation** system determines if the action was executed at a possibly remote time in the past. Such actions are seldom expressed as imperatives, but rather as preconditions. All of the actions in the Installed-Phone process must have been performed since the purchase of the device, and therefore are not said to have been performed at a temporally remote time in the past. An example of the a temporally remote action can be found in the following example, taken from a cordless telephone manual:

(23) *If you have touch-tone service*, move the TONE/PULSE SWITCH to the Tone position. (Excursion, 1989) [2]

In this set of action expressions, the action of installing the touch-tone service ("if you have touch-tone service") is an action performed at a possibly temporally remote time in the past. This distinction tends to lead to a precondition expression as was used here.

[2]This thesis will typically add a reference to the end of all examples that have come directly from the corpus, indicating which manual they came from. Italicized examples are actual IMAGENE output. All other examples are contrived for explanatory purposes.

5.1.3 The Procedural Obviousness of an Action

The **Procedural-Obviousness** system determines if the action is one that is understood by the reader. Obvious actions have two attributes: (1) They are presumed by the writer to be understood by the reader prior to the reading of the instruction manual; and (2) They form the over-arching goal of the current paragraph in the instruction manual. The co-occurrence of these two features typically leads to a precondition expression in spite of the importance of the action being expressed. None of the actions in the Installed-Phone process are known in this way to the reader before reading the passage. An example of this can be found in a set of instructions for making a call on a cordless telephone.

(24) Return the OFF/STBY/TALK switch to STBY *after your call.* (Code-a-phone, 1989)

What is unusual about this passage is that its main topic, that of making a call, is expressed in a precondition, rather than in a more grammatical prominent imperative. The analysis for current study indicates that, although it is somewhat uncommon, actions that are considered obvious to the reader and are the main topic of a passage are demoted in this way. None of the Installed-Phone process's actions are of this sort.

5.1.4 The Mention Status of an Action

The **Mention-Status** system determines if the action being expressed has already been prescribed to the reader in the form of an imperative. This information could be determined by referencing the rhetorical structure constructed by IMAGENE so far. The previous mentions of these actions must either be in the form of sequential commands or as purposes with sequential sub-actions. Two of the Installed-Phone actions are repeated mentions, the **Install** ("when the 7010 is installed") and the **Charge** ("and the battery has charged ...") actions. They were explicitly prescribed to the reader with imperative commands in the installation section of the cordless phone manual, which is immediately before the Installed-Phone passage.

5.1.5 The Logical Type of an Action

The final system, **Logical-Type**, determines whether the current action is expected to be executed in the process or if there are cases where it doesn't need to be executed. Conditional actions are those which have preconditions that may not be met in all instantiations of the plan execution. They are further divided into actions that are highly probable and those that are not. The Installed-Phone text actions are non-conditional. The precondition in Example (23), discussed above, is conditional in that it may not have occurred at all.

This is, admittedly, a difficult issue to address in the text planning environment, as the probability of, say, a particular phone number being provided with touch-tone service is difficult to determine. Its effect on the rhetorical and lexico-grammatical form of action expressions, however, requires that this sort of information be made available. All the actions in the Installed-Phone process are expected to be performed by the reader in the course of using the cordless phone, and are thus not conditional in the sense described here.

5.1.6 Discussion

It is important to note with these feature determinations that IMAGENE does not include the data structures and code necessary to respond automatically to the inquiries that determine the outcome of these systems. The current study has attempted to put the distinctions in terms that are, in principle, derivable from information that could be made available to the text generator. The implementation of this source of information, however, may prove difficult.

For example, the determination of procedural obviousness is not completely clear. The first of the two factors related to it, i.e., that the reader must know how to perform the action, would be particularly difficult for the text generation system to discern. One operational approach to the problem could be built within the context of adaptive planning (Alterman, 1988). This planning framework allows the use of, among other things, previous plans, in the planning and execution of a particular task. In this context, obvious actions could be operationalized as those actions for which the planner has a previous plan that is applicable to the current task. With respect to the telephone manuals, the use of a standard telephone plan would be a previous plan that could be used in the execution of a cordless telephone procedure; presumably this is precisely why the action is considered obvious in example (24).

These difficulties, however, should not detract from importance of the problem of choosing lexical and grammatical forms based upon functional features. In this light, the results of the current study concern the specific features that affect rhetorical and lexico-grammatical form as determined in a detailed corpus-based study, and, as such, can be seen as a set of prescriptions for information that should be included in the text generation environment, as discussed by Paris and Maier (1991).

In summary, given the determinations of this list functional features, IMAGENE selects the appropriate rhetorical status for the current action expression. The following sections will detail the reasoning that makes this determination for each rhetorical status.

5.2 Purpose Relations

The purpose relation occurs frequently in instructional text, and it has received, as was shown in Chapter 2, a fair amount of attention in the literature. The fundamental reason for demoting an action to purpose status is to signal a super-ordinate/sub-ordinate procedural relationship, termed a *generation* relation (Goldman, 1970; Balkanski, 1992; Di Eugenio, 1992). This procedural status of being at a higher level in the procedural hierarchy is easy to determine from the output of a planner and from the PRL and leads directly to the choice of a purpose expression in the action systems. The systems in Figure 4.10 do not include the system that makes this determination. It is made, instead, by the **Reader-Knowledge** system shown in the last chapter in Figure 4.7. If the reader is not expected to know about an action, sub-actions are added to make it more clear.

The determination of the appropriate level of detail at which to express a process is highly relevant to the rhetorical status of purpose. IMAGENE makes this determination in the structure systems, in the Reader-Knowledge system, shown in Chapter 4, Figure 4.7. As was discussed there, the structure systems start with the root of the hierarchical structure of the process and include as much detail as the reader is expected to need. The lowest level actions are not considered generation actions, even though they could have been if the details of their execution had been included in the output of the structure systems. One could, for example, express the **Move** action from the Installed-Phone text as a purpose and associated it with its subordinate actions, but this was not considered necessary.

The **Converse** action, shown in Figure 5.3, is the only generator action in the Installed-Phone process. It is expressed as a "for" prepositional phrase with a nominalization as the complement ("for telephone conversations"). Generator actions need not have only one sub-action as this one did. Consider the following text:

(25) To end a previous call, hold down flash [6] for about two seconds, then release it. (Code-a-phone, 1989)

The Process Structure for this expression would have a single high-level node ("ending") with two children at the leaf level ("holding down" and "releasing"). The expression of purposes such as these may take many different grammatical forms depending upon context; this is the subject of Chapter 6.

Thus, the rhetorical status of purpose is chosen only when the action being expressed is a generator action. The effect that some of the other functional features discussed in this chapter have on the lexico-grammatical form of purpose expressions will be discussed in Chapter 6.

5.3 Precondition Relations

Precondition relations are considerably more diverse than purpose relations. They arise in a number of different functional contexts, having to do with any of the interpersonal, textual, or ideational meta-functions. Here is a high-level list of the contexts in which an action is rhetorically demoted to precondition status:

- The action, performed by a non-reader, is being monitored by the reader — Readers are commonly expected to monitor a particular condition and do something when it changes. The culmination of this *monitor action* is typically expressed as a precondition;
- The action is considered obvious — Instruction writers consider some actions so obvious that they need not be explicitly stated as imperatives, discussed with respect to example (24) above;
- The action has been previously mentioned — Instruction manuals commonly contain multiple expressions of a single action, where the first is typically an imperative command and the second is stated as a precondition;
- The action is conditional — Any action that may or may not have been performed by the current stage of the process is considered conditional and is expressed as a precondition. "If" clauses typically encode this type of relation.

The Installed-Phone process includes no non-reader actions that are being monitored by the reader. The Remove-Phone text, however, does include an example of such a process:

Remove-Phone Text:

> When instructed (approx. 10 sec.) remove phone by firmly grasping top of handset and pulling out. Return to seat to place calls.

The "When instructed" expression is based on an action performed by the device, the Airfone in this case, that is being monitored by the reader. The reader is explicitly instructed to monitor a lighted display for instructions by the immediately previous sentence, "Observe lighted display above for instructions."

Neither does the Installed-Phone process does not include an obvious action. As discussed above, however, example (24) does. The "after your call" in that example is an important procedural component of the process being expressed, but has been rhetorically demoted because it was considered obvious to the reader.

The Installed-Phone process does include two previous mentioned actions, **Install** and **Charge.** As discussed in the previous section, both of these actions were explicitly prescribed in an early section of the manual, resulting in an altered rhetorical status selection.

The issues of obviousness and previous mention can be seen as specific contexts in which Dixon's subjects (1988) found implicit action expressions to be acceptable. His implicit action expressions are termed rhetorical demotions in the current study, and, at least in the case of preconditions, are commonly found in the context of obviousness or previous mention.

There are no conditional actions in the Installed-Phone text; an example can be seen in the following text:

(26) If you have touch-tone service, move the TONE/PULSE SWITCH to the Tone position. (Excursion, 1989)

In this process the actual possession of touch-tone service is conditional, that is, the phone company may or may not have installed touch tone service for the reader, leading to the "if" expression.

Given the decision to express an action as a precondition, for any of the four reasons just given, IMAGENE is capable of generating three basic forms of that action, all of which can either be fronted or not fronted and linked with various linkers. A precise statement of the range of choices for preconditions will be given in Chapter 6, but it is helpful here to discuss the basic forms of expression that are possible. The basic forms employed in instructional text and correspondingly in IMAGENE are: (1) the relational; (2) the passive; (3) the perfective; and (4) the procedurally obvious.

The relational form can be found in the following text:

(27) The BATTERY LOW INDICATOR will light when the battery in the handset is low. (Excursion, 1989)

The expression "when the battery is low" expresses the relevant resulting condition related to an action, rather than the action itself. This hides the temporally prolonged and possibly temporally remote action of the battery loosing its charge. IMAGENE maintains, in its lexicon, semantic links between actions (drain) and their appropriate predicate adjectives (low). This allows it to find the appropriate relational form for an action should it be specified for use by the system network.

Both the passive and the perfective forms can be found in the first sentence of the Installed-Phone text. The first clause ("When the 7010 is installed") is an agentless passive expression. Currently, there are not enough perfective forms in the corpus to warrant special treatment, so IMAGENE produces the passive form for the second clause as well ("the battery is charged").

Finally, a rhetorically demoted action can be found in example (24). Such expressions of obvious actions, as has already been discussed, are typically may using an "after" prepositional phrase.

5.4 Results

There are two types of results that IMAGENE supports. The first type is not based on an action in the process structure at all, but rather is a span added by the system networks to signal a state. The second type is based on a non-reader action that is not the result of an explicit command to monitor a particular device state. This section discusses these two types of results.

The Installed-Phone process includes the first type of result, called a Go-On-Condition expression ("The phone is now ready to use"). The point of this expression is to signal the readiness of the device for normal usage. It typically occurs after the installation instructions and before the instructions for normal use. In the case of the Installed-Phone text, installing the phone and moving the OFF/STBY/-TALK switch to STBY prepares the unit for use. This can be seen as the expression of both an action and the state which it brings about.

The latter result form is much more common, and expresses a non-reader action that is seen from the point of view of the reader as a result of the actions that they perform. Consider the following example of this:

(28) Battery Low Indicator
The BATTERY LOW INDICATOR will light when the battery in the handset is low. (Excursion, 1989)

Here a future tense form is used to express the action (*"The BATTERY LOW INDICATOR will light"*). Because the draining action is performed by a non-reader and not explicitly being monitored by the reader, the action was demoted to precondition status and was combined with the result expression. This is a case where the descriptive RST analysis (Mann and Thompson, 1987b) would not mark the future tense expression as a result at all, even though the action is very similar to other results and its expression shares most of the grammatical features of result expressions. IMAGENE is able to identify this because it starts with actions, not with rhetorical relations.

5.5 Concurrent Commands

Concurrent expressions arise when there is concurrency represented in the process structure that must be explicitly stated. Whereas at the lowest levels, every process that a person ever performs is concurrent, these levels of detail are seldom expressed. Some other, more detailed instructions, such as the ones for using a hammer found in a set of instructions for house-wives (Livers, 1992), do include this information. It is, therefore, important to include facility for dealing with concurrency when necessary.

Because the analysis performed for IMAGENE has largely addressed local rhetorical relations, that is those rhetorical relations that exist at the lowest levels of the rhetorical structure of an instructional text, IMAGENE deals with only with concurrency of low-level actions. It does not have the facility to address higher-level concurrency such as the concurrency typically expressed with "meanwhile" connectives (see Mellish and Evans 1988 for preliminary work on this). The Installed-Phone process includes no concurrent actions.

Table 5.1: The Frequency of Rhetorical Status Types

Rhetorical Status	Clauses	Phrases	Total Count
Sequence	384	4	388
Purpose	101	18	119
Precondition	91	7	98
Result	59	1	60
Concurrent	31	3	34
Totals	666	33	699

An expression of concurrency can be found in the Furnace text:

Furnace Text:

> Depress knob and hold for 60 seconds after pilot has been lighted. Release knob and turn to ON position.

Here, the knob must be depressed while lighting the pilot light. This example will be discussed in more detail in Chapter 8.

5.6 Sequential Commands

Most actions in instructional text are leaf-level actions prescribed for the reader, and, thus, do not fit into the categories discussed above. These actions are marked as imperative commands. The **Move, Extend-Base,** and **Extend-Handset** actions from the Installed-Phone text are examples of this. These imperatives constitute the majority of the rhetorical status selections for the 699 action expressions in the corpus. Table 5.1 gives a quick sketch of the relative frequencies of each rhetorical status; more will be said about this is Chapter 7.

5.7 Summary

This chapter has discussed the rhetorical status selection systems found in the action systems sub-network. These systems are executed immediately upon entering the action systems for each action, and direct the flow of control in the network to the appropriate grammatical form selection sub-network for the chosen rhetorical status. Further, the feature selections made in these systems typically affect the grammatical form selection process as well, as will be discussed in Chapter 6.

This chapter has shown how IMAGENE can start with simple action nodes in a procedural hierarchy and use functional information to determine what rhetorical status each should have in the resulting rhetorical structure. The details of the remaining grammatical form selection problem are the subject of the next chapter.

Chapter 6

Grammatical Form Selection

This chapter will discuss, in some detail, the systems that affect the selection of the grammatical form of each action expression. The previous chapter, Chapter 5, set the stage for this by discussing the systems that affect the selection of the rhetorical status of the expression. As mentioned in the previous chapter, the break between these two groups of systems is not a clean one. They are all part of the same network and there are occasions when the systems discussed in the rhetorical status selection chapter determine grammatical form and, conversely, when the systems discussed in this chapter determine the rhetorical status. The distinction, nonetheless, is interesting theoretically as well as useful pedagogically and will be retained in this thesis.

This chapter will discuss each of the fundamental relations used in the study: purpose, precondition, result, concurrency, and sequence, particularly with respect to how the action systems sub-network maps functional features of the environment onto lexico-grammatical form. It should be noted that there is some complexity in the network that will be abstracted away in this chapter as was the case in the last. An attempt to discuss every system in detail would become overly tedious and time consuming. Because of this, a full list of the functional features that these systems encode, for both rhetorical status selection and grammatical form selection, is given in Appendix D.

6.1 Purpose Relations in Instructional Text

Chapter 5 indicated that there is a single functional context in which purposes occur. This is the context where IMAGENE is expressing an action which has a generation relation with a set of other actions (called its sub-actions). The task, therefore, of the rhetorical status selection systems is a simple matter of looking at the structure of the plan being expressed, a matter of the ideational content of the process itself. The task of the grammatical form selection systems for purpose, however, is more complex. There are a large number of lexico-grammatical forms in which purposes are typically expressed, each used in a particular functional context. This section discusses the system networks that have been included in IMAGENE to differentiate among these forms.[1]

The first step in analyzing lexico-grammatic form selection, as discussed in the corpus analysis chapter, Chapter 3, is to identify the range of commonly occurring forms in the corpus for the relation in question, in this case purpose. There have been a large number of studies of the grammatical aspects of "purposes" in linguistics as noted in Chapter 2. The current study has focussed on the forms of purpose used almost exclusively in instructional text, called the rationale clauses by Huettner (1989). The two syntactic tests that she gives for identifying this particular form are: (1) the "to" may be replaced by "in order to", and (2) the clause may be fronted. The current study has looked at a number of rationale clauses, as well as some lexico-grammatical forms not specifically covered by this test. Here

[1]The system network discussed in this section is a revised version of a network described in an earlier publication (Vander Linden et al., 1992b).

are representative examples of all of them:

(29a) *To end a previous call,* hold down FLASH [6] for about two seconds, then release it. (Code-a-phone, 1989)

(29b) Follow the steps in the illustration below, *for desk installation.* (Code-a-phone, 1989)

(29c) The OFF position is primarily used *for charging the batteries.* (Code-a-phone, 1989)

(29d) *For frequently busy numbers,* you'll want to use REDIAL [7], and the pause will have to be in Redial memory. (Code-a-phone, 1989)

(29e) When instructed (approx. 10 sec.) *remove phone by* firmly grasping top of handset and pulling out. (Airfone, 1991)

(29f) *Return handset to wall unit from which it was taken.* Insert heel first as shown, then push top in firmly. (Airfone, 1991)

(29g) Tilt pan down slightly at the rear *so that the fluid drains out.* (Reader's Digest, 1981)

There are two issues of choice at the lexico-grammatical level in these examples. First, the purpose expressions can occur either before or after the actions which they pertain to. This is the issue of slot. Secondly, there are seven grammatical forms to choose from.

- Example (29a) uses a "to" infinitive form (TNF);
- Example (29b) uses a "for" prepositional phrase with a nominalization ("installation") as the complement;
- Example (29c) uses a "for" preposition with a gerund phrase as the complement;
- Example (29d) uses a "for" preposition with a noun phrase that refers to the goal of the corresponding action as the complement. This is termed a *Goal Metonymy;*
- Example (29e) uses a simple imperative for the purpose with "by" conjoining participial forms of the intended actions;
- Example (29f) uses a simple imperative for the purpose, with the intended actions in a separate sentence following the purpose.
- Example (29g) uses a simple imperative for the with "so that" conjoining present tense action forms of the intended actions at the end;

The frequency with which these slots and forms appeared in the 119 purpose expressions in the corpus is given in Table 6.1. The four examples in the category labeled "Other" were similar in form to the following, and not addressed by IMAGENE:

(30a) Your call may be ended in any of the following manners: .... (Excursion, 1989)

(30b) If the valve keepers are hard to remove, *they can be loosened by* ... (Volkswagon, 1987)

There are other studies of purpose expressions from the point of view of representation and understanding which are of use here (Di Eugenio, 1992; Balkanski, 1992), but unlike them, the current study is critically interested in discerning principled reasons for choosing between these various slots and forms in context. Di Eugenio, for example, has worked with "by" purposes and "to" infinitive (TNF) purposes, but doesn't appear to have distinguished the two forms in her analysis of the procedural relationships between actions. She is, therefore, concerned with diversity of lexico-grammatical form only to the extent that the parser/understander be prepared to recognize a purpose when it sees one, but not to distinguish the forms. This is an issue upon which IMAGENE critically depends; it must be capable of detailing the reasons to choose one form over the other.

Conversely, there are certainly issues for which Di Eugenio's analysis must be more detailed than IMAGENE's. For example, she is very interested in the use of purpose expressions to specify underdetermined sub-action expressions, as in: "Place a plank between two ladders to make a simple

Table 6.1: The frequency of various form and slot combinations of purposes in instructional text

	Initial	Final
To-Infinitive	38	33
For-Nominalization	2	7
For-Gerund	0	3
For-Goal-Metonymy	1	5
By-Purpose	11	1
Adjoined-Purpose	4	0
So-That-Purpose	0	10
Other	4	

scaffold." (1992, page 120). Here, the action of placing a ladder between two ladders clearly underspecifies the task of making a simple scaffold. Although the issue of action sufficiency has played some role in the current study, it has not been found to have a major effect on grammatical form. Di Eugenio's example, for instance, is a simple "to" infinitive in final position, just like many of the rest of the purposes. This case illustrates the difference in concern between studies of understanding and of generation in the context of purpose clauses.

This section now discusses the issues of choosing the slot and the form of purpose expressions. These issues are treated largely independently by IMAGENE. The slot is determined by the sub-network shown in Figure 6.1. The form is determined by the sub-network in Figure 6.3. This purpose sub-network as a whole formalizes Thompson's notion of the "vastly different functions" for initial and final purpose clauses (Thompson, 1985) in the context of instructional text. In keeping with her analysis, the form selection sub-network (figure 6.3) is largely independent of the slot selection sub-network. Taken together, they generate a greater range of purpose expressions than is typical in generation systems (see e.g. Hovy, 1988b) and identify the functional reasons for choosing one form over the other.

6.1.1 Purpose Slot

The slot selection sub-network, shown in figure 6.1, places the purpose expression in the final position in most cases. This matches the data given in Table 6.1, where the majority of the purpose expressions are in the final position. The exceptions to this are when the scope of the purpose is global, the purpose is considered optional, or the purpose is considered contrastive. These three exceptions are handled by the three systems depicted in the figure.

The first exception, handled by the **Scope** system, concerns the number of actions the purpose pertains to. One could not, for instance, restate (29a) as "?? Hold down FLASH [6] for about two seconds, then release it *to end a previous call.*" As Thompson points out (1985), the purpose clause is often used, as it is here, to state a context in which the prescribed actions are to be interpreted, and thus should be fronted. The restatement also implies, incorrectly, that the purpose applies to the last action alone rather than to the sequence of actions. The **Scope** system does not actually set the textual order of the purpose expression; it constructs the rhetorical structure and allows later systems to set the textual order. For example, in the Remove-Phone text it transforms the hierarchical structure shown in Figure 6.2A, into an RST-like structure, as shown in Figure 6.2B.

The remaining exceptions occur when the purpose is considered optional or contrastive and are handled by **Optionality** and **Contrastiveness** respectively. Here are examples of them from the corpus:

(31) *For more information and wall installation instructions*, see the Installation Notes on page 3. (Code-a-phone, 1989)

(32) To place call, dial AREA CODE and NUMBER. *To end call*, press red HANG UP button. (Airfone, 1991)

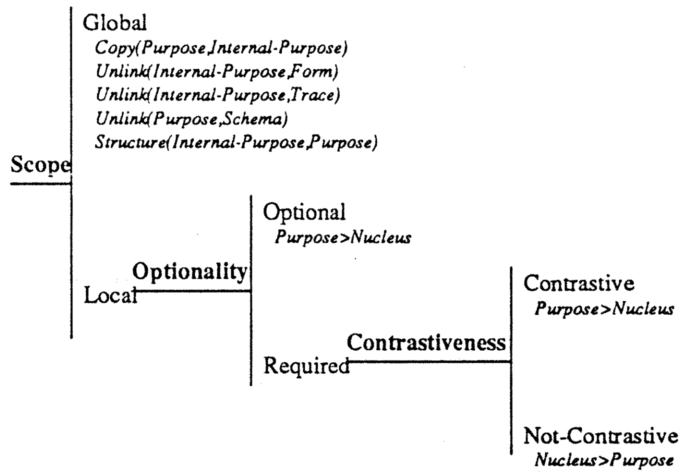

Figure 6.1: The Purpose Slot Selection Network

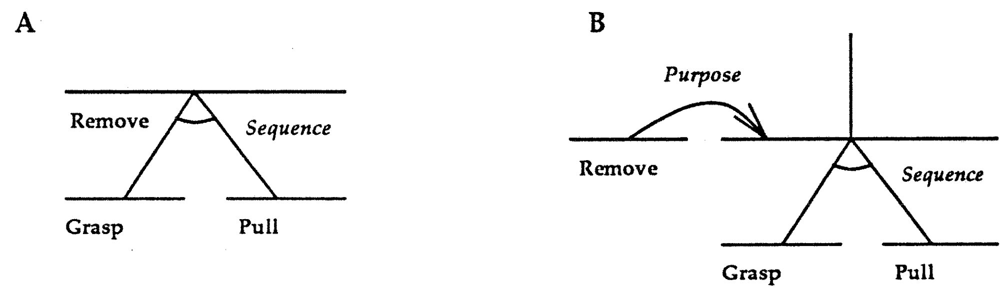

Figure 6.2: A Structural View of Purpose Demotion

In (31), the purpose action is optional, that is, the reader may or may not want more information at this point in the text. The purpose expression is, therefore, stated first to set the appropriate context for interpreting the prescribed sub-action. In (32), the purpose of ending a call is stated in contrast to placing a call in the previous sentence. It is thus fronted to set the appropriate context for the prescribed action. One other exception to this will be covered in the next section with the discussion of **Conditional-Status.**

The results of this study predict a number of cases where purposes should not be fronted, which is in contradiction of the general claim made by Dixon (1987a). He claimed that purposes should always be fronted because this facilitates the top-down construction of a procedural plan by readers as they go through the text. The current study shows cases where this rule is not followed by technical writers.

6.1.2 Purpose Form

The form selection sub-network, shown in figures 6.3 and 6.4, determines the grammatical form of purpose expressions. The first element of the form selection sub-network is **Conditional-Status** which determines if the high-level purpose being expressed has special conditions or destinations pertaining to it. If so, either a "by" purpose or an adjoined purpose expression are used, depending upon the complexity of the resulting sentence as determined by **Sentence-Complexity.** The slot of these forms is always initial and is determined here, rather than in the slot selection sub-network just discussed. If the resulting sentence has more than five propositions involved in it, the adjoined form is used, otherwise the "by" purpose form is used. Consider the following examples of these situations:

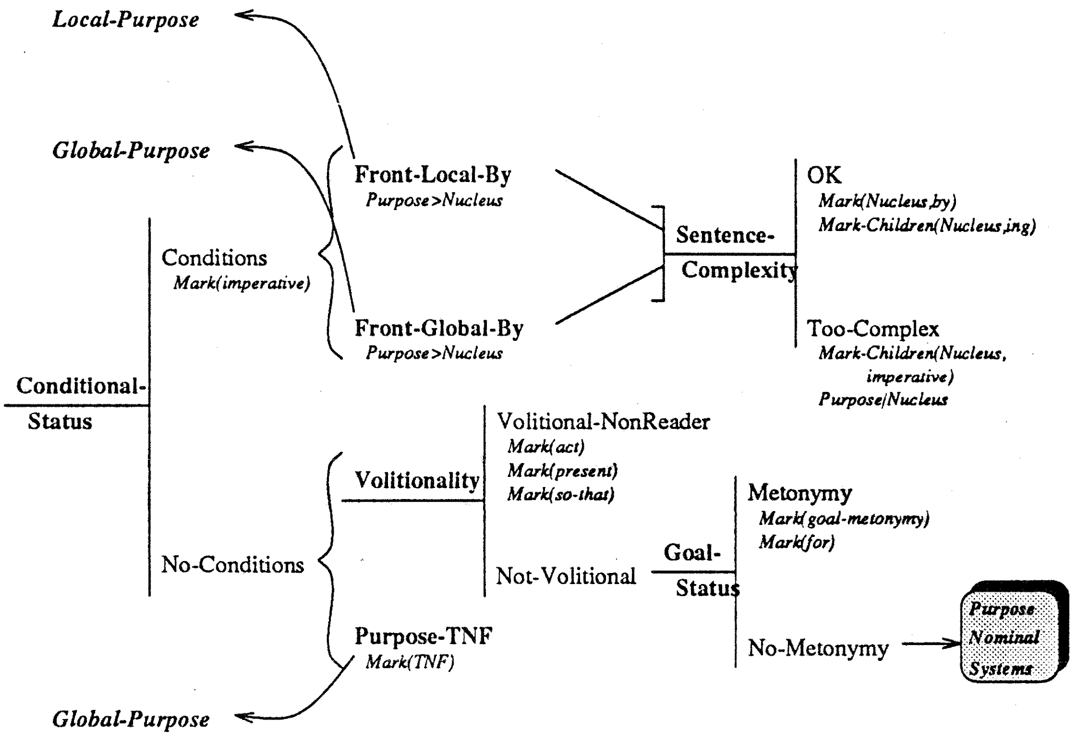

Figure 6.3: High-Level Systems for the Purpose Form System Network

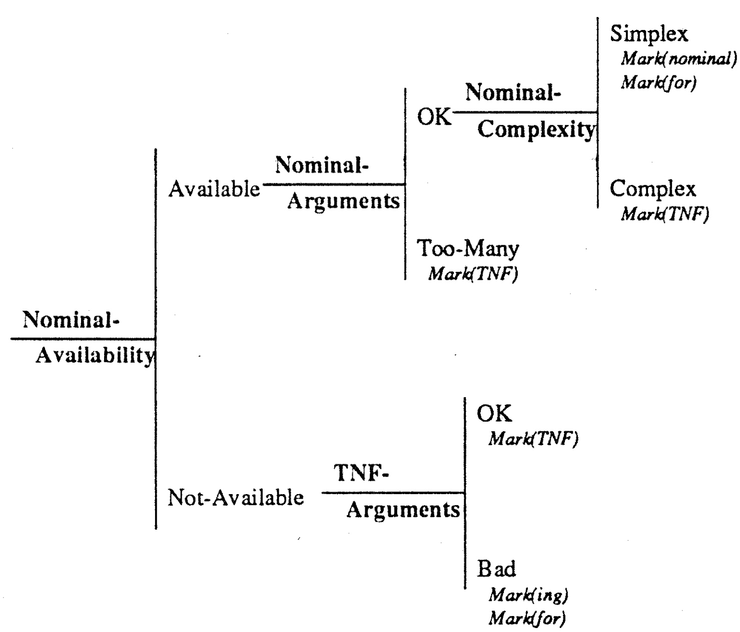

Figure 6.4: Nominalization Systems of the Purpose Form System Network

(33a) When instructed (approx. 10 sec.) *remove phone by* firmly grasping top of handset and pulling out. (Airfone, 1991)

(33b) ?? *To remove phone* when instructed (approx. 10 sec.), firmly grasp top of handset and pull out.

(34a) Return handset to wall unit from which it was taken. Insert heel first as shown, then push top in firmly. (Airfone, 1991)

(34b) ?? Return handset to wall unit from which it was taken by inserting heel first as shown, then pushing top in firmly.

In example (33a), there is a precondition on the high-level purpose of removing the phone, a feature that correlates well with the use of the "by" form. Example (33b) makes the incorrect implication that the prescribed actions only work "when instructed." In the second example, the "by" form was desired, but the number of propositions in the resulting sentence, (34b), was too great, forcing the use of the adjoined form, (34a). This use of the adjoined purpose form is an example of a case where the rhetorical structure of a text need not be explicitly signaled with a lexical or grammatical cue (except textual order), called an "inferred connective" by Crothers (Crothers, 1979). RST allows the representation of this situation because its relations are not defined in terms of lexico-grammatical forms (Mann and Thompson, 1989).

When a purpose does not have conditions upon it and the scope is global, **Purpose-TNF** marks the purpose as a "to" infinitive (TNF). The fact that the **Global-Purpose** is required as well as the input conditions represented normally in the figure is indicated with an arrow pointing to the additional input conditions. This simplification of notation makes the diagrams easier to segment and present in graphic form. Example (29a) illustrated this. These sorts of context-setting purposes are not demoted to phrase status.

The **Volitionality** system determines if the purpose expresses the desire of the reader to get some inanimate substance to perform in some way. This context usually leads to the use of the "so that" purpose as shown in example (29g). Quite often these substances are liquids, but may also include other inanimates. The key point is that the action of the reader is intended to get the substance to perform is some way. Consider the following alternate forms for expressing purpose:

(35a) Sit the person up leaning slightly forward *so that blood and saliva can drain from his mouth.* (Rosenberg, 1985)

(35b) ?? Sit the person up leaning slightly forward in order to allow blood and saliva to drain from his mouth.

Both forms present the same basic information, but the form in example (35a) is most commonly used in the corpus. Huettner (1989) identifies this "so that" form as a tensed rational clause.

Goal-Status determines if the use of Goal Metonymy is warranted. It occurs in purposes where the goal of the purpose clause is more important than the action, as in:

(36) *For frequently busy numbers*, you'll want to use REDIAL [7], and the pause will have to be in Redial memory. (Code-a-phone, 1989)

This is an ellipsis where the full purpose would be something like "to handle frequently busy numbers" or "for dealing with frequently busy numbers". The object (or goal) of the verb, in this case the busy numbers, is considered more important than the verb itself, allowing the metonymy.

The remainder of the form selection sub-network, shown in figure 6.4, is based on a general characteristic of instructional text — it is oriented toward reader actions. Because purposes are not explicit reader actions, they tend to be demoted to phrase status whenever possible. Thus, **Nominal-Availability** will realize a prepositional phrase with a nominalization as the complement whenever the appropriate nominalization exists. The form selection sub-network currently represents three discrete points along the continuum from fully nominal to fully verbal forms expressing the same action (Quirk et al., 1985), namely the nominalization, the gerund, and "to" infinitive. These are the forms that currently exist in the corpus.

This analysis of nominalizations is an example of the descriptive nature of the current study of instructional text. The descriptive observation has been made that when nominalized forms of a verb exist in the lexicon, they tend to be used. A full explanatory account, in the spirit of current Discourse-Functional studies, would attempt to identify the precise aspect of the action or the context of its expression which would dictate the use of a nominalization, thus resulting in the development of a nominalized form in the English Language. Such an account is beyond the scope of the current study.

This use of phrases with nominalizations as propositional, rhetorical units is common in the instructional text used in the current study, as well as in academic text (Cumming, 1991) and formal text in general (Hovy, 1987). Feraci (1974), in fact, has argued that, for grammatical reasons, the "for" phrase with a sentential object is a better deep representation of purpose, rationale, and objective infinitives. IMAGENE's architecture, as was claimed in Chapter 2 and can be seen now, implements a particular interpretation of Cumming's proposal that nominalizations be dealt with at two levels, one at which the actions are not specified for nominal or clausal expression, and another where they are. IMAGENE's Process Structure can be seen as the former level, its Text Structure as the latter.

Even if a nominalization exists, however, it still may not be used depending upon the determination of **Nominal-Arguments** and **Nominal-Complexity**. These systems, based on the examples in our corpus, restrict nominalizations to single, non-complex arguments. Consider the following examples:

(37a) Use the VOL LO/HI [2] switch *to adjust volume to your preferred listing level.* (Code-a-phone, 1989)

(37b) ?? Use the VOL LO/HI [2] switch *for volume adjustment to your preferred listening level.*

(38a) FLASH uses proper timing *to avoid an accidental hangup.* (Code-a-phone, 1989)

(38b) ?? FLASH uses proper timing *for accidental hangup avoidance.*

In cases (37a) and (38a), taken from the corpus, there were nominalizations available, namely "adjustment" and "avoidance", but neither was used. The "adjustment" nominalization in (37b) was not used because it required more than one argument. The "avoidance" nominalization in (38b) was rejected because the argument "accidental hangup" was itself a nominalization and thus too complex. In both cases, the "to" infinitive form was preferred.

If no nominalization is available, **TNF-Arguments** will typically produce the "to" infinitive (TNF), unless the infinitive form requires unwanted arguments. Here is an example of this case:

(39a) The BATT LOW Light [9] comes ON when the battery is weak. The handset must be returned to the base *for recharging.* (Code-a-phone, 1989)

(39b) ?? The BATT LOW Light [9] comes ON when the battery is weak. The handset must be returned to the base *to recharge (the battery?).*

Examples similar to (39a) were found in the corpus, while those similar to the alternate "to" infinitive expression, (39b) were not. This latter example seems suspect because one would be required to restate "the battery" which is unacceptable in this context.

6.2 Precondition Relations in Instructional Text

Chapter 5 indicated that there are a number of functional contexts in which actions are expressed as preconditions, rather than as imperatives, characterizing these contexts in terms of their unique functional features. The task of the grammatical form selection systems for precondition, is no less complex. As with the purposes discussed in the previous section, there are a large number of lexico-grammatical forms in which preconditions are typically expressed. This section discusses the system networks that have been included in IMAGENE to differentiate among these forms. Here are representative examples of the precondition expressions found in the corpus:

(40a) *If light flashes red,* insert credit card again. (Airfone, 1991)

(40b) *When the 7010 is installed* and the battery has charged for twelve hours, move the OFF/STBY/TALK [8] switch to STBY. (Code-a-phone, 1989)

Table 6.2: The frequency of various form and slot combinations of Preconditions in instructional text

	Initial	Final
Present Action	22	5
Present Passive	5	7
Present Relational	14	11
Phrasal	2	5
Imperative Sentence	3	0
Other	24	

(40c) The BATTERY LOW INDICATOR will light *when the battery in the handset is low.* **(Excursion, 1989)**

(40d) *If you leave the OFF/STBY/TALK [8] switch in TALK,* move the switch to PULSE, ... **(Code-a-phone, 1989)**

(40e) Return the OFF/STBY/TALK switch to STBY *after your call.* **(Code-a-phone, 1989)**

Each precondition is based on an action that would normally have been expressed with an imperative, active clause, but for various reasons has been rhetorically demoted to a precondition status. As a precondition, any action can be stated either in terms of the action itself or the relevant condition brought about by that action.

Example (40a) contains an action expression, "If light flashes red". Action statements such as this can either be expressed as present tense action clauses, as in this case, or as stative passives (defined in Quirk et al., 1985) as shown in example (40b) ("When the 7010 is installed") depending upon the type of action being expressed. Example (40d) shows another case where the present tense active form is used, in this case to express a decision to make a practice of leaving the OFF/STBY/TALK switch in TALK position.

Example (40c) contains the conditional form, "When the battery in the handset is low". This is an expression of the relevant condition brought about by the action of "draining the battery", called the *Terminating Condition* because it states the condition which is true following the termination of the action. Terminating conditions are typically expressed as relational clauses as seen in this example. This relation between actions and conditions is common in the automated planning literature where the conditions are typically called postconditions, and has been discussed in the psychological literature as well (Dixon, 1987a).

Example (40e) contains an action expressed as a precondition which is neither fronted nor in clause form. This form of rhetorical demotion was discussed in Chapter 5.

The examples just given do not represent every expressional form in the corpus for precondition, but do illustrate the three major issues of choice, the choices of slot, linker, and form. The frequency with which these slots and forms appeared in the 98 precondition expressions in the corpus is given in Table 6.2. The forms of precondition expressions, as can be inferred from the relatively large number of examples marked as "Other" in the chart, are much more diverse than either purpose expressions or the result expressions discussed in the next section. The examples of type "Other" include a wide range of other lexico-grammatical forms currently not addressed by IMAGENE, including gerunds, future tenses, and perfect tenses. The following sections detail the system networks designed to automate these choices.

6.2.1 Precondition Slot

The slot selection sub-network, shown in figure 6.5, determines whether or not to front a particular precondition expression. In the corpus, preconditions are typically fronted, and therefore the subnetwork will default to fronting. There are four exceptions to this default, but only one is dealt with by the systems shown in the slot selection sub-network. The other three are dealt with in other systems within the precondition sub-network in conjunction with linker and form selection.

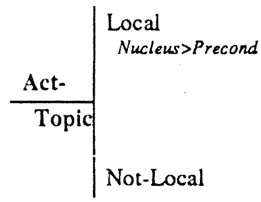

Figure 6.5: The Precondition Slot Selection Network

(41a) The BATTERY LOW INDICATOR will light *when the battery is the handset is low.* (Excursion, 1989)
(41b) Return the OFF/STBY/TALK switch to STBY *after your call.* (Code-a-phone, 1989)
(41c) The phone will ring *only if the handset is on the base.* (Code-a-phone, 1989)
(41d) In the STBY (standby) position, the phone will ring *whether the handset is on the base or in another location.* (Code-a-phone, 1989)

The slot selection for example (41a) could go either way, except that it is the first sentence in a section titled "Battery Low Indicator", making the discussion of this indicator the local topic of conversation, and thus the appropriate theme of the sentence. This distinction is made by **Act-Topic**, the single system shown in Figure 6.5. There is no ordering realization statement if the act is not thematic because the structure systems have already ordered the expressions in procedural order. **Act-Topic** happens to be the only system in the precondition sub-network that deals exclusively with slot. All the other precondition systems dealing with slot deal with issues of form or linker as well and are thus not included in figure 6.5. They will be discussed in later sections.

Example (41b) is the example of rhetorical demotion discussed several times in this thesis. The action is considered obvious and is thus demoted to phrase status and put at the end of its immediately following action. Jou and Harris (1990) present an interesting psychological study of this type of phenomenon. They tested the relative efficacy of pairs of sentences like: (a) "While John played, Mary sang." and (b) "Mary sang while John played." The results favored the main-clause first rule, used in (b). This shows Psycholinguistic evidence for the expression of the precondition in the final slot even if it puts the action expressions out of order. Chapter 8 will discuss the Furnace text, a passage which used such a demotion in an inappropriate context.

Examples (41c) and (41d) show preconditions that are not fronted because of the syntax used to express the logical nature of the precondition. In (41c), the condition is expressed as an exclusive condition which is never fronted. One could, perhaps, say "Only if the handset is on the base, will the phone ring.", but this form is never used in the corpus. Neither is the condition in (41d) ever fronted in the corpus. This form lists all the possible alternatives, making itself automatically true.

Although preconditions are typically expressed as fronted clauses, this section has discussed a number of cases where preconditions are not fronted. These can be seen as cases where Dixon's claim that explicit actions should come first (Dixon, 1982) is violated by writers instructional text. The current study claims that writers have good reasons for not fronting a precondition expression, reasons that are based on the discourse and grammatical factors that have been discussed in this section.

6.2.2 Precondition Linker

The precondition linker selection sub-network, see figure 6.6, determines which linker to use for the precondition. Two parallel systems are entered, **Condition-Probability** and **Changeable-Type.** The determinations of the these systems, along with the determinations already made by the rhetorical status selection systems discussed in Chapter 5, help determine the linker to be used.

Condition-Probability determines how probable a particular action is in the procedural context. Highly probable actions are typically marked with "when". Those actions which are not highly probably are marked with "If" or some similar linker, as determined by the **Complexity** system and its descendants.

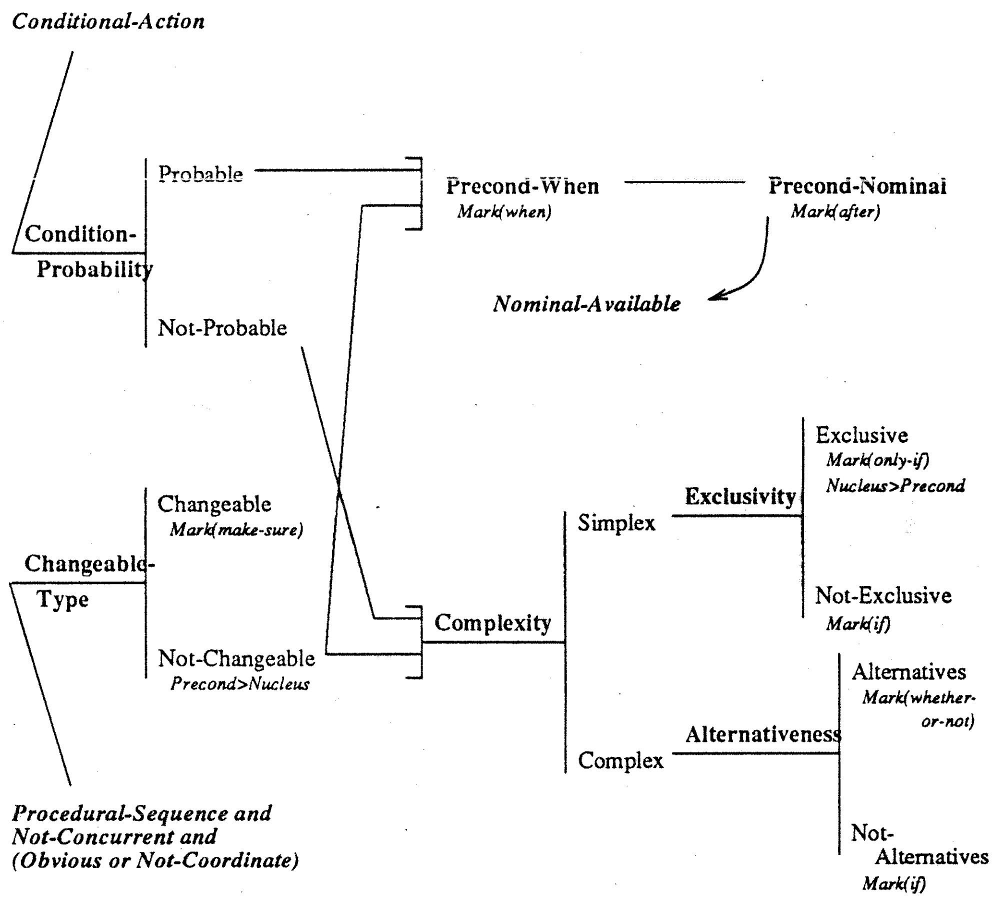

Figure 6.6: The Precondition Linker Selection Network

Precond-When is entered when the action is conditional and further is highly probable, a distinction made by the **Condition-Probable** system shown in Figure 5.2 in the previous chapter. This commonly occurs when the precondition is part of an ongoing sequence of actions prescribed by the instructions; it constitutes the bulk of the preconditions in the corpus. Here is an example:

(42) Lift the handset and set the OFF/STBY/TALK [8] switch to TALK. *When you hear dial tone,* dial the number on the Dialpad [4]. (Code-a-phone, 1989)

The precondition in (42) occurs in a sequence of actions and is conditional in that it may not actually happen, say if the telephone system is malfunctioning in some way, but is none the less highly probable. **Precond-Nominal** is entered immediately after **Precond-When** whenever the precondition is being stated as a nominalization. It overwrites the linker choice with "after" in only this case.

Preconditions that the user is expected to be able to change if necessary and which come at the beginning of sections that contain sequences of prescribed actions are called *Changeable* preconditions. They tend to be specially marked. Here is an example:

(43) 1. *Make sure the handset and base antennas are fully extended.* 2. Set the OFF/STBY/TALK SWITCH to Talk. (Excursion, 1989)

This text comes at the beginning of a primitive-type section on placing a call and is stated to ensure that the phone is ready for this operation. The reader is expected to be able to determine if the precondition is true, and if it is not, to perform an unspecified action to ensure that it is. In example (43), the reader will check the antennas and extend them if they are not already extended. This type of precondition is marked as a "Make sure" imperative clause by **Changeable-Type**.

The **Complexity** system is entered for actions which are not probable and not changeable. It determines the logical nature of the preconditions and sets the linker accordingly. The three possible linkers chosen by this sub-network are "if", "only if", or "whether ...or ...". Here are examples of them:

(44a) The phone will ring *only if the handset is in the base*. (Code-a-phone, 1989)

(44b) *If you have touch-tone service*, move the TONE/PULSE SWITCH to the Tone position. (Excursion, 1989)

(44c) In the STBY (standby) position, the telephone will ring *whether the handset is on the base or in another location*. (Code-a-phone, 1989)

Example (44a) is an exclusive condition, that is, the phone will ring if the precondition is true, and will not ring if it is false. This distinction is made by the **Exclusivity** system and it selects the slot as well as the linker. The slot of such preconditions is always after their actions. Example (44b) contains the standard "if" condition. Example (44c) contains a complex precondition where the multiple alternatives cover all the logically possible values of the condition. This type of precondition, identified by the **Alternativeness** system, is always true and is never fronted by the slot selection sub-network. The feature Not-Alternatives is never used in the corpus, but was included in the **Alternativeness** system for logical completeness.

6.2.3 Precondition Form

As noted above, preconditions can be expressed as either a terminating condition or as an action. The choice between the two is made by the form selection sub-network, shown in figures 6.7 and 6.8. This choice depends largely upon the type of action on which the precondition is based. The actions in the corpus can be divided into five categories which effect the grammatical form of precondition expressions:

- Monitor Actions;
- Giving Actions;
- Habitual Decisions;
- Placing Actions;
- Other Actions.

The first four actions are special categories of actions that have varying act and terminating condition forms of expression. The last category, other actions, encompasses all actions not falling into the previous four categories and comprises the largest share of precondition actions in the corpus. This section will discuss each category in turn, starting with the following examples of the first four action types:

(45a) *Listen for dial tone*, then dial AREA CODE + NUMBER slowly. (Airfone, 1991)

(45b) *If you have touch-tone service*, move the TONE/PULSE SWITCH to the Tone position. (Excursion, 1989)

(45c) *If you leave the OFF/STBY/TALK [8] switch in TALK*, move the switch to PULSE, and tap FLASH [6] the next time you lift the handset, to return to PULSE dialing mode. (Code-a-phone, 1989)

(45d) The phone will ring *only if the handset is on the base*. (Code-a-phone, 1989)

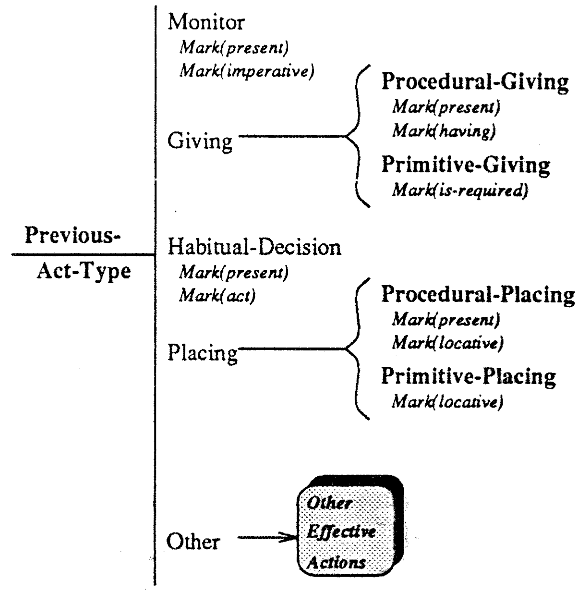

Figure 6.7: The Precondition Form Selection Network

Monitor actions, as shown in (45a), concern explicit commands to monitor conditions in the environment. In this case, readers are being commanded to listen for a dial tone, with the underlying assumption that they will not continue on with the instructions unless one is heard.

Similarly, IMAGENE expresses giving and placing actions as terminating conditions, shown in (45b) and (45d). The corpus does not include active forms of these actions, such as "?? *If the phone company has given you touch-tone service,* do . . ." or "?? Do . . . *if you have placed the handset on the base.*"

An *Habitual decision* is a decision to make a practice of performing some action or of performing an action in some way. When stated as preconditions, they take the present tense form in (45c). This expression refers not to a singular action of leaving the OFF/STBY/TALK switch in TALK position, but rather to the decision to habitually leave it is such a state. The singular event would be expressed as "*If you have left the OFF/STBY/TALK switch in TALK,* do . . ." which means something quite different from the expression in (45c) which is stated in present tense.

The bulk of the preconditions in the corpus are based on other types of actions, that is, some other type of material action. The precondition form selection sub-system determines the appropriate form for them as well. In general, the other action systems are based on the actor of the action. Reader actions are expressed either as present tense passives or as present tense actions, depending upon whether the action is a repeated mention or not. These determinations are made by the gates **Repeated-Reader** and **Not-Repeated-Reader**. An example of the former can be found in the first clause of the Installed-Phone text, "When the 7010 is installed". Here, the present tense, agentless passive is used because the action, as discussed in Chapter 1, is a repeated mention. If the reader action is not a repeated mention, a simple present tense active form is used as in:

(46) *If you make a dialing error,* or want to make another call immediately, FLASH gives you new dial tone without moving the OFF/STBY/TALK switch. (Code-a-phone, 1989)

The **Act-Hide** system and its descendants are entered for non-obvious, non-reader actions. There are four basic forms for these precondition expressions, examples of which are shown here:

(47a) *If light flashes red,* insert credit card again (Airfone, 1991)

(47b) *When you hear dial tone,* dial the number on the Dialpad [4]. (Code-a-phone, 1989)

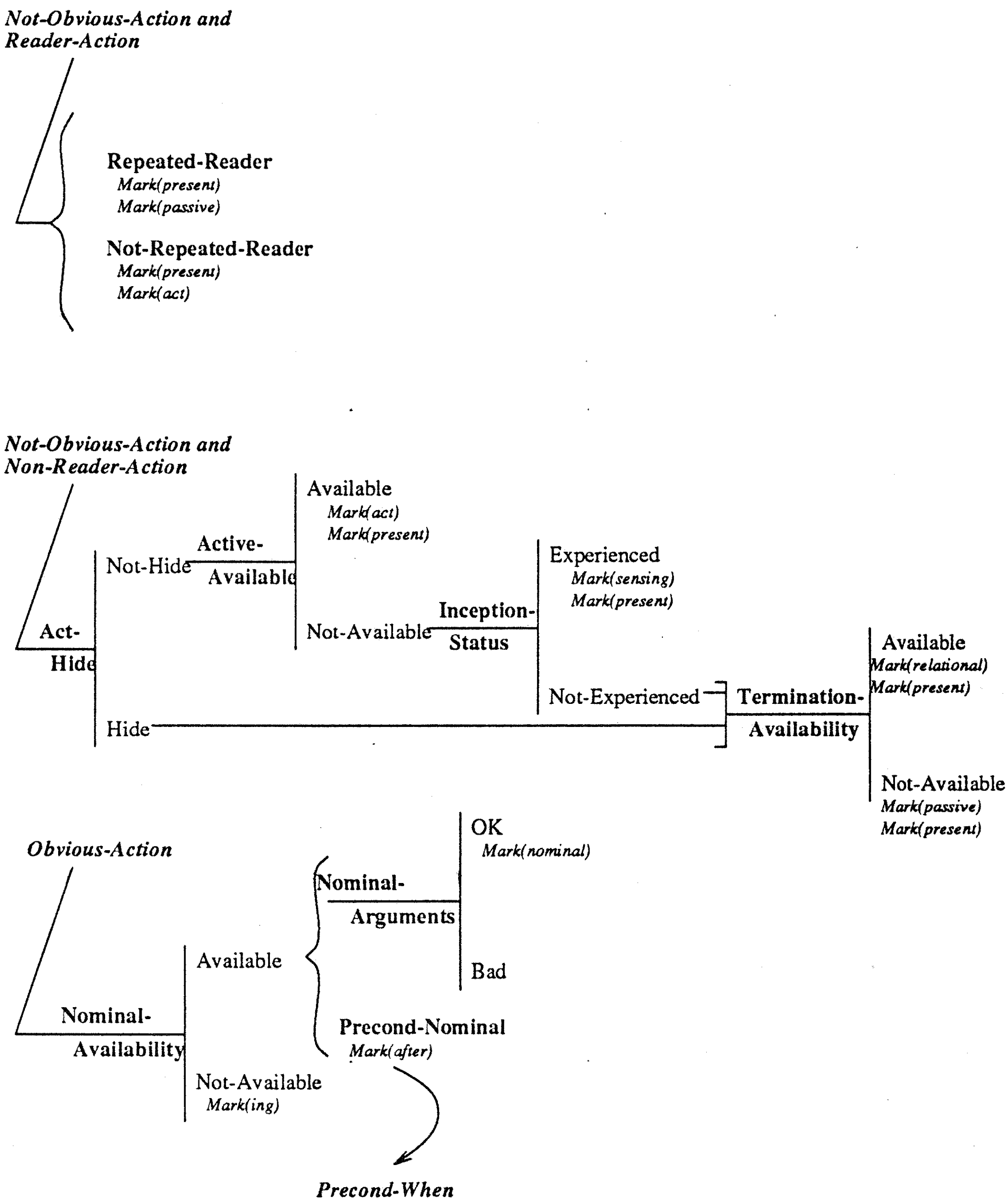

Figure 6.8: The Other Effective Actions Selection Network

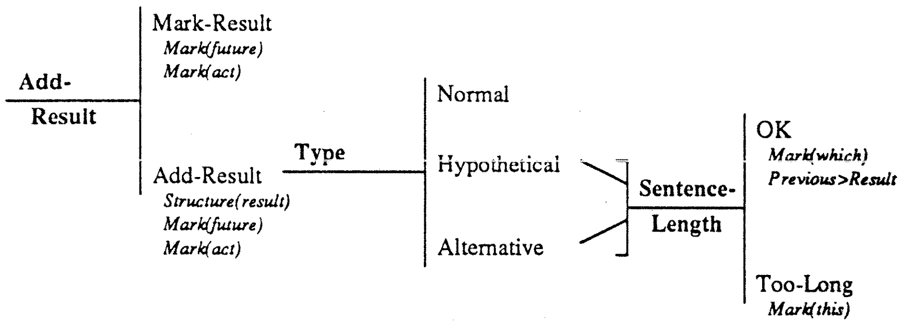

Figure 6.9: The Result System Network

(47c) The BATTERY LOW INDICATOR will light *when the battery in the handset is low.* (Excursion, 1989)

(47d) *When instructed (approx. 10 sec.)* remove phone by firmly grasping top of handset and pulling out. (Airfone, 1991)

Act-Hide distinguishes actions which are overly complex or long duration and those that are not. Those which are not will be expressed either as present tense actions, as the one in example (47a), if the action form is available in the lexico-grammar. **Active-Available** makes this determination. If no action form is available, then **Inception-Status** is entered. If the inception of the action is expected to have been witnessed by the reader, then the present tense sensing action form is used, as shown in example (47b).

Termination-Availability is entered either if the action is to be hidden or if the inception of the action was not expected to be experienced by the reader. In these cases, the relational form of the terminating condition is used if it is available. An example of this is shown in example (47c). The long duration action of the battery draining is not expressed in the relational form used there. If the relational form is not available, the present tense, agentless passive is specified, as shown in example (47d).

If the action is an obvious action, the **Nominal-Availability** system is entered. This system and its children are very similar to the nominalization systems is the purpose network. If the nominalization is available, it is used if the arguments do not violate the conditions discussed in the purpose section. Example (48) shows a nominalized precondition form:

(48) Lift the handset and set the OFF/STBY/TALK [8] switch to TALK. Return the OFF/STBY/TALK switch to STBY *after your call.* (Code-a-phone, 1989)

6.3 Result Relations in Instructional Text

Chapter 5 indicated that there are two functional contexts in which results appear. One of these types of expression is based on a non-reader action in the process structure and the other, called a *Go-On Condition,* is not based on an action at all. The former is handled by the systems shown in Figure 6.9 and the latter by systems from the rhetorical status selection shown in the previous chapter in Figure 5.2. Here are representative examples of the two types of results from the corpus:

(49a) 3. Place the handset in the base. *The BATTERY CHARGE INDICATOR will light.* (Excursion, 1989)

(49b) 5. After 24 hours, connect the telephone line cord and your is EXCURSION is ready to use. (Excursion, 1989)

This section will discuss these forms and when they are chosen. The slot for result relations is always final, as seen in example (49), so the result network exclusively addresses the form of the result. The frequency with which these slots and forms appeared in the 59 result clauses in the corpus is given in

Table 6.3: The frequency of various form and slot combinations of results in instructional text

	Initial	Final
Future/Action	0	21
Relational	0	5
Until	0	18
Present/Action	0	13
Other	2	

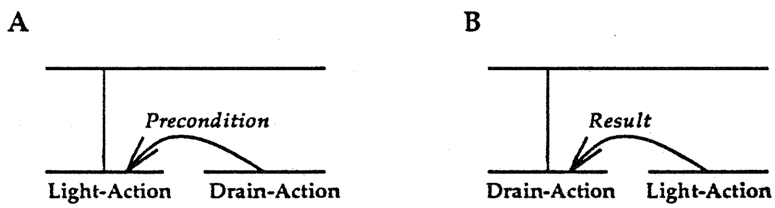

Figure 6.10: Two Rhetorical Forms for Non-Reader Actions

Table 6.3.[2] The two examples in the category labeled "Other" were similar in form to the following, and not addressed by IMAGENE:

(50) The last number you dialed (up to 32 digits) is automatically redialed. (Code-a-phone, 1989)

The first system in the result network, **Add-Result**, is a preliminary system used to determine the result's rhetorical context. Some results are not the standard satellite node in the TRL, but are rather the nucleus of some other relation, normally a precondition relation. The following text is such a case:

(51) The BATTERY LOW INDICATOR will light when the battery in the handset is low. (Excursion, 1989)

Here, the rhetorical structure is like that shown in figure 6.10A rather than the normal result case shown in 6.10B. The text corresponding to example 6.10B would be something like "The battery drains down. The BATTERY LOW INDICATOR will light." or perhaps "Drain the battery. The BATTERY LOW INDICATOR will light." The point is that the lexico-grammatical form of the **Light-Action** in figure 6.10A has much in common with the standard result type even though is not marked rhetorically as a result. Because IMAGENE begins with actions and proceeds through rhetorical structures to lexico-grammatical forms, it is able to recognize the common ground and act accordingly.

Beyond this special case, the form tends to be a future tense, material action clause as in example (49a). This form is not the only form the result system network specifies, but is, none the less, so much more common that **Add-Result** immediately specifies it, allowing **Type** and **Sentence-Length** to override it in special cases. This use of future tense marks the expression as irrealis and functions to ensure that readers don't mistakenly think that they must perform the action. As can be seen from Table 6.3, there are a number of present tense result expressions that the current analysis does not handle. No functional explanation for why they are not future tense is readily apparent.

For those cases that are not simple future tense results, **Type** determines the type of action that the result pertains to. For "Normal" actions, nothing is changed. *Hypothetical* and *Alternative* actions, on the other hand, are handled differently. Hypothetical actions are those that may or may not happen to the reader, and Alternative actions, those that the reader may or may not choose to execute. Because they

[2] "Until" expressions are included in this table because they were labeled as results in the descriptive analysis of the corpus. IMAGENE does not actually generate them as results because they have so few lexical and grammatical similarities with other types of results. Rather, it deals with them as special cases of imperative commands, as will be discussed in the next section.

are not results that the reader can assume to be the case, they are not stated independently, but rather are tied grammatically to the actions that bring them about. Two examples of this from the corpus are:

(52a) If you do not wish to return the handset to the base, position the OFF/STBY/TALK SWITCH to Stby. *This will disconnect the line and enable your telephone to ring when an incoming call is received.* (Excursion, 1989)

(52b) When the handset is away from the base and not in use, the OFF/STBY/TALK SWITCH should never be lift in the Talk position. Instead, it should be set to Stby, *which will allow the handset to ring.* (Excursion, 1989)

In these examples, the specified results are contingent upon "non-normal" actions, and are grammatically tied to the expression of those actions using "which" or "this" to refer to the action. This, evidently, makes it less likely that the reader will incorrectly expect them to be the case when the action has not been performed. The use of hypotaxis or separate clauses, determined by **sentence-length**, is based on the structural complexity of the clauses; in the corpus, the separate clause form was used if the action proposition was already combined with more than one other proposition. For example, the action in (52a) is coupled with a precondition and would thus be too long to be combined with the result. Example (52b) shows an acceptable combined form.

Go-On conditions are not handled in the result network shown in figure 6.9, but rather, in the rhetorical status selection network shown in the previous chapter in Figure 5.2. The system **Procedural-Node-Type** determines if a Go-On condition node is appropriate, and inserts when necessary. The form of the newly inserted node is marked with present tense and relational form.

6.4 Concurrent Expressions

IMAGENE includes facilities for the expression of concurrent relationships. There are surprisingly few examples of concurrency in the corpus. Instructions for complex processes tend to assume that the reader is familiar with deep levels of concurrency involved and thus need not express them. Consumer electronic devices are, no doubt, explicitly designed to avoid extensive use of concurrent actions. There are, none the less, a number of cases of concurrency in the corpus, from which the current study derived the generalizations formalized in the systems shown in figure 6.11. Representative examples of the lexico-grammatical forms used for concurrency are given here:

(53a) *While cutting*, keep a 1/8-in.-thick piece of plywood under the puzzle to minimize the burr on the underside. (Williams, 1991)

(53b) Press down until the cement sets, *holding the cut edges together as closely as possible.* (Reader's Digest, 1981)

(53c) *When you press the Hour Set Button and Minute Set Button simultaneously* while pressing the Alarm Display/Cancel Button, the Clock Display returns to "12:00". (Panasonic, 1987)

(53d) Drive a Nail: *Hold it in place with your thumb and forefinger* and tap gently with hammer a few times until it is firmly set, then *remove finger* and hit it straight in. (Livers, 1992)

Example (53) shows the four basic forms of concurrent expressions in the corpus. Notice that these forms came from non-telephone manuals. Concurrency is one of the issues that necessitated the inclusion of other forms of instructions in the corpus. The standard fronted participial form linked with "while" is shown in example (53a). This form is used in several functional contexts with respect to concurrency. Another participial form is found in example (53b). Example (53c) expresses concurrency using the adverb "simultaneously". Finally, the durative expression of concurrency is shown in example (53d).

IMAGENE uses these forms in various functional contexts to effectively express concurrency. The initial functional distinction is made in the **Type** system, shown in Figure 6.11. This system distinguishes between simultaneous concurrency and skewed concurrency. These temporal relations are called *simultaneous* and *overlapping* relations by Allen (1983).

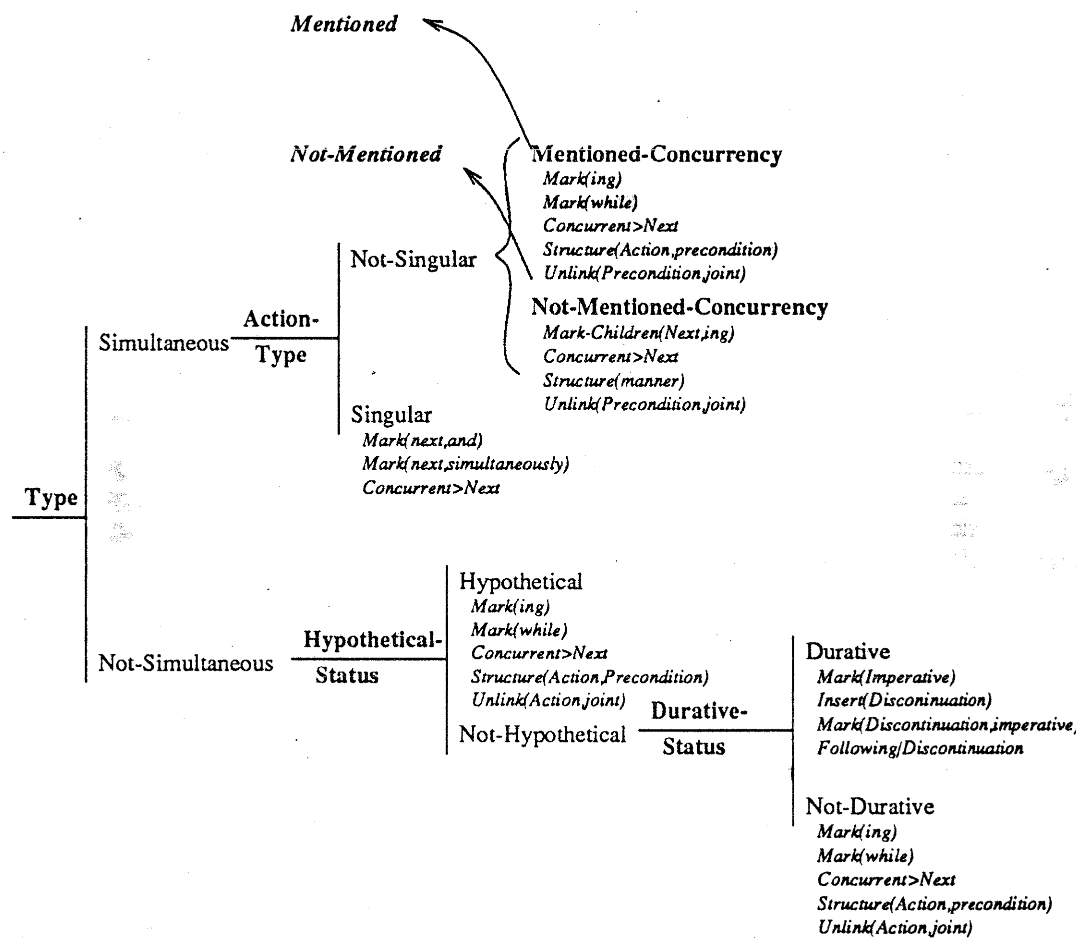

Figure 6.11: The Concurrency System Network

Action-Type divides simultaneous concurrency into the concurrent application of the same action to different objects, called *singular* concurrency, and those concurrent sets that involve mixed actions. Singular concurrency is expressed as in example (53c) using the adverb "simultaneously".

Non-singular concurrency is expressed either with the "while" participial clause, as in example (53a) or the adjunct participial clause shown in example (53b). These two forms of concurrency express the primary action as a telic action, an action with an endpoint, and the secondary action as an atelic action. The duration of the secondary action is implied by the endpoint of the primary action. This mechanism is described by Karlin (1988) and will be discussed in the next section.

When the non-singular concurrent action has been mentioned before, the conjoined clause form in example (53a) is used, otherwise the adjunct participial form in example (53b) is used. This latter form has been studied by Webber and Di Eugenio (1990) in the context of their study of free adjuncts in instructional text. Their study also considered what they called *Augmentation* adjuncts which specified elements of the actions such as instruments and manner. These elements are not considered in the current study because they do not deal with relations between actions, but rather with the nature of the actions themselves.

Non-simultaneous actions are characterized by concurrent processes where the performance of one of the actions is started before the others, usually subsuming them. This first action will be called the *primary action.* **Hypothetical-Status** divides this type of concurrent actions into those actions which are not part of the required sequence of actions in the process and those which are. The former, called hypothetical actions, are expressed with the fronted "while" participial clause as well, as in:

(54) Note: *While pressing the Alarm Display/Cancel Button,* if you press the Sleep Button, the seconds will also appear. (Panasonic, 1987)

Here, the concurrent process rooted in the primary action of pressing the Alarm Display/Cancel button Sleep button is provided as an optional note to the reader.

Durative-Status divides the non-hypothetical actions into those whose primary action which can be expressed with a durative verb and those which cannot. Durative verbs are those whose semantics imply that the action should be performed until further notice, as is the case with "hold it in place" in example (53d). They are categorized by Moens and Steedman as extended events (1988) in that they are extended in time rather than atomic, and as having atelic aspect by Hopper and Thompson (1980). The action of holding the nail in place is expected to be continued until the explicit discontinuation command "remove finger" is given. **Durative-Status**, therefore, uses realization statements to insert a new node in the rhetorical structure for the discontinuation command needed for durative commands.

6.5 Sequential Action Expressions

IMAGENE expresses all actions that are not placed in the rhetorical categories discussed as imperatives. The imperative form selection sub-network shown in Figure 6.12 performs the lexico-grammatical selection for these expressions. In general, it specifies a simple imperative, action form for the expression, but this section will discuss some special cases.

Add-Imperative determines if the action has already been lexico-grammatically marked by some realization statement earlier in the grammatical form selection process. If so, the Skip-Imperative choice is made and no grammatical markings are specified. If not, the expression is marked as an imperative. In either case, the **Type** system is entered, which identifies a number of special cases of imperative action expressions. Examples of these cases are given here:

(55a) Drive a Nail: Hold it in place with your thumb and forefinger and tap gently with hammer a few times *until it is firmly set,* then remove finger and hit it straight in. (Livers, 1992)

(55b) Fire it until it boils. (Gillberg, 1992)

(55c) 2. Lower door handle over card. NOTE: *Door will remain locked until handset is replaced.* (Airfone, 1991)

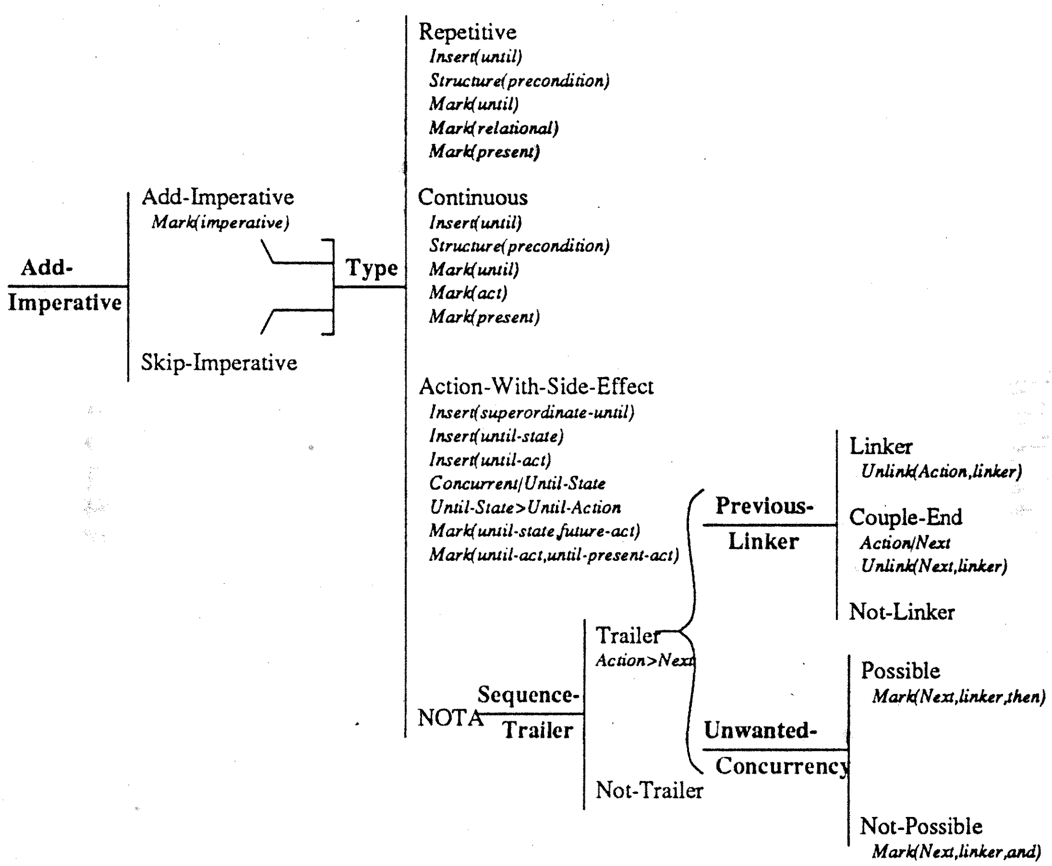

Figure 6.12: The Imperatives System Network

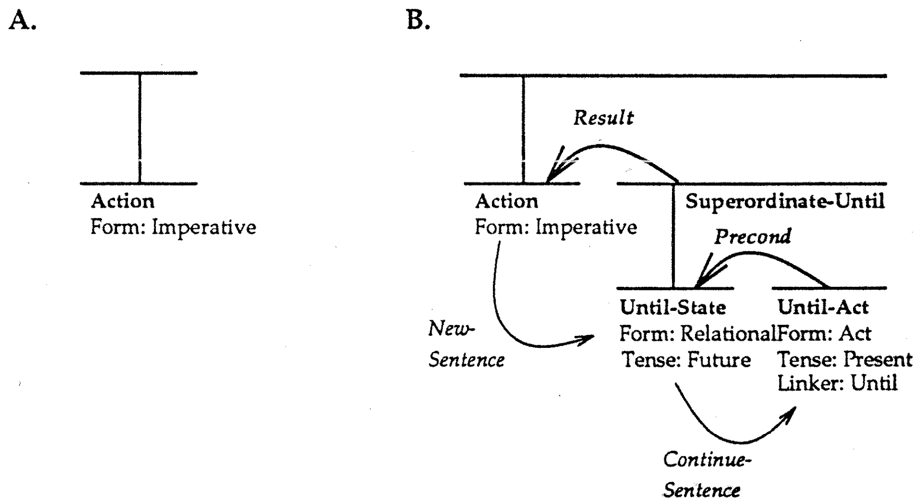

Figure 6.13: The New Nodes Inserted for Actions with Side-Effects

Some actions are either repetitive or continuous, making them impossible to perform correctly unless some explicit condition for stopping is given in the text. In both cases, **Type** inserts an "until" node into the rhetorical structure to do this. For repetitive actions, it marks the "until" node as a relational, as in example (55a). Here the instructions indicate that the nail should be hit with hammer a few times, "until it is firmly set". Continuous actions are treated similarly, except that their "until" nodes are marked as present tense action clauses, as in example (55b). Rösner and Stede 1992a) found it necessary to posit an entirely new relation, which they called the **until** relation, to deal with this case. In the current study it was marked, in the descriptive analysis as a result relation, and constructed in the TRL separately as a unique relation type.

This discussion is based largely on Karlin's analysis of "until" expressions in recipes (1988). She observes that any action which is what Moens and Steedman call a *process* or a *culminated process* can occur with a modifier of duration. These duration markers may take any of the following the forms:

1. Explicit Temporal expressions as in "Do X for 10 minutes."
2. Gradable Terms as in "Do X briefly."
3. Co-extension as in "Do X while doing Y." where the duration of action X is stated in terms of the duration of action Y, which is presumably known.
4. State Change expressions — These may take either of two forms:
   (a) Implicit as in "Chop the onion.", where the size of the onion specifies the duration of the process.
   (b) Explicit as in "Do X until Y."

The current study addresses both the explicit state change expression (the "until" clause) and the co-extension form (discussed in the context of concurrency in the previous section). The other forms are not based on separate action expressions and are thus not considered in the current study.

Occasionally, actions may have side effects that last for specific lengths of time. Example (55c) shows such an example. Lowering the door handle will cause the door to be locked "until handset is replaced." In these cases, **Type** inserts three new rhetorical nodes as shown in Figure 6.13: (1) **Superordinate-Until,** (2) **Until-State**, and (3) **Until-Act**. The realization statements, summarized in figure 6.12, then structure, order, and mark these new nodes as shown in Figure 6.13B.

Actions which are not repetitive or continuous, and do not have expressed side effects are left as simple imperatives. This form constitutes that majority of the action expressions represented in the corpus.

Clause combining for the special action types are handled individually in the realization statements for these types. For actions which are NOTA (None Of The Above), the **Sequence-Trailer** system and its descendants are entered. The initial distinction is made by the **Trailer** system, which determines if there is a trailing sequential action for the node at all. If there is, then two determinations are made, combining and linking. The clauses are combined, in the absence of other factors, in two action couples. This is performed by the **Previous-Linker** system.

The appropriate linker is chosen by **Unwanted-Concurrency**; it may be either "and" or "then" depending upon the possibility of unwanted concurrency of the two actions. If the nature of the actions is such that there may be concurrency and it would be unwanted, "then" is used, otherwise "and" is used. If the two actions are expressed in isolation, IMAGENE would produce one of the following two forms:

(56a) Grasp the top of the handset, and pull out.

(56b) Grasp the top of the handset, then pull out.

In the first, the form of expression does not explicitly rule out the possibility of concurrent execution of the two actions, whereas in the second it does. IMAGENE specifies "and" unless there is the possibility of concurrent execution of the actions which would result in an unwanted side-effect. In the instructions for a garbage disposal (Waste King, 1979), for example, the instruction "Wait about 15 seconds, then turn off water." is given. Here it is definitely possible to wait and turn off the water at the same time, and clearly the instruction writer wants to prevent this interpretation.

6.6 Summary

This chapter has detailed the system networks constructed for use in IMAGENE. The distinctions made in these networks were motivated as much as possible with references to the relevant work in Linguistics, Psychology, and Computer Science, and also with intuitive arguments concerning their acceptability. Ultimately, however, the justification for the system network comes with an empirical validation of the predictions it makes compared with the instructional text corpus. This justification is the subject of the next chapter.

Chapter 7

Results

So far, this thesis has focussed attention on the general philosophy of the approach to generating effective instructions, namely that a corpus of real texts should serve as a model for text generation and as a rule for the assessment of the generated text. Chapter 3 discussed the nature of the corpus that has been used, and how the linguistic analysis was performed. Chapter 4 discussed in some detail the implementation of IMAGENE and Chapters 5 and 6 discussed how the results of the analysis were encoded. This study has even, in some places, discussed how the predictions of the IMAGENE model actually sound better and, further, match or modify current research in Psycholinguistics and Computational Linguistics. Although these intuitive motivations and references to the literature have been helpful in generating new functional distinctions to use in the model and may give the model's predictions more credence, clearly, the point of the whole study is to build a model whose predictions match the real text using some quantifiable testing criterion. This chapter defines such a criterion and details how it is used.

Another deeper goal of this chapter is to use the testing criterion to assess the quality of the IMAGENE model as a representation of the characteristics of the instructional register. IMAGENE makes distinct, testable predictions concerning the rhetorical and grammatical features of effective instructional text. These predictions were seen, in Chapter 2, as important contributions of the current study to related investigations of instructional text.

This chapter defines the criterion for success used in the current study and details how well IMAGENE's predictions match the characteristics of a wide range of instruction types. IMAGENE is shown to produce rhetorical expressions that match the expressions in the corpus exactly in 53.9% of the cases. It is also shown to produce acceptable instructions in another 39.95% of the cases. It concludes with a discussion of the nature of the instructional register and the diversity of the text therein.

7.1 The Criterion for Success

Quite basically, a complete test of the predictions of IMAGENE would involve running the system networks with Process Structures built to match the procedural structures of all the text in the current corpus. This allows the assessment of how closely the rhetorical and grammatical predictions made by IMAGENE match those used in the actual text. In its strongest sense, this criterion is too difficult to be practical. It requires that IMAGENE be capable of producing the appropriate SPL statement for every statement in the corpus, or at least the IMAGENE sanctioned version of the statement, and further, that Penman must have the appropriate lexical items to actually do the surface generation. Implementing the lexical items for the current corpus, which contains 1365 different lexical items, is far beyond the scope of the IMAGENE project and would not add much to the analysis. The focus of the current study in on lexical and grammatical choice at the rhetorical level, not on surface realization.

A weaker, but still satisfying criterion was used, one which specifically tests the phenomena that IMAGENE was designed to address.. IMAGENE was required to produce a TRL structure that matches the relevant grammatical characteristics of the original expression in the corpus. Relevant grammatical characteristics are defined to be those grammatical aspects that are specifically addressed by the analysis

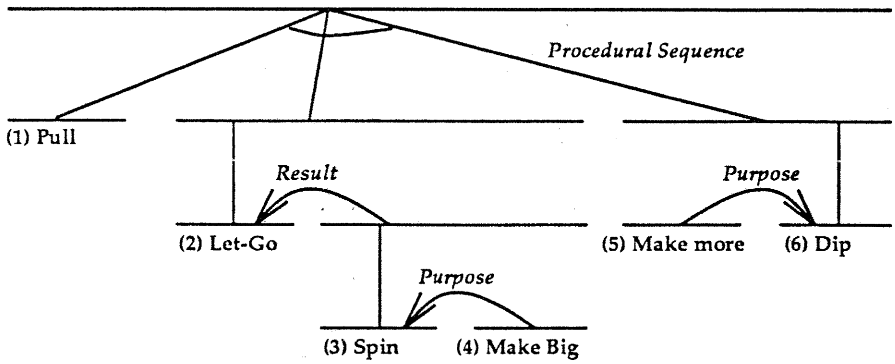

Figure 7.1: The RST analysis of the Bubble-Plane Example

and represented in the TRL, namely the specified linker, the grammatical form (tense, expression type), textual order, and clause combining. No coded PRL structure is required for this test. It requires that IMAGENE produce the matching text structure, specifying all the relevant lexical and grammatical features, but does not require that Penman actually generate the text, thus the lexical and perhaps grammatical forms need not be added to Penman for surface-level realization. Bypassing the actual sentence realization step also speeds the testing process up considerably.

A special set of routines was developed to record important information concerning the execution of IMAGENE. These routines allowed the IMAGENE system network to run and produce a TRL structure as usual, stopping short of calling Penman for the surface realization. In addition, they collected two kinds of important statistics: (1) a list of which textual features were chosen for each action in the example, and (2) a record of the match of IMAGENE's predictions for the linker, form, slot, and combining of the actions. The actions were identified by the code that matched the primary key for the clause or phrase in the corpus database, including the source type and the clause or phrase index. This made it easy to collect statistics at the end of the test runs concerning the accuracy of IMAGENE's predictions for the various relation types and the various source types.

This approach to analysis is distinct from the normal method of assessing the quality of a generation system's output, which is based on an introspective analysis of readability. Kukich performed such an analysis on the output of Ana, assessing what she called the *fluency* of its text (1983). Her analysis was helpful in identifying obvious problems with the output, but failed to give any quantitative measure of the match of Ana's output with a corpus of stock market reports, the domain of her study. Mellish and Evans (1988) did a similar analysis of their generator's output.

To make this discussion of the success criterion and of the testing routines more concrete, this section will detail the analysis of the following example passage, taken from the instructions for simple child's toy (Bubble plane, 1991):

Bubble-Plane Text:

> 2. PULL red knob attached to string, then let go. The propeller will spin to make big bubbles. To make more bubbles, dip wand into bubble solution.

This instructional text expresses six actions, the RST analysis of which is shown in Figure 7.1.

1. pulling the red knob ("PULL red knob attached to string");
2. letting it go ("then let go");
3. the propeller's spinning ("The propeller will spin");
4. making big bubbles ("to make big bubbles");

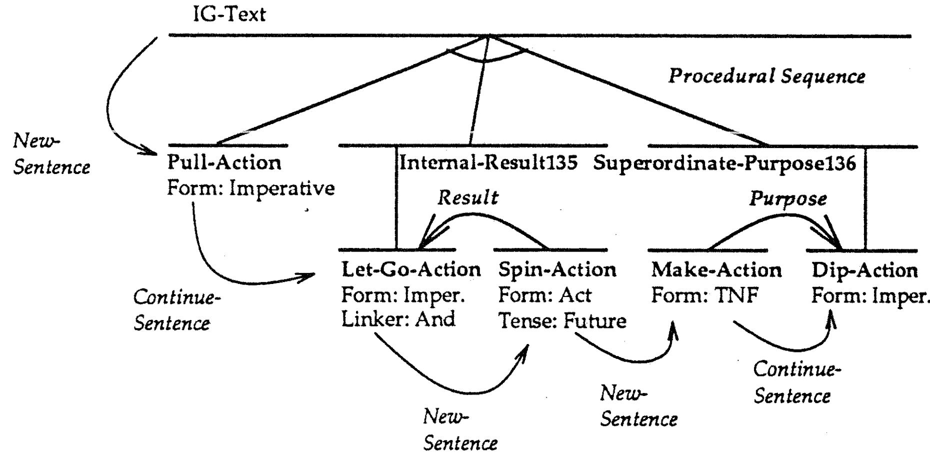

Figure 7.2: A Sample TRL structure

5. making more bubbles ("To make more bubbles");

6. dipping the wand ("dip wand into bubble solution").

The first step in testing an example such as this is to verify that IMAGENE is capable of handling all of the actions identified in the text. In this example, all the actions are modeled, except for action 4 ("to make big bubbles"). This action is a purpose being attributed to the device itself, in this case the Bubble Plane. Notice that its expression is shown in the RST diagram in Figure 7.1 as text span (4), but is not shown in the TRL structure produced by IMAGENE, shown in Figure 7.2. IMAGENE does not currently model non-reader purposes because they occurred so infrequently in the corpus, thus the example is run without reference to this action. The testing routines will collect all the data concerning the other actions, and when IMAGENE is finished, the information concerning this action is added manually.

Also, it must be noted that the tests were run on small spans of text, on the order of 5 to 10 clauses or propositional phrases, such as the Bubble-Plane example. This was done for mostly for convenience, as running large spans would have been prohibitively difficult. With small spans of text, mistakes made in answering the inquiries required re-running only the current text. Making a single mistake on a large span of text would require not only redoing the local span of text, but all the previous spans as well. More importantly, the current study has addressed local rhetorical relations, making the structuring issues inherent in large spans of text beyond the scope of the analysis.

The TRL output by IMAGENE for the example is shown graphically in figure 7.2 and in raw form in figure 7.3. This structure specifies three leaf-level reader actions ("pulling", "letting go", and "dipping"), a result ("spinning"), and a purpose ("making"). IMAGENE added two internal nodes as well, **superordinate-purpose136** and **internal-result135**, to make the rhetorical structure consistent with the current study's use of RST hierarchies. This section will now discuss each of the primary lexical and grammatical features that were tested, the linker, form, slot, and clause combining, particularly with respect to the Bubble Plan example.

The testing routines record the match of IMAGENE's predictions based on the judgments of the tester. At the end of each example run, the routines display the resulting TRL commands for each of the expressible nodes in the text structure and determine the match of each one in turn. For the Bubble Plane example, the match of the various expressions is shown in figure 7.4. Note that the non-reader purpose expression is counted incorrect for all of the lexical and grammatical features.

The linkers are specified correctly for most of the clauses. The linker for the second clause, however, is incorrectly marked as "and". IMAGENE's model of conjunctions for sequential imperatives allowed the use of "then" only when there was the possibility of explicit unwanted concurrency, which is not the case here. Thus, the expression of the Let-Go-Action is marked as having an incorrect linker, the

```
(tell (:about pull-action trl:span
       (trl:continue-sentence let-go-action)
       (trl:form 'trl::imperative)))

(tell (:about let-go-action trl:span
       (trl:result spin-action)
       (trl:form 'trl::imperative)
       (trl:linker 'and)
       (trl:new-sentence spin-action)))

(tell (:about spin-action trl:span
       (trl:form 'trl::act)
       (trl:tense 'trl::future)
       (trl:new-sentence make-action)))

(tell (:about internal-result135 trl:span
       (trl:procedural-subspan let-go-action)))

(tell (:about make-action trl:span
       (trl:form 'trl::tnf)
       (trl:continue-sentence dip-action)))

(tell (:about ig-text trl:span
       (trl:new-sentence pull-action)
       (trl:procedural-subspan pull-action)
       (trl:procedural-subspan internal-result135)
       (trl:procedural-subspan superordinate-purpose136)))

(tell (:about dip-action trl:span
       (trl:form 'trl::imperative)
       (trl:purpose make-action)))

(tell (:about superordinate-purpose136 trl:span
       (trl:procedural-subspan dip-action)))
```

Figure 7.3: TRL Commands for the Sample Structure

Action Code	Linker	Form	Slot	Clause Combining
Pull-Action	ok	ok	ok	ok
Let-Go-Action	bad	ok	ok	ok
Spin-Action	ok	ok	ok	bad
Non-Reader Purpose	bad	bad	bad	bad
Make-Action	ok	ok	ok	ok
Dip-Action	ok	ok	ok	ok

Figure 7.4: The Level of Match for the Running Example

form, slot, and clause combining would be marked as correct as will be seen shortly.

IMAGENE marked the grammatical form for the sequential reader actions as imperative, as future tense action for the result, and as "to" infinitive for the purpose. These all matched the forms in the corpus. Notice that match of form included what are called tense and form in the TRL. A correct marking form indicates that both match the corpus.

IMAGENE also specified the correct textual order. The sequential reader actions are in temporal order as they should be, the result is not fronted (results never are fronted by IMAGENE) and the purpose is fronted (because it is considered optional).

The clauses are combined into sentences in the correct way except for the Spin-Action expression. As discussed above, the test run was executed without reference to the non-reader purpose of making bubbles (clause 4). Because of this, the Spin-Action expression is not combined with the following clause as it is in the corpus.

The non-reader purpose clause attached to the result was not specified at all. In this case, the `Add-Example` command is use to mark the purpose clause in the text would be marked as incorrect for all the grammatical features (linker, form, slot, and combining).

The sample run of the statistics gathering routines on the Bubble-Plane text result in the values for the usage and match statistics shown in Figure 7.5. The `*usage*` variable indicates the IMAGENE features chosen for each of the example actions in the test run. In the Bubble-Plane run, for example, the feature `procedural-sequence` was chosen for 5 of the action expressions: bpl-c10, bpl-c11, bpl-c12, bpl-c14, and bpl-c15, while the feature `non-reader-action` was chosen for only one expression, bpl-c12. This facility is useful for identifying under-utilized systems in the network, that is, those features which are chosen by a small number of examples in the full corpus test. It is also useful for locating examples that pertain to specific functional features that may be of interest.

The `*match*` variable indicates the match for the four types of lexical and grammatical specifications made by IMAGENE. In the Bubble-Plane run, for example, the specification of linker made by IMAGENE matches for 4 examples: bpl-c14, bpl-c10, bpl-c12, and bpl-c15, but doesn't match for 2 examples: bpl-c13 and bpl-c11. These statistics, as will be seen shortly, are useful for computing the accuracy of IMAGENE's predictions for the full corpus, or for portions of the corpus.

Given the value of `*match*`, it is interesting to determine the accuracy of IMAGENE's predictions for various aspects of the corpus. For example, one might want to know the general match of the predictions for the Bubble-Plane example. These are computed by `Compute-Stats` as shown in figure 7.6. The routine takes a parameter indicating the particular actions that are of interest, in this case clauses 10 through 15 of the Bubble-Plane text (bpl) which are represented as `bpl-c10` through `bpl-c15`. The results show 66.67% accuracy for linker and combining and 83.33% accuracy for form and slot. The combined accuracy is listed as 50%. This indicates that one-half of the action expressions in the Bubble-Plane example were reproduced correctly (Pull-Action, Make-Action, and Dip-Action), and the other half had one or more problems with them (Let-Go-Action, Spin-Action, and the problem non-reader purpose).

In conclusion, this section will discuss some of the features of the corpus text that were not considered in the testing. For example, the rhetorical markings were not tested directly. Rather, the grammatical forms that they gave rise to were tested. The analysis of the corpus indicated that the range of possible forms for each of the rhetorical relations were disjoint, making testing of just the grammatical form sufficient. Further, from the point of view of text generation, the underlying relations matter only insofar as they help to determine the appropriate lexical and grammatical form.

Notice also that the exact form of the referring expression, anywhere from the full NP to a zero NP, is not tested. IMAGENE currently implements a simple algorithm that uses the full NP specified in the PRL (using an explicit article) on the first occurrence and a pronoun for all the rest. Thus, in the first sentence, IMAGENE would have produced "Pull the red knob, then let it go." IMAGENE also produces definite articles for objects related to the device and indefinite article for all others. It would, for example, insert "the" in the first clause and would use "it" in the second. Neither would IMAGENE have included the post-modifier "attached to string". Although some facility in the PRL could have been rigged up to specify it, this was not done as such modifiers are beyond the scope of the current study.

Secondly, the lexical choice for the verbs was not really tested. Although the TRL specified the

```
? *usage*
((root-node root)
 (not-concurrent-procedure root bpl-c14)
 (procedural-sequence bpl-c10 bpl-c11 bpl-c12 bpl-c14 bpl-c15)
 (known-procedure bpl-c10 bpl-c11 bpl-c12 bpl-c15)
 (not-concurrent-node bpl-c10 bpl-c11 bpl-c12 bpl-c15)
 (reader-action bpl-c10 bpl-c11 bpl-c14 bpl-c15)
 (not-temporally-remote-prior bpl-c10 bpl-c11 bpl-c12 bpl-c14
                              bpl-c15)
 (not-procedurally-obvious bpl-c10 bpl-c11 bpl-c12 bpl-c14
                           bpl-c15)
 (not-repeated-mention bpl-c10 bpl-c11 bpl-c12 bpl-c14 bpl-c15)
 (not-conditional-action bpl-c10 bpl-c11 bpl-c12 bpl-c14 bpl-c15)
 (add-imperative bpl-c10 bpl-c11 bpl-c15)
 (not-repetitive-continuous-side-effect-action bpl-c10 bpl-c11
                                               bpl-c12 bpl-c15)
 (sentence-sequence-trailer bpl-c10)
 (not-sentence-unwanted-concurrency-possible bpl-c10)
 (not-sentence-previous-linker bpl-c10)
 (no-go-on-condition-node bpl-c10 bpl-c11 bpl-c12 bpl-c15)
 (not-sentence-sequence-trailer bpl-c11 bpl-c12 bpl-c15)
 (nonreader-action bpl-c12)
 (not-action-monitor-result bpl-c12) (add-result bpl-c12)
 (result-complex-context bpl-c12)
 (result-normal-action bpl-c12) (unknown-procedure bpl-c14)
 (purpose-complex-context bpl-c14)
 (purpose-local bpl-c14) (purpose-optional bpl-c14)
 (not-purpose-conditions bpl-c14)
 (not-purpose-volitional-non-reader bpl-c14)
 (not-purpose-goal-metonymy bpl-c14)
 (not-purpose-nominal-available bpl-c14)
 (purpose-tnf-arguments-0k bpl-c14))
? *match*
((linker (bpl-c14 bpl-c10 bpl-c12 bpl-c15)
         (bpl-c13 bpl-c11))
 (form (bpl-c14 bpl-c11 bpl-c10 bpl-c12 bpl-c15)
       (bpl-c13))
 (slot (bpl-c14 bpl-c11 bpl-c10 bpl-c12 bpl-c15)
       (bpl-c13))
 (combining (bpl-c14 bpl-c11 bpl-c10 bpl-c15)
            (bpl-c13 bpl-c12)))
```

Figure 7.5: The Values for *usage* and *match*

```
? (compute-stats '(bpl-c10 bpl-c11 bpl-c12 bpl-c13 bpl-c14
                   bpl-c15))

Choice Match Statistics:

Linker: 66.67%  total occurrences: 6

Form: 83.33%  total occurrences: 6

Slot: 83.33%  total occurrences: 6

Combining: 66.67%  total occurrences: 6

Combined:  50.0%
(6 200/3 250/3 250/3 200/3)
?
```

Figure 7.6: Sample Run of the Statistics Collection Routines

appropriate verb, this is a simple result of the fact that the correct item is given to IMAGENE explicitly in the PRL. The PRL also contains the appropriate argument structure for the verb, as well as some of the non-rhetorical obliques. This simplification is warranted as these sorts of choice are beyond the scope of the current project. The critical issue for the current study is the form of the verb that IMAGENE specifies.

These simplifications have allowed the current study to focus on the construction of appropriate rhetorical structures and their grammatical realization. The task of actually dealing with the lexical choice issues would require a similar linguistic study of the feature in question.

7.2 The Results

Using the statistics gathered in an exhaustive test of IMAGENE's predictions for the expression of each action in the corpus, as described in the previous section, an analysis of the match of IMAGENE's predictions for various categorizations of the actions can be reported. This section will discuss the match of the predictions for the four major rhetorical relations represented in the analysis: Purpose, Precondition, Result, Sequence. The relative distributions of these relation types are shown in figure 7.7.

The accuracy of IMAGENE's predictions for the major relations dealt with in the current study are now discussed in detail. An overview of the results can be found in figure 7.8. This figure contains a cluster of four values for each of the major rhetorical relation types. The cluster indicates the percentage of the actions in the corpus for which IMAGENE's predictions are correct with respect to linker, form, slot, and clause combining.

IMAGENE's purpose network, as described in Chapter 6, correctly predicts 61.61% of the 119 action expressions marked as purposes in the corpus. The levels of match for each of the four aspects of lexical and grammatical forms that concern the current study are shown in figure 7.8.[1] Purposes have the best overall prediction rate in the current study. As was mentioned in Chapter 5, they arise in a single rhetorical context, leading to the better results. This "well-behavedness" has lead to successful studies in a number of contexts, as discussed in Chapter 2.

IMAGENE's precondition network correctly predicts 49.94% of the 98 action expressions marked as preconditions in the corpus. Figure 7.8 shows the breakdown for these expressions as well. The

[1]The reason the 61.61% number is lower than any of the percentages in the table for linker, form, slot, and combining is that an action expression is only considered correct if all four of these features is correct. Only 61.61% of the examples met this criterion.

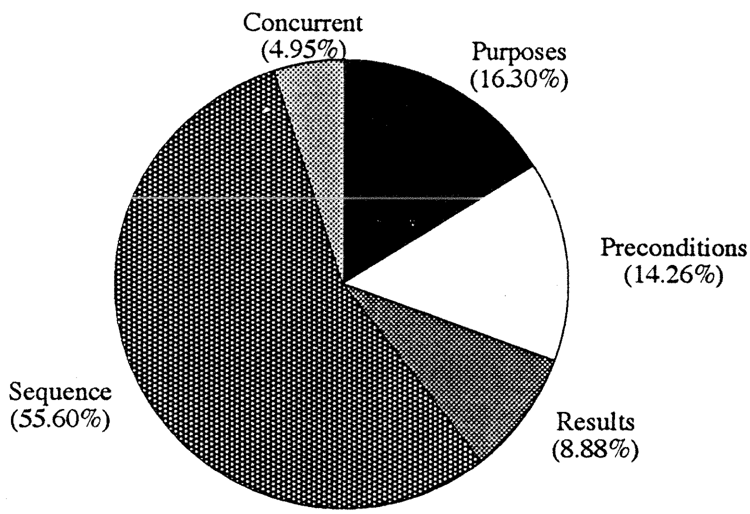

Figure 7.7: The Relative Distributions of Rhetorical Relations in the Corpus

accuracy of IMAGENE's predictions for linker and form of preconditions are lower than its predictions for purposes, on the order of 10% lower. This reflects the difficulty of dealing with the diversity of types of action expressions that eventually become preconditions. As discussed in Chapter 5, while purposes arise in a single rhetorical context, preconditions arise in several. The current precondition sub-network in IMAGENE, discussed in Chapter 6, attempts as much as possible to treat the linker and form selection processes for these various sorts of preconditions separately. IMAGENE's predictions for the slot and form of preconditions is better than for purposes. This reflects one consistent aspect of preconditions, that they are typically expressed before the action that they pertain to, and are combined in the same sentence with its expression.

IMAGENE's results network correctly predicts 49.18% of the 59 action expressions marked as results in the corpus. Figure 7.8 shows that IMAGENE's predictions of linker and form for results are similar to those for preconditions.

Finally, IMAGENE's sequence network correctly predicts 56.54% of the 388 action expressions that are marked as sequence in the corpus. This number is slightly lower than the number for purpose, in spite of the fact that the values displayed in figure 7.8 appear to be much higher. Evidently the issues of linker, form, slot, and combining are more independent in the generation of sequential imperatives than for the other rhetorical relations.

A note must be made here concerning the accuracy of the system networks in general. It is not clear whether the system network produced for IMAGENE is "optimal" in any way. The intent was to produce baseline system network that made effective use of the features determined to be important in the corpus analysis. The problem of modifying the network or even implementing the results in a different paradigm, such as a localist connectionist network for example, is beyond the scope of the current study. The focus here has been on identifying the relevant features for instructional text generation, and on producing a baseline implemented model of them.

7.3 The Instructional Register

The cumulative accuracy rate of IMAGENE's predictions of the rhetorical form of text in the instructional register is 53.9%.[2] In the context of a single corpus of text on which a system network could be trained and tested, this result could clearly be improved by adding a number of systems that addressed single examples in the corpus. This would not give a very accurate picture of the characteristics of the

[2]This value is lower than any of those given for the four grammatical features in the previous section because it indicates the percentage of time that IMAGENE's prediction matched for all of them at once.

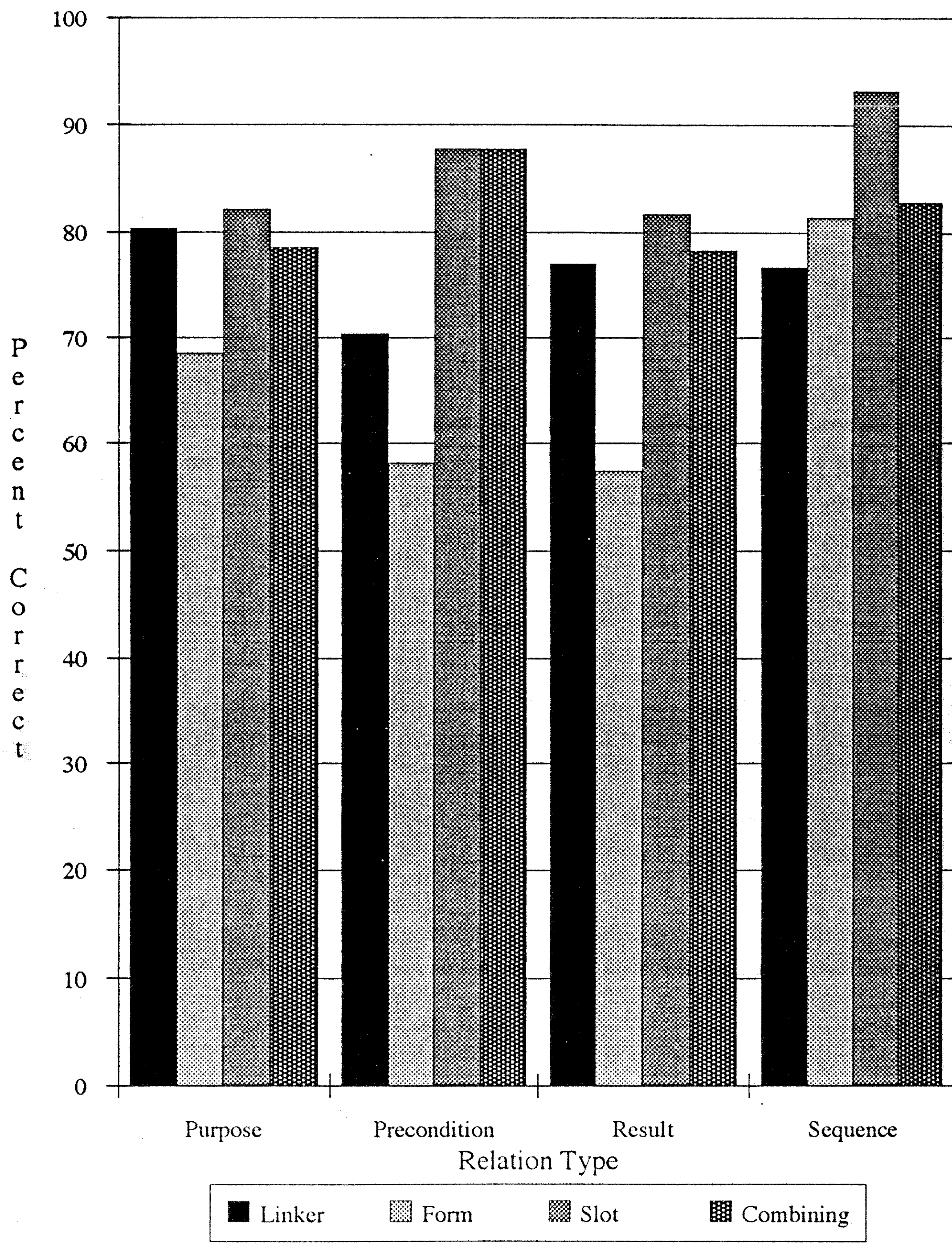

Figure 7.8: Coverage of the Four Major Relation Types

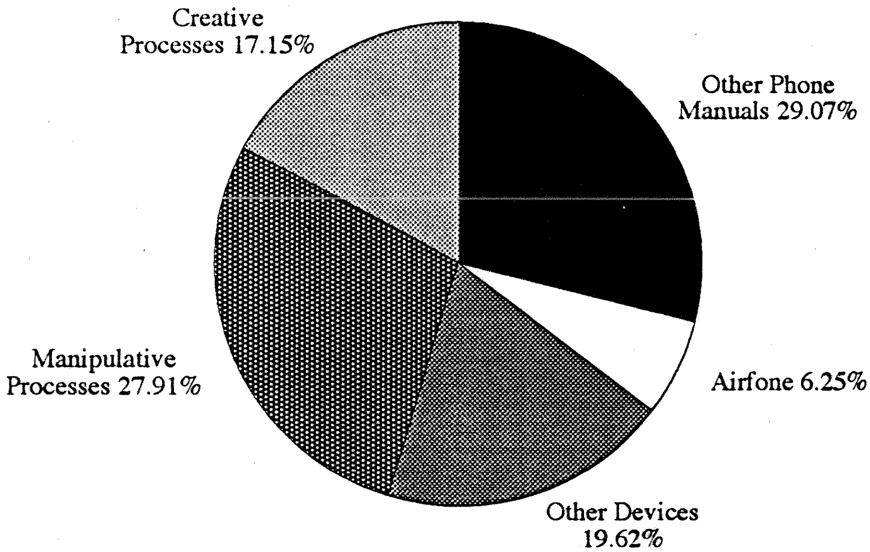

Figure 7.9: The Relative Distribution of Expressions from the Various Source Texts

instructional register as a whole, neither was this approach taken in the current study. As has been indicated, the current study has employed both a training corpus and a testing corpus. The predictions made by IMAGENE are the result of the analysis of the telephone manuals. The general form of the architecture and of the system networks has remained largely the same since they were first written and tested on the telephone corpus (approximately 300 clauses, 2000 words) and reported in earlier publications (Vander Linden et al., 1992b; Vander Linden et al., 1992a). The network was then applied to the full corpus, which included instructions for vastly different types of devices and processes.

The intent of this approach was to develop a detailed analysis of a relatively "tight" corpus, and see how far into the range of instructional text those results would apply. In addition to the telephone manuals, then, a number of other types of instructions were loaded into the corpus. The distribution of these instruction types is illustrated in Figure 7.9. No attempt was made to create a balanced corpus, i.e., one which represents all the types of instructions in their proper distributions. The various types of instructional text included are discussed in some detail in Chapter 3.

There were some changes made to the networks after the telephone manuals were analyzed, designed to accommodate constructions that were either under-represented in the telephone corpus or absent altogether. Changes were made to accommodate "until" clauses, "so that" clauses, and concurrency. These grammatical forms were not common in the original telephone texts, although some of them did occur. The changes made to the networks were largely done with simple extensions, rather than by modifying anything that was already there. Thus, with a couple of minor exceptions, the test results reported in figure 7.10 show IMAGENE's accuracy when trained or developed on the telephone manuals and then applied to the "Other Devices", the "Manipulative Processes", and the "Creative Processes". [3]

In this figure, the Airfone telephone manual and the other two telephone manuals were represented separately. "Other devices" denotes other types of consumer electronic devices including manuals for a television and a clock-radio. Manipulative processes are those instructions, aside from the consumer electronic device manuals, that manipulate a particular object, as opposed to creating a new object. The manipulative process instructions include auto repair manuals, the Bubble Plane manual, and first aid. The creative process instructions include recipes and craft manuals.[4]

Note also that the "Other Phone Manuals" values have been split between procedural and primitive

[3]Given this approach to the analysis, the learning procedure employed has much in common with the method proposed by Quinlan and implemented in ID3 (Quinlan, 1986). In the ID3 approach, the training set of the corpus is expanded in cases where there are insufficient examples on which to base a full analysis, just as was done here in the case of "until" and "so that" clauses and expressions of concurrency.

[4]For a detailed list of the various source texts see the discussion in Chapter 3 or the chart included in Appendix A.

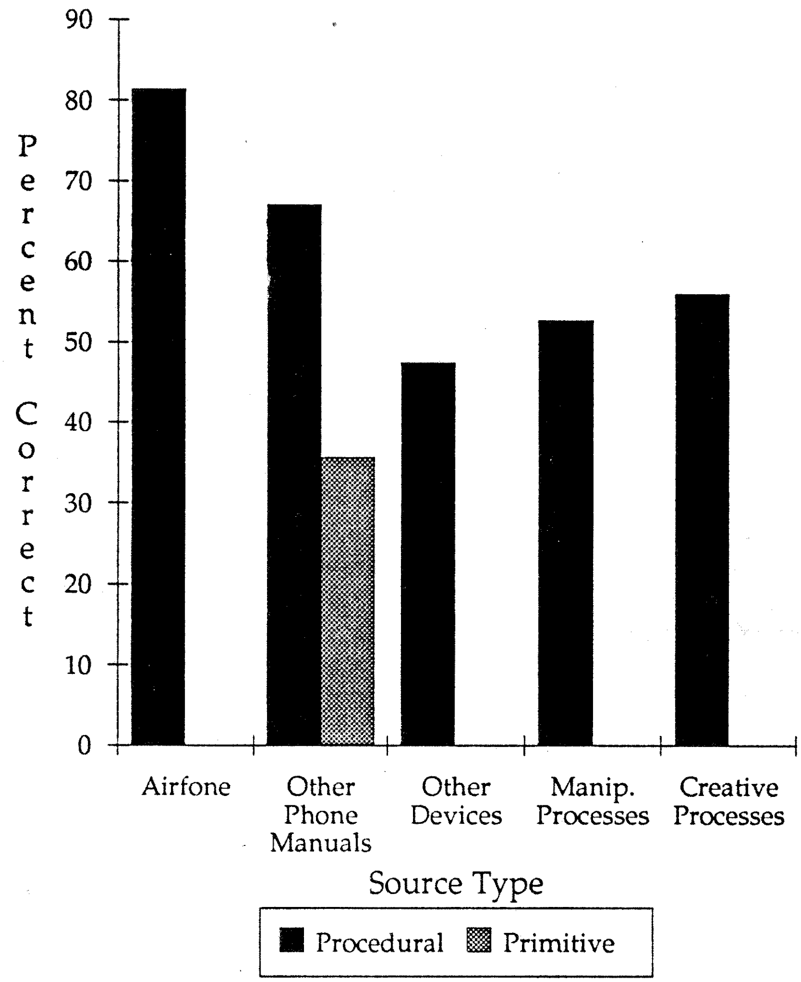

Figure 7.10: The Relative Accuracy of IMAGENE Predictions for the Various Source Texts

type passages, and that IMAGENE's accuracy on the procedural text (66.97%) is considerably higher than its accuracy for the primitive text (35.56%). The analysis conducted for the current study has focused on the use of rhetorical tools to express procedural relationships, and thus is most accurate for procedural text. Here is an example of the sort of primitive text organization that IMAGENE does not model very well:

```
OFF/STBY/TALK Switch [8]

OFF - The 7010 will not ring.  OFF is used primarily for
charging the batteries.  If the telephone will not be used
for a long time, set the switch to OFF and place the handset
on the base.

STBY - In the STBY (standby) position, the telephone will
ring whether the handset is on the base or in another
location.  Switch to TALK to answer or make a call.

TALK - The phone will ring only if the handset is on the
base.  With TALK, the 7010 works like an ordinary phone no
need to move the switch to answer or make calls.  Remember,
in TALK position, with the handset away from the base, your
caller will hear a busy signal and your 7010 will not ring.
```

Clearly, this passage includes some procedural text which IMAGENE does handle, such as the last sentence in the description of the "OFF" setting ("If the telephone will not be used for a long time, set the switch to OFF and place the handset on the base"). It also, however, includes a number of grammatical forms that are motivated by the needs of primitive organization, such as the first sentence in the description of the "OFF" setting ("OFF — The 7010 will not ring.")

Because of this focus on procedural relationships, heavily primitive styles of instructions were deliberately excluded from the corpus for the other source types, leading to results shown in figure 7.10. They were included from the original telephone manuals because, at the time of the pilot study (Vander Linden et al., 1992b), the decision to focus on procedural relationships had not yet been made.

The approach taken, therefore, of starting with an analysis of a relatively tight corpus of telephone manual text, and expanding to a more general corpus, permits the current study to comment on the characteristics of procedural texts, in the instructional register as a whole. It was unclear at the start of the work how far the results could be effectively applied. The typical writers and readers for, say, recipes and auto-repair manuals, are quite different. Nonetheless, both types of text have fit into IMAGENE's rhetorical analysis quite nicely, suggesting that there are some functional characteristics of instructions that are common to the expression of both types of processes. Clearly, recipes have distinct characteristics, but the things that set them apart are their special requirements concerning referring expressions, which have been dealt with by other researchers (e.g., Dale, 1992).

The range of instructional types represented in the corpus did not have as much an effect of the accuracy of IMAGENE' predictions as did the range of writing styles found in the text. The small excerpt from an instruction manual for using the Macintosh interface proved the most difficult for IMAGENE, registering only 17.39% accuracy. This excerpt was included because it contained a number of issues identified as difficult for IMAGENE including overlapping rhetorical relations (Ford, 1986) and a fronted "by" purpose expression.

The various source texts did display some level of uniformity with respect to the distribution of the rhetorical relation types. This information can be seen in Figure 7.11. Most of the texts had, at least, representatives from all the rhetorical relation types, with the exception of the recipes (labeled "r1" and "r2" in the diagram). "R2" in particular, was composed almost exclusively of sequential actions, and "r1" was similar.

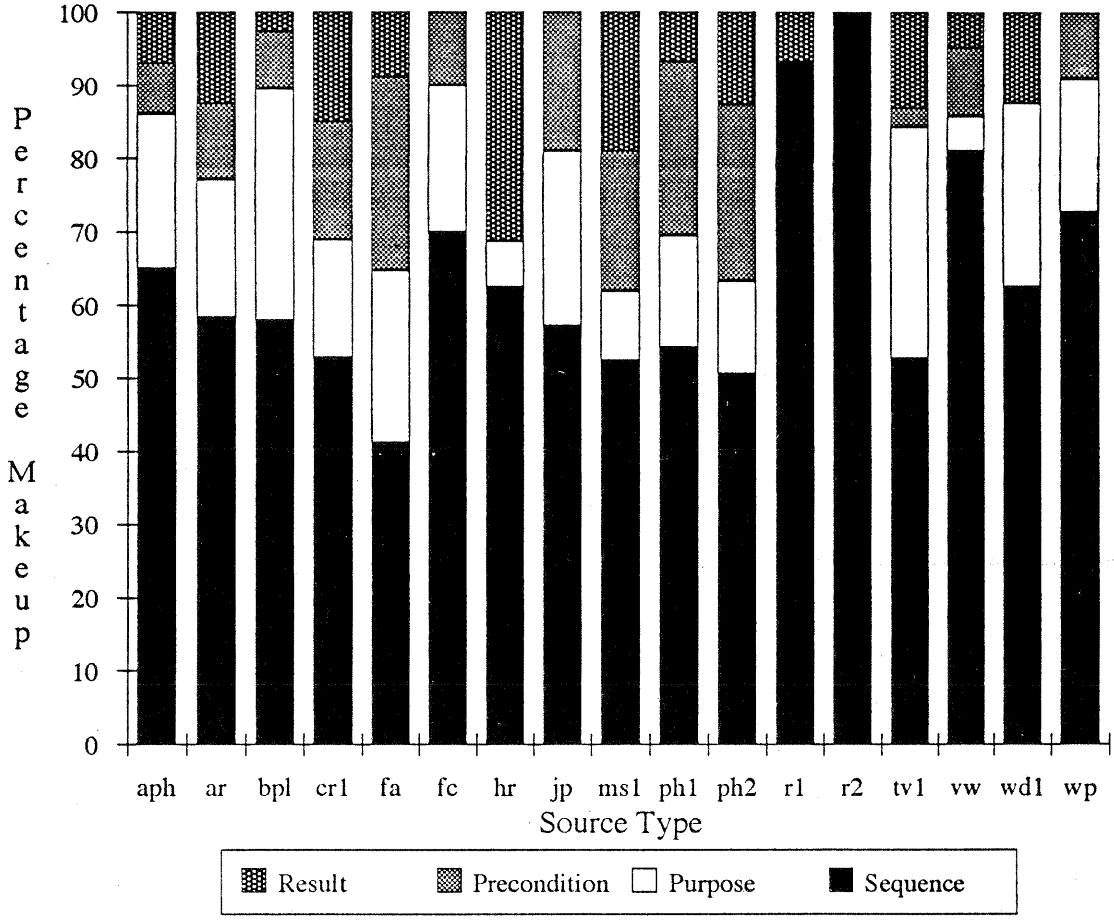

Figure 7.11: The Relative Distributions of the Relation Types in the Various Source Texts

7.4 Summary

This chapter has presented the meat of the current study, the analysis of the accuracy of IMAGENE's rhetorical and grammatical predictions with respect to the instructional corpus. It detailed the approach to match verification taken and the computational tools used to implement it. It concluded with a detailed discussion of the accuracy of IMAGENE's predictions for the fundamental rhetorical relations in the corpus study and also for the various types of texts included in the instructional corpus.

One note should be made here about the statistics themselves. The cumulative match rate of 53.9% may seem low, particularly for a corpus of this size. There are, however, two issues which render the coverage more understandable.

First, the predictions of the system are never ungrammatical. They may not "exactly" match the text in corpus, but they do come close. For example, IMAGENE specified an "and" rather than a "then" in the Bubble-Plane example discussed earlier. This was counted as an incorrect linker in spite of the fact that the linker chosen by IMAGENE, by any introspective account, is quite understandable. This can be characterized, therefore, as a "soft" failure. "Hard" failures, those action types that IMAGENE simply does not handle, occur as well, but are not as common. The non-reader purpose expression in the Bubble Plane text ("to make big bubbles") is an example, as non-reader purposes are simply not modeled by IMAGENE. In the corpus test, 53.9% of the action expressions are predicted correctly by IMAGENE, 39.95% are soft failures, and 6.15% are hard failures.[5]

Secondly, IMAGENE's predictions may even be better in some cases than the text in the corpus. Although the general philosophy of the approach taken in the current study is to assume that the choices made by the writers of the corpus are correct, there are cases where the forms in the corpus are probably inappropriate. Such a determination would be the subject of a psycholinguistic study, and is thus beyond the scope of the current study. The Furnace text, which will be discussed in the next chapter, is probably such a case.

[5] In these statistics, hard failures are defined as those examples for which IMAGENE made no correct predictions, that is, its predictions for linker, form, slot, and combining predictions are all marked as incorrect. In the cases where IMAGENE didn't model a particular type of action, all four predictions were marked in this way.

Chapter 8

Conclusions

This thesis has argued that the issue of detailed forms of expression and how they are systematically used within a specific language genre should be a fundamental issue in text generation. Rather than choosing, by hand, a number of expressional forms that are sufficient to disambiguate a short list of functions of the domain, determined ahead of time to be important, a study in text generation should attempt to deal with the whole range of ideational, textual, and interpersonal issues that affect rhetorical and grammatical choice in real texts. It has further argued that the best method for addressing this issue is the use of a detailed corpus-based analysis of the language domain in question. Rather than viewing the language produced by trained writers within a genre as suspect, it was viewed as a source for the fundamental rhetorical and grammatical tools that are most effective at communicating the type of information common to the genre. In a sense, then, the current study has conflated that which "is" with that which "ought to be". The consistent patterns of use which surface in a broad range of instructional texts are taken to be principles of effective instruction writing. The specific claims made in the context of these assumptions are testable using Psycholinguistic methodologies, such as those of Dixon (1987a) discussed in Chapter 2.

This concluding section will start with a review of IMAGENE's solution to the running examples that were introduced in the first chapter, and in particular, a detailed discussion of the solution to the Furnace text. It will then review the fundamental contributions of the current study and discuss the implications of the study for future text generation research.

8.1 The Running Examples Revisited

The introductory chapter included a number of texts that served, at least in part, to illustrate and define the issues that IMAGENE has addressed. They also served as running examples throughout the thesis as various aspects of the analysis and implementation were discussed. The Installed-Phone text served as an example IMAGENE's operation in the introductory chapter, Chapter 1, and as an example of the analysis in the corpus analysis chapter, Chapter 3. The Remove-Phone text served as an example for the discussion of the details of IMAGENE's implementation in Chapter 4. A complete trace its run through IMAGENE is included in Appendix C. As these examples have been dealt with in some detail elsewhere, they will not be discussed again here.

The unique aspects of the Furnace text are the subject of the current section. Its analysis provides particular insight into what the current study's approach has to offer to the field of text generation and to the study of improving instructional text in general (Vander Linden, 1993). The text is repeated here for reference:

Furnace Text:

> Depress knob and hold for 60 seconds after pilot has been lighted. Release knob and turn to ON position.

The question raised in Chapter 1 concerned the procedural status of the action of actually lighting

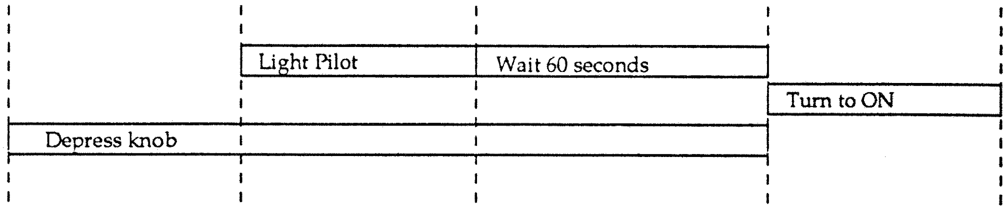

Figure 8.1: A Graphic Representation of the Furnace Process

the pilot. As it is not stated as an imperative, it is unclear, from this text, if it must be performed at all. The pilot could, for example, be an auto-lighting pilot. Upon careful study of the entire text, it can be determined that this is not the case with this pilot. Thus, there are clearly problems with the form of expression in this text[1], but before we get too hard on the technical writer, consider the complexity of the process being expressed, depicted graphically in figure 8.1. There are four basic actions (depressing, lighting, waiting and turning), each of which would presumably be expressed as imperatives were it not for the concurrency involved.

The Process Structure constructed for this process is depicted graphically in 8.2 and textually in Figures 8.3 and 8.4. The concurrency involved in the furnace process requires that this process structure contain more than the normal sequential hierarchic decomposition. It must also represent the concurrency. The PRL supports this by allowing a concurrent child type as well as a sequential child type. The relationship between **Depress-Action** and **Internal-Action2** is of the concurrent type. All the others are of sequential type. Dealing with concurrency has also necessitated the use of non-expressible, internal nodes in the representation. In the furnace process structure, there are two internal nodes. **Internal-Node1** allows the inclusion of a complex, concurrent process in a sequence of actions, here as the first action in a sequential pair of actions. **Internal-Action2**, as discussed above, allows the representation of the concurrent execution of a single action, in this case **Depress-Action** and a sequence of actions, in this case **Light-Action** and **Wait-Action**. IMAGENE builds a similar structure with internal nodes that are not expressible, i.e., they are not in the sentence trace sequence.

The Text Structure built by IMAGENE for this process is depicted graphically in 8.5 and textually in Figure 8.6. The corpus study has shown that the use of the passive, perfective form, as was used in the original furnace text ("after pilot has been lighted"), is not at all common in this context. IMAGENE, when given the relevant aspects of the context, produces the following more commonly used form:

(57) *Depress the knob. Light the pilot, and wait for sixty seconds. Release the knob, and turn it to the ON position.*

An action, intended to be performed concurrently with other actions, is generally expressed, if possible, with a *durative* verb ("depress"), followed by the concurrent actions ("light" and "wait"), and culminated with a *discontinuation* command ("release"). A durative verb is one whose action is assumed to be continued until an explicit discontinuation verb is encountered. The semantics of the verbs themselves are sufficient to imply the concurrency. In this example, then, the principles of expression derived from a corpus-based analysis uncovered what appears to be a more effective expression for this procedure. It is interesting to note that early in the IMAGENE project, the author came up with two alternate expressions for this text which were derived entirely introspectively:[2]

- "While holding down the knob, light the pilot. Continue holding down the knob for another 60 seconds."
- "Depress the knob. While holding the knob down, light the pilot. Release the knob 60 seconds after lighting."

[1] In fact, the Furnace text itself was given to the author by the person who owned the furnace in question and had trouble understanding the directions.

[2] These two alternate forms are taken directly from a slide prepared for the proposal defense for the IMAGENE project.

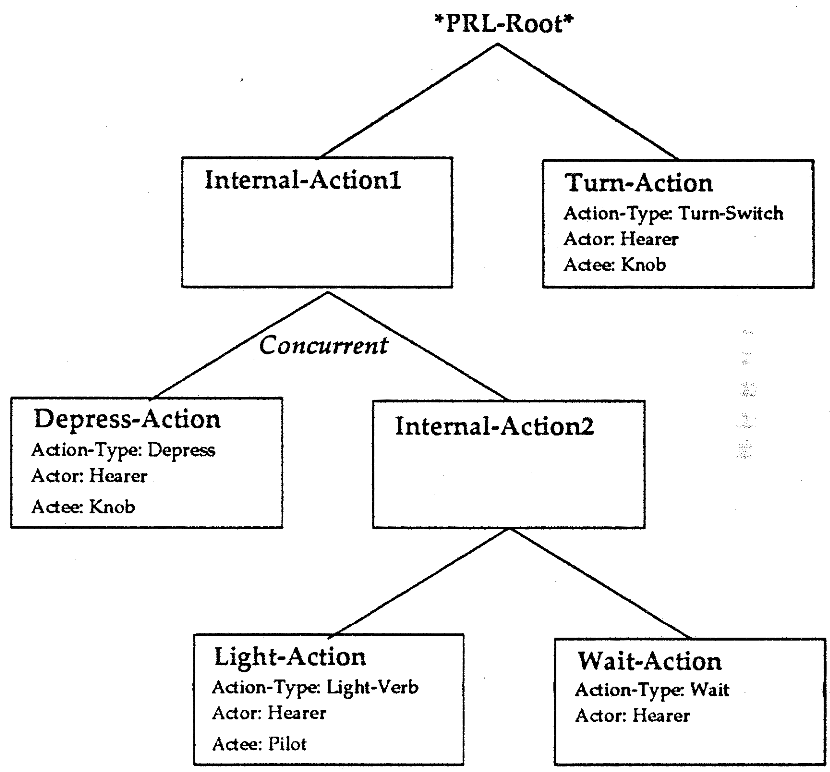

Figure 8.2: A Graphic Representation of the Furnace Process Structure

```
(tell (:about *prl-root* Action
              (subaction internal-action1)
              (subaction turn-action)
              (subaction-type procedural)))

(tell (:about internal-action1 Action
              (subaction depress-action)
              (subaction internal-action2)
              (subaction-type concurrent)))

(tell (:about depress-action action
              (action-type it::depress)
              (actor hearer)
              (actee knob)))

(tell (:about internal-action2 Action
              (subaction light-action)
              (subaction wait-action)))

(tell (:about light-action Action
              (action-type it::light-verb-trans)
              (actor hearer)
              (actee pilot)))

(tell (:about wait-action Action
                (action-type it::wait)
              (actor hearer)
              (duration 60)
              (duration-units it::second-time)))

(tell (:about turn-action action
              (action-type it::turn-switch)
              (actor hearer)
              (actee knob)
              (destination off-position)))

(tell (:about release-action action
              (action-type it::release)
              (actor hearer)
              (actee knob)))
```

Figure 8.3: The PRL Commands for the Actions in the Furnace Process

```
(tell (:about knob object
              (object-type it::knob)
              (object-status device)))

(tell (:about hearer object
              (object-type it::hearer)))

(tell (:about pilot object
              (object-type it::pilot)
              (object-status device)))

(tell (:about off-position object
              (object-type it::on-position)
              (object-status device)))
```

Figure 8.4: The PRL Commands for the Objects in the Furnace Process

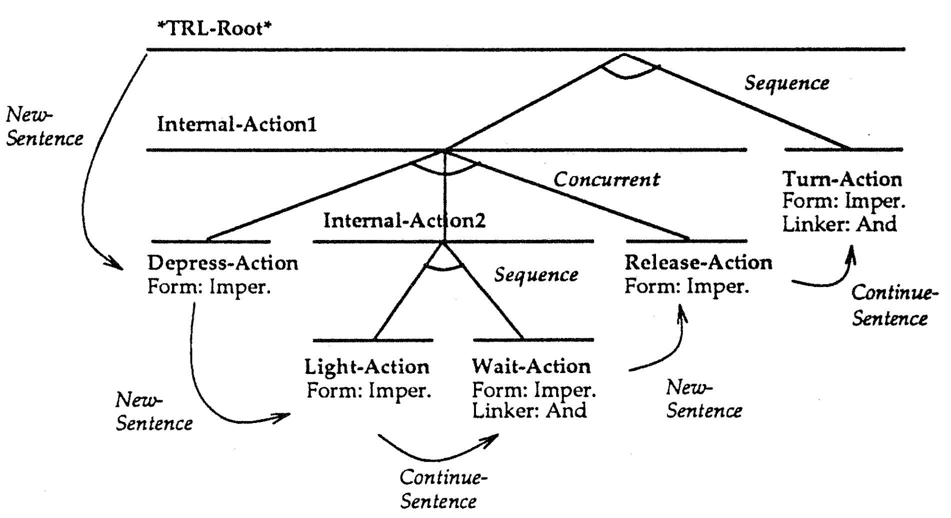

Figure 8.5: A Graphic Representation of the Furnace Text

```
(tell (:about light-action trl:span
       (trl:continue-sentence wait-action)
       (trl:form 'trl::imperative)))

(tell (:about internal-action1 trl:span
       (trl:procedural-subspan release-action)
       (trl:procedural-subspan internal-action2)
       (trl:procedural-subspan depress-action)))

(tell (:about internal-action2 trl:span
       (trl:procedural-subspan wait-action)
       (trl:procedural-subspan light-action)))

(tell (:about wait-action trl:span
       (trl:form 'trl::imperative)
       (trl:linker 'and)
       (trl:new-sentence release-action)))

(tell (:about depress-action trl:span
       (trl:form 'trl::imperative)
       (trl:new-sentence light-action)))

(tell (:about turn-action trl:span
       (trl:form 'trl::imperative)
       (trl:linker 'and)))

(tell (:about ig-text trl:span
       (trl:new-sentence depress-action)
       (trl:procedural-subspan turn-action)
       (trl:procedural-subspan internal-action1)))

(tell (:about release-action trl:span
       (trl:continue-sentence turn-action)
       (trl:form 'trl::imperative)))
```

Figure 8.6: The TRL Commands for the Furnace Text Structure

Although it is unclear whether these forms or IMAGENE's version is the most effective, it is clear that IMAGENE's version is quite different. This shows the diversity of language that can be used in instructions and, more importantly, leads one to suspect our ability to introspectively derive the relevant forms to be tested for relative effectiveness in a Psycholinguistic context. The corpus-based approach advocated in this thesis can be seen as a source of distinct and realistic forms of expression for such purposes.

The nature of IMAGENE's implementation of the results also provides further insight to the writing process. The contextual information input to IMAGENE can be manipulated in an attempt to produce a form that more closely matches the one found in a particular text, such as the furnace text. This process sheds some light on the functional criteria the writer may have been using when they produced the form in the manual. For example, consider the following forms, one from the furnace domain and the other from the telephone domain:

(58a) *Depress the knob. Wait for sixty seconds after lighting the pilot. Release the knob, and turn it to the ON position.*

(58b) *Return the OFF/STBY/TALK switch to the STBY position after your call.*

If IMAGENE is told that the action of lighting the pilot is well-known or obvious to the reader, it produces the form in example (58a). Earlier in the furnace text, the technical writer did, in fact, mention that the pilot must be manually lighted and may have assumed that this would render the action obvious to the reader. Unfortunately, such a mention doesn't make the lighting action obvious. IMAGENE requires both that the action be known to the reader prior to the reading of the manual and that it be the focus of the current paragraph, before it is considered obvious. This is the case with the action of making a call on a cordless telephone expressed in example (58b). Such manuals tend to assume that the reader already knows how to use a normal (non-cordless) phone, thus making the "your call" action obvious in the context of a paragraph on making a call. Notice that the expressional forms in example (58) closely match the form in the original furnace text. The actions that are considered obvious, the "lighting" and the "your call" actions, are expressed out of temporal order, grammatically marked with the preposition "after". This rhetorical demotion, however, is inappropriate in the Furnace text.

IMAGENE also varies the expressional form of an action that has already been mentioned in the text. Consider these examples:

(59a) *Depress the knob. When the pilot is lighted, wait for sixty seconds. Release the knob, and turn it to the ON position.*

(59b) *When the phone is installed, and the battery is charged, move the OFF/STBY/TALK switch to the STBY position. The phone is now ready to use.*

In example (59a), IMAGENE, because it has been told that the action has already been mentioned in the previous text, has rhetorically demoted the action of lighting the pilot to a precondition expression. Although the furnace text did mention the fact that the pilot must be lighted manually, this previous mention did not actually specify this action as an imperative. IMAGENE requires that previous mentions cited in this context be explicit imperative commands. This pattern of repeated mention, although of questionable use in this furnace text, does occur in instruction manuals. A preliminary section will tell the readers how to install a device using imperative commands, perhaps giving detailed sub-steps for installation, and a subsequent section on how to use the device will begin with a rhetorically demoted action in precondition form, such as the ones in example (59b). Here, both of the preconditions expressed ("the phone is installed" and "the battery is charged") have been prescribed to the reader in an earlier section as imperatives, thus warranting the rhetorical demotion. This form of rhetorical demotion is, again, not appropriate for the lighting action in the furnace example.

8.2 Contributions

This thesis is fundamentally based on IMAGENE, a substantive working model of text generation within the domain of instructional text, but its implications lie far beyond this system. This section reviews these implications.

8.2.1 The Corpus-Based Approach

The primary contribution of the current study is the specification of a detailed corpus-based method of analysis of text corpora as applied to text generation. The use of a database representing the rhetorical and grammatical aspects of the corpus, discussed in Chapter 3, is the hallmark of this approach. The methodology, taken largely from linguists working in the Discourse-Functional community, employs a hypothesis proposal and text cycle to aid in determining the functional features of the context of expression that weigh on the precise rhetorical and grammatical choices made by real writers in the text genre.

It is important to note that this approach is applicable to any lexical or grammatical aspect of any linguistic genre. The large corpus of instructional text amassed in the current study will, no doubt, be helpful in the future for analyzing other aspects of instructional text, such as referring expressions, negations, and macro-structure, but the design of the database is general to all text types, and quite flexible[3].

8.2.2 The Analysis of Instructional Text

One important by-product of the analysis conducted for IMAGENE, is the analysis of instructional text itself. The fundamental idea that actions themselves lie behind nearly all of the expressions in instructional text, namely preconditions, purposes, results, and action expressions, is critical here. This observation has implications for the development of planners for use in producing the process structures used in generating instructional texts.

Further, the rhetorical, grammatical, and lexical characteristics of the instructional genre should be helpful in other domains of study, including Reading Research, Linguistics, and Natural Language Processing. These fields would benefit from the specific and testable predictions made by IMAGENE concerning the generation of effective instructions. These benefits, in the context of instructional text, are obvious, but the results of the IMAGENE project may also provide insight to similar issues in other genres as well, very much in the same way that Linguistic research in other genres benefitted the analysis conducted here.

Because the discussion of the features of the communicative context that have been determined to be important in the generation of instructional text was spread out over Chapters 4 through 6, a more concise list of them is presented in Appendix D. This list identifies all of the functional distinctions that are implemented in IMAGENE, and discusses the consequences of each one for the rhetorical, grammatical, and lexical characteristics of the resulting instructional text.

8.2.3 The Feature-Based Implementation of Text Generation

The final major contribution of the current work is the demonstration of the use of a feature-based implementation of text generation. The Penman-like implementation of IMAGENE should serve as a base-line implementation for future work on generation of instructional text and also for generation of other specific sublanguages.

8.3 Future Work

The study of instructional text specifically, and the work on generating texts in general is far from complete. The current study, focusing primarily on the rhetorical aspects of instructions, has given rise to a number of related topics that should be addressed in order to complete the arguments begun here.

The analysis of other linguistic phenomena in instructional text — This thesis has claimed that the approach taken to the study of rhetorical relations in instructional text is applicable to other aspects of the lexical and grammatical form of instructions. One such issue is referring expressions. A number of studies have addressed this issue, but it is far from clear if they ever took a corpus of

[3] Although the database is not appropriate for interactive discourse, other similar databases have been developed for interactive, spoken language (Du Bois and Schuetze-Coburn, 1993; Lampert and Lampert, 1993).

real instructions as seriously as the approach taken in the current study has. Other similar issues include negations and macro-structure.

The study of multi-lingual instructions — The approach to linguistic analysis can also be applied in the context of multiple languages. This could address the extremely practical problem of generating multi-lingual instructional texts, resulting in a system that could produce drafts of the parallel texts in the appropriate language and thus avoiding the difficult and expensive task of machine translation.

The addition of a procedural planner — Many of the claims and assumptions made in the current study assume the use of a procedural planner. Such a planner would be critical in determining the content of the instructional text, both at the macro and the micro levels. It would also be critical in implementing the text-level inquiries, a step that is required for fully automating the instruction generation process.

The addition of a user model — Closely related to the addition of a planner is the addition of a component that would model the knowledge and skills of the user/reader. It is clear from the current study that the level of the reader's knowledge affects the lexical and grammatical form of action expressions. Thus, the implementation of an automated user model for use in modeling the user's knowledge is also required for fully automating the instructional generation process.

The Psycholinguistic verification of the expressional claims — The current study has made a number of testable claims concerning the precise lexical and grammatical nature of effective instructional text. These claims are specific enough to serve as a basis for the test examples in a psycholinguistic study such as those discussed in Chapter 2.

The integration of a design tool for specifying processes in a non-linguistic manner — The ultimate goal of this work is to serve as part of a larger design environment in which processes and device interfaces may be specified in a non-linguistic manner. This design tool, in conjunction with the procedural planner and user model mentioned above, could be used to provide the input structures that have been assumed in the current study.

The use of spreading activation networks rather than system networks — Some of the lexical and grammatical aspects of the IMAGENE's predictions are characterized by sets of mutually interacting constraints, a phenomenon that linear system networks do not address adequately. Clause combining is such a phenomenon; there are mutually conflicting reasons to combine, and to separate clauses, reasons that are hard to linearize in a system network formalism. An alternate computational formalism that is better suited to this phenomenon is the localist network which implements spreading activation with continuous activation values. Such networks have been used in generation (Ward, 1990; Ward, 1991) before, but not with respect to rhetorical issues such as those considered in the current study. One problem with localist networks is the difficulty in assigning the appropriate weights to the connections between nodes, a problem that might be solved using statistics concerning the level of co-occurrence of two text features, collected directly from the instructional text corpus.

As the study of text generation becomes more sophisticated in its attention to the issue of the detailed linguistic aspects of the language it is trying to generate, it will begin to produce more and more effective, readable text. This study is hopefully a step in that direction.

Bibliography

AECMA (1988). *AECMA Simplified English: A Guide for the Preparation of Aircraft Maintenance Documentation in the International Aerospace Maintance Language*. Association Europeenne Des Constructeurs de Materiel Aerospatial. AECMA Document PSC-85-16598, Change number 4.

Airfone (1991). *Inflight Entertainment & Information Guide*. United Airlines.

Allen, J. F. (1983). Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843.

Allen, J. F. (1984). Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154.

Allen, J. F. (1987). *Natural Language Understanding*. Benjamin/Cummings, Menlo Park, CA.

Alterman, R. (1988). Adaptive planning. *Cognitive Science*, 12:393–421.

Alterman, R., Zito-Wolf, R., and Carpenter, T. (1991). Interaction, comprehension, and instruction usage. *The Journal of the Learning Sciences*, 3 & 4:361–398.

Arens, Y. and Hovy, E. (1990a). How to describe what? towards a theory of modality utilization. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society*, July 25–28, Cambridge, MA. Lawrence Erlbaum Associates.

Arens, Y. and Hovy, E. H. (1990b). The problem of modality selection. In *Proceedings of the Annual Meeting of the Cognitive Science Society*. Cambridge, MA.

Bach, E. (1982). Purpose clauses and control. In Jacobson, P. and Pullum, G. K., editors, *The Nature of Syntactic Representation*, pages 35–57. D. Reidel, Dordrecht, Holland.

Badler, N. I., Webber, B. L., Kalita, J. K., and Esakov, J. (1991). Animation from instructions. In Badler, N. I., Barsky, B. A., and Zeltzer, D., editors, *Making Them Move: Mechanics, Control, and Animation of Articulated Figures*, chapter 3, pages 51–93. Morgan Kaufmann, San Mateo, CA.

Balkanski, C. T. (1992). Actions, beliefs and intentions in rationale clauses and means clauses. In *Tenth National Conference on Artificial Intelligence*, July 12–16, San Jose, California.

Bateman, J. A. and Hovy, E. H. (1991). Computers and text generation: Principles and uses. In Butler, C. S., editor, *Computers and Texts: An Applied Perspective*, chapter 2. Basil Blackwell. Forthcoming.

Bateman, J. A. and Paris, C. L. (1989). Phrasing a text in terms the user can understand. In *Proceedings of the 11th International Joint Conference on Artificial Intelligence*. August 20–25, 1989, Detroit, Michigan, also published as ISI technical report ISI/RS-89-240.

Bateman, J. A. and Paris, C. L. (1990). A methodology for investigating register. This paper is from the USC/Information Sciences Institute.

Bates, M. (1978). The theory and practice of augmented transition networks. In Bloc, L., editor, *Natural Language Communication with Computers*. Springer, New York.

Beekman, J. and Callow, J. (1974). *Translating the Word of God*. Zondervan, Grand Rapids, MI.

Beekman, J., Callow, J., and Kopesec, M. (1981). The semantic structure of written communication. Technical report, Summer Institute of Linguistics, Dallas, TX.

Berry, M. (1981a). Systemic linguistics and discourse analysis: A multi-layered approach to exchange structure. In Coulthard, M. and Montgomery, M., editors, *Studies in Discourse Analysis*, chapter 6, pages 120–145. Routledge & Kegan Paul.

Berry, M. (1981b). Towards layers of exchange structure for directive exchanges. *Network*, 2:23–32.

Biber, D. (1988). *Variation Across Speech and Writing*. Cambridge University Press, Cambridge.

Brachman, R. J. (1979). On the epistemological status of semantic networks. In Findler, N. V., editor, *Associative Networks: Representation and Use of Knowledge by Computers*. Academic Press, New York.

Brachman, R. J. and Schmolze, J. G. (1985). A overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9:171–216.

Brill, D. (1992). *LOOM Reference Manual Version 1.4*. USC Information Sciences Institute.

Britton, B. K. (1986). Capturing art to improve text quality. *Educational Psychologist*, 21(4):333–356. Special Issue: Design of Instructional Text.

Britton, B. K. and Gülgöz, S. (1991). Using Kintsch's computational model to improve instructional text: Effects of repairing inference calls on recall and cognitive structures. *Journal of Educational Psychology*, 83(3):329–345.

Britton, B. K., Van Dusen, L., Gülgöz, S., and Glynn, S. M. (1989). Instructional texts rewritten by five expert teams: Revisions and retention improvements. *Journal Of Educational Psychology*, 81(2):226–239.

Britton, B. K., Van Dusen, L., Gülgöz, S., and Glynn, S. M. (1990). The impact of inferences on instructional text. In Graesser, A. C. and Bower, G. H., editors, *The Psychology of Learning and Motivation*, volume 25, pages 53–70. Academic Press, San Diego, CA.

Brown, C. (1987). Towards a theory of nl generation: The connection between syntax, context, and semantics. Technical Report LCCR TR 87-8, Simon Fraser University.

Brown, C., Dahl, V., Hamilton, S., Massam, D., Massicotte, P., and Pattabhiraman, T. (1986a). Tailoring government and binding theory for automatic natural language generation. Technical Report LCCR TR 86-4, Simon Fraser University.

Brown, C., Pattabhiraman, T., Boyer, M., Massam, D., and Dahl, V. (1986b). Tailoring conceptual graphs for use in nl translation. Technical Report LCCR TR 86-14, Simon Fraser University.

Brown, D. E. and Clement, J. (1987). Overcoming misconceptions in mechanics: A comparison of two example-based teaching strategies. Paper presented at the annual meeting of the American Educational Research Association, Washington, D. C.

Bubble plane (1991). *Bubble Plane Instructions*. Playschool, Inc., Pawtucket, RI 02862.

Butler, C. S. (1985). Discourse systems and structures and their place within an overall systemic model. In Benson, J. D. and Greaves, W. S., editors, *Systemic Perspectives on Discourse*, volume 1: Selected Theoretical Papers from the 9th International Systemic Workshop, chapter 11, pages 213–228. Ablex.

Cawsey, A. (1990). Generating explanatory discourse. In Dale, R., Mellish, C., and Zock, M., editors, *Current Research in Natural Language Generation*, chapter 4. Academic Press. Selected readings from the 2nd European NLG Workshop, April, 1989, Edinburgh.

Chafe, W. L., editor (1980). *The Pear Stories: Cognitive, Cultural, and Linguistic Aspects of Narrative Production*. Ablex, Norwood, NJ.

Chafe, W. L. (1984). How people use adverbial clauses. In *Proceedings of the 10th Annual Meeting of the Berkeley Linguistics Society*.

Charrow, V. R., Holland, V. M., Peck, D. G., and Shelton, L. V. (1980). Revising a medicaid recertification form: A case study in the document design process. Technical report, American Institutes for Research, Washington, D. C.

Chomsky, N. (1957). *Syntactic Structures*. Mouton, The Hague.

Chomsky, N. (1965). *Aspects of the Theory of Syntax*. MIT Press, Cambridge, MA.

Chomsky, N. (1981). *Lectures on Government and Binding: The Pisa Lectures*. Foris, Dordrecht, Holland.

Church, K. W. and Mercer, R. L. (1993). Introduction to the special issue on computational linguistics using large corpora. *Computational Linguistics*, 19(1):1–24.

Code-a-phone (1989). *Code-A-Phone Owner's Guide*. Code-A-Phone Corporation, P.O. Box 5678, Portland, OR 97228.

Cohen, P. R. and Levesque, H. J. (1985). Speech acts and rationality. In *Proceedings of the 23rd Annual Meeting of Association for Computational Linguistics*, Chicago, IL, pages 49–59.

Crothers, E. (1979). *Paragraph Structure Inference*. Ablex, Norwood, NJ.

Cumming, S. (1984). The sentence in Chinese. *Studies in Language*, 8(3):365–395.

Cumming, S. (1988). The lexicon in text generation: Progress and prospects. This is a revised version of a paper presented at the 1986 Workshop on Automating the Lexicon.

Cumming, S. (1990). Natural discourse hypothesis engine. In McKeown, K. and an Sergei Nirenburg, J. M., editors, *Proceedings of the Fifth International Workshop on Natural Language Generation*, June 3–6, Dawson, PA.

Cumming, S. (1991). Nominalization in English and the organization of grammars. In *Proceedings of the IJCAI-91 Workshop on Decision Making Throughout the Generation Process*, August 24–25, Darling Harbor, Sydney, Australia.

Dale, R. (1988). *Generating Referring Expressions in a Domain of Objects and Processes*. PhD thesis, Centre for Cognitive Science, University of Edinburgh.

Dale, R. (1990). Generating recipes: An overview of Epicure. In Dale, R., Mellish, C., and Zock, M., editors, *Current Research in Natural Language Generation*, chapter 9. Academic Press. Selected readings from the 2nd European NLG Workshop, April, 1989, Edinburgh.

Dale, R. (1992). *Generating Referring Expressions: Constructing Descriptions in a Domain of Objects and Processes*. MIT Press, Cambridge, MA.

Dale, R., Mellish, C., and Zock, M. (1990). Introduction. In Dale, R., Mellish, C., and Zock, M., editors, *Current Research in Natural Language Generation*, chapter 1. Academic Press. Selected readings from the 2nd European NLG Workshop, April, 1989, Edinburgh.

Danlos, L. (1984). Conceptual and linguistic decisions in generation. In *Proceedings of the 10th International Conference on Computational Linguistics*, July 2–6, Stanford, CA, pages 501–504. Association for Computational Linguistics.

Danlos, L. (1987). *The Linguistic Basis of Text Generation*. Cambridge University Press, Cambridge, England. Dominique Debize and Colin Henderson's English translation of *Génération automatique de texts en langue naturelles*, Ph.D. Dissertation, University of Paris, 1985.

Davey, A. (1978). *Discourse Production: A Computer Model of Some Aspects of a Speaker*. Edinburgh University Press.

de Beaugrande, R. (1980). *Text, discourse and process.* Ablex, Norwood, NJ.

Di Eugenio, B. (1992). Understanding natural language instructions: The case of purpose clauses. In *Proceedings of the Annual Meeting of Association for Computational Linguistics*, Newark, DE, pages 120–127.

Dilley, S., Bateman, J., Thiel, U., and Tissen, A. (1992). Integrating natural language components into graphical discourse. In *Proceedings of the Third Conference on Applied Natural Language Processing*, 31 March - 3 April, Trento, Italy.

DiMarco, C. (1990). *Computational Stylistics for Natural Language Translation.* PhD thesis, University of Toronto. Available as technical report CSRI-239.

Dixon, P. (1982). Plans and written directions for complex tasks. *Journal of Verbal Learning and Verbal Behavior*, 21:70–84.

Dixon, P. (1987a). Actions and procedural directions. In Tomlin, R. S., editor, *Coherence and Grounding in Discourse*, pages 69–89. John Benjamins. Outcome of a symposium, Eugene, OR, June, 1984.

Dixon, P. (1987b). The processing of organizational and component step information in written directions. *Journal of Memory and Language*, 26:24–35.

Dixon, P., Faries, J., and Gabrys, G. (1988). The role of explicit action statements in understanding and using written directions. *Journal of Memory and Language*, 27(6):649–667.

Donin, J., Bracewell, R. J., Frederiksen, C. H., and Dillinger, M. (1992). Students' strategies for writing instructions. *Written Communication*, 9(2):209–236.

Du Bois, J. W. (1980). Beyond definiteness: The trace of identity in discourse. In Chafe, W. L., editor, *The Pear Stories: Cognitive, Cultural, and Linguistic Aspects of Narrative Production*, chapter 5. Ablex.

Du Bois, J. W. and Schuetze-Coburn, S. (1993). Representing hierarchy: Constituent structure for discourse databases. to appear.

Duffy, T. M. (1985). Readability formulas: What's the use? In Duffy, T. M. and Waller, R., editors, *Designing Usable Texts*, pages 113–143. Academic Press, Orlando, FL.

Duffy, T. M., Higgins, L., Mehlenbacher, B., Cochran, C., Wallace, D., Hill, C., Haugen, D., McCaffrey, M., Burnett, R., Sloane, S., and Smith, S. (1989). Models for the design of instructional text. *Reading Research Quarterly*, 24:434–457.

Elhadad, M. (1991). FUF: The universal unifier — user manual, version 5.0. Technical report, Columbia University.

Elhadad, M. (1992). *Using Argumentation to Control Lexical Choice: A Functional Unification-Based Approach.* PhD thesis, Columbia University.

Elhadad, M. and Robin, J. (1992). Controlling content realization with functional unification grammars. In Dale, R., Hovy, E., Röesner, D., and Stock, O., editors, *Aspects of Automated Natural Language Generation*, Lecture Notes in Artificial Intelligence 587, pages 89–104. Springer Verlag, Berlin.

Excursion (1989). *Excursion 3100.* Northwestern Bell Phones, A USWest Company.

Fawcett, R. P. (1990). The COMMUNAL project: Two years old and going well. *Network*, 13/14:35–39.

Fawcett, R. P., van der Mije, A., and van Wissen, C. (1988). Towards a systemic flowchart model for discourse structure. In *New Developments in Systemic Linguistics*, volume 2: Theory and Application, pages 116–143. Pinter.

Feiner, S. K. and McKeown, K. R. (1990). Coordinating text and graphics in explanation generation. In *Proceeding of AAAI-90, Boston, MA.*

Feraci, R. A. (1974). *Aspects of the Grammar of Infinitives and For-Phrases*. PhD thesis, MIT.

Firbas, J. (1966). On defining the theme in functional sentence analysis. *Travaux Linguistiques de Prague*, 1:267–280.

Firth, J. R. (1957). *Papers in Linguistics*. Oxford University Press, London.

Flesch, R. (1948). A new readability yardstick. *Journal of Applied Psychology*, 32:221–233.

Flower, L., Hayes, J. R., and Swarts, H. (1980). Revising functional documents: The scenario principle. Technical report, Carnegie-Mellon University, Pittsburgh, PA. Document Design Project Tech. Rep. no. 10.

Ford, C. (1986). Overlapping relations in text structure. In *Proceedings of the Second Annual Meeting of the Pacific Linguistics Society*.

Ford, C. (1988). *Grammar in ordinary interaction: the pragmatics of adverbial clauses in conversational English*. PhD thesis, UCLA.

Fox (1991). *The Fox Documentation*. Fox Software, Inc., 134 W. South Boundary, Perrysburg, Ohio 43551.

Fox, B. A. (1987). *Discourse Structure and Anaphora: Written and Conversational English*. Cambridge University Press.

Fox, B. A. and Thompson, S. A. (1990). A discourse explanation of the grammar of relative clauses in English conversation. *Language*, 66(2):297–316.

Frederiksen, C. H. (1972). Effects of task-induced cognitive operations on comprehension and memory processes. In Freedle, R. O. and Carroll, J. B., editors, *Language Comprehension and the Acquisition of Knowledge*, pages 211–246. Wiley, New York.

Frederiksen, C. H. (1986). Cognitive models and discourse analysis. In Cooper and Greenbaum, editors, *Written Communication Annual. Vol I: Linguistic Approaches to the Study of Written Discourse*. Sage, Beverly Hills, CA.

Gardner, H. (1987). *The Mind's New Science: A History of the Cognitive Revolution*. Basic Books, New York, the paperback edition edition.

Gillberg, C. (1992). Heroic raku. *Ceramics Monthly*, pages 34–35.

Givón, T. (1979). *On Understanding Grammar*. Academic Press, New York.

Givón, T., editor (1983). *Topic Continuity in Discourse: A Quantitative Cross-Language Study*. TSL #3. John Benjamins, Amsterdam.

Givón, T. (1989). *Mind, Code, and Context: Essays in Pragmatics*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Goldman, A. I. (1970). *A Theory of Human Action*. Prentice Hall, Englewood Cliffs, NJ.

Gopen, G. D. and Swan, J. A. (1990). The science of scientific writing. *American Scientist*, 78:550–558.

Graves, M. F. (1991). Commentary: Improving instructional text: Some lessons learned. *Reading Research Quarterly*, 26(2):110.

Graves, M. F., Slater, W. H., Roen, D. D., Redd-Boyd, T., Duin, A. H., Furniss, D. W., and Hazeltine, P. (1988). Some characteristics of memorable expository writing: Effects of revisions by writers with different backgrounds. *Research in the Teaching of English*, 22:242–265.

Green, S. J. (1992). A basis for a formalization of linguistic style. In *Proceedings of the Annual Meeting of Association for Computational Linguistics*, (Student Session), Newark, DE, pages 312–314.

Green, S. J. and DiMarco, C. (1993). Stylistic decision-making in natural language generation. In *submitted to ACL-93*.

Gregory, M. (1967). Aspects of varieties differentiation. *Journal Of Linguistics*, 3(2):177–198.

Grimes, J. E. (1975). *The Thread of Discourse*. Mouton, The Hague.

Grishman, R. (1986). *Computational Linguistics: An Introduction*. Cambridge University Press, Cambridge.

Grishman, R. and Kittredge, R., editors (1986). *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*. Lawrence Erlbaum Associates, Hillsdale, NJ.

Grosz, B. J. and Sidner, C. L. (1986). Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3).

Halliday, M. A. K. (1961). Categories of the theory of grammar. *Word*, 17:241–292.

Halliday, M. A. K. (1976). *System and Function in Language*. Oxford University Press, London. edited by G. R. Kress.

Halliday, M. A. K. (1978). *Language as a Social Semiotic: The Social Interpretation of Language and Meaning*. University Park Press, Baltimore.

Halliday, M. A. K. (1984). Language as code and language as behaviour: A systemic-functional interpretation of the nature of ontogenesis of dialogue. In Fawcett, R. P., Halliday, M., Lamb, S. M., and Makkai, A., editors, *The Semiotics of Culture and Language Vol. 1: Language as Social Semiotic*, pages 3–35. Frances Pinter.

Halliday, M. A. K. (1985). *An Introduction to Functional Grammar*. Edward Arnold, London.

Halliday, M. A. K. and Hasan, R. (1985). *Language, context, and text: Aspects of language in a social-semiotic perspective*. Deakin University Press.

Harris, Z. (1968). *Mathematical Structures of Language*. Interscience Publishers, John Wiley & Sons, New York.

Hartley, J. (1985). *Designing Instructional Text*. Kogan Page, London, 2nd edition.

Hartley, J. (1986). Planning the typographical structure of instructional text. *Educational Psychologist*, 21(4):245–251. Special Issue: Design of Instructional Text.

Hartley, J., Trueman, M., and Burnhill, P. (1980). Some observations on producing and measuring readable writing. *Programmed Learning and Instructional Technology*, 17:164–174.

Hasan, R. (1978). Text in the systemic-functional model. In Dressler, W. U., editor, *Current Trends in Textlinguistics*, pages 228–246. Walter de Gruyter.

Hjelmslev, L. (1943). *Prolegomena to a Theory of Language*. University of Wisconsin Press, Madison, WI.

Hobbs, J. R. (1978). Resolving pronoun references. *Lingua*, 44.

Hobbs, J. R. (1979). Coherence and coreference. *Cognitive Science*, 3:67–90.

Hobbs, J. R. (1982). Towards an understanding of coherence in discourse. In Lehnert, W. G. and Ringle, M. H., editors, *Strategies for Natural Language Processing*, chapter 8. Lawrence Erlbaum Associates.

Hobbs, J. R. (1985). On the coherence and structure of discourse. Technical Report CSLI-85-37, Center for the Study of Language and Information, Stanford University.

Holland, M. and Rose, A. (1981). A comparison of prose and algorithms for presenting complex instructions. Technical Report 17, Document Design Project, Washington, D. C. American Institutes for Research.

Holtz, H. (1988). *The Complete Guide to Writing Readable User Manuals.* Dow Jones – Irwin.

Hopper, P. J. and Thompson, S. A. (1980). Transitivity in grammar and discourse. *Language*, 56(2):251–299.

Hovy, E., Lavid, J., Maier, E., Mittal, V., and Paris, C. (1992). Employing knowledge resources in a new text planner architecture. In Dale, R., Hovy, E., Röesner, D., and Stock, O., editors, *Aspects of Automated Natural Language Generation*, Lecture Notes in Artificial Intelligence 587, pages 57–72. Springer Verlag, Berlin.

Hovy, E. H. (1986). Putting effect into text. In *Proceedings of the Eigth Annual Conference of the Cognitive Science Society*, Amhert, MA, pages 669–675. Poster Session.

Hovy, E. H. (1987). What makes language formal? In *Proceedings of the Ninth Annual Conference of the Cognitive Science Society*, Seattle, WA, pages 959–964. Poster Session.

Hovy, E. H. (1988a). *Generating Natural Language Under Pragmatic Constraints.* Lawrence Erlbaum Associates, Hillsdale, NJ.

Hovy, E. H. (1988b). Planning coherent multisentential text. In *Proceedings of the Twenty-Sixth Annual Meeting of the Association for Computational Linguistics.* State University of New York, Buffalo, NY, USA, June 7-10.

Hovy, E. H. (1989). Approaches to the planning of coherent text. Technical Report ISI/RR-89-245, USC Information Sciences Institute. From the 4th International Workshop on Text Generation, Catalina Island, CA, July, 1988.

Hovy, E. H. (1990). Pragmatics and natural language generation. *Artificial Intelligence*, 43(1).

Hovy, E. H. and McCoy, K. F. (1989). Focusing your RST: A step toward generating coherent multi-sentential text. In *Program of the Eleventh Annual Conference of the Cognitive Science Society*, August 16–17, Ann Arbor, MI, pages 667–674. Lawrence Erlbaum Associates.

Hudson, R. A. (1971). *English Complex Sentences – An introduction to systemic grammar.* North–Holland, Amsterdam.

Huettner, A. K. (1989). *Adjunct Infinitives in English.* PhD thesis, University of Massachusetts, Swarthmore College.

Huettner, A. K., Vaughan, M. M., and McDonald, D. D. (1987). Constraints on the generation of adjunct clauses. In *Proceedings of the 25th Annual Meeting of Association for Computational Linguistics*, Stanford, CA, pages 207–214.

Jackendoff, R. (1990). *Semantic Structures.* The MIT Press, Cambridge.

Jacobs, P. S. (1985). *A Knowledge-Based Approach to Language Production.* PhD thesis, University of California, Berkeley.

Jacobs, P. S. (1987). Knowledge intensive natural language generation. *Artificial Intelligence*, 33(3):325–378.

Jones, C. (1991). *Purpose Clauses: Syntax, Thematics, and Semantics of English Purpose Constructions*, volume 47 of *Studies in Linguistics and Philosophy.* Kluwer Academic Publishers, Dordrecht.

Jou, J. and Harris, R. J. (1990). Event order versus syntactic structure in recall of adverbial complex sentences. *Journal of Psycholinguistic Research*, 19(1):21–42.

Kaczmarek, T. S., Bates, R., and Robins, B. (1986). Recent developments in NIKL. In *Fifth National Conference on Artificial Intelligence*, Philadelphia, PA, pages 978–985.

Karlin, R. F. (1988). Defining the semantics of verbal modifiers in the domain of cooking tasks. In *Proceedings of the 26th Annual Meeting of Association for Computational Linguistics*, June 7–10, Buffalo, NY, pages 61–67.

Kintsch, W. (1974). *The Representation of Meaning in Memory*. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ.

Kintsch, W. (1988). The role of knowledge in discourse comprehension: A construction-integration model. *Psychological Review*, 95(2):163–182.

Kintsch, W. and van Dijk, T. A. (1978). Toward a model of text comprehension and production. *Psychological Review*, 85:363–394.

Kintsch, W. and Vipond, D. (1979). Reading comprehension and readability in educational practice and psychological theory. In Nilsson, L. G., editor, *Memory: Process and Problems*, pages 329–365. Lawrence Erlbaum Associates, Inc., Hillsdale, NJ.

Kittredge, R., Korelsky, T., and Rambow, O. (1991). On the need for domain communication knowledge. *Computational Intelligence*, 7(4):305–314.

Kittredge, R. I. (1982). Variation and homogeneity of sublanguages. In Kittredge, R. and Lehrberger, J., editors, *Sublanguage: Studies of Language in Restricted Semantic Domains*, Foundations of Communication, chapter 4, pages 107–137. de Gruyter, Berlin.

Kittredge, R. I. (1983). Semantic processing of texts in restricted subdomains. *Comp. & Math. with Appls.*, 9(1):45–58.

Kittredge, R. I. and Polguère, A. (1991). Generating extended bilingual texts from application knowledge bases. In *International Workshop on Fundamental Research for the Future Generation of Natural Language Processing, Kyoto, Japan*, pages 147–160.

Kittredge, R. I., Polguère, A., and Goldberg, E. (1986). Synthesizing weather forecasts from formatted data. In *Proceedings of the 11th International Conference on Computational Linguistics*, Bonn, pages 563–565.

Klare, G. R. (1963). *The Measurement of Readability*. Iowa State University Press, Ames, IA.

Klare, G. R. (1984). Readability. In Pearson, P. D., editor, *Handbook of Reading Research*, chapter 22, pages 681–744. Longman, New York.

Knott, A. and Dale, R. (1992). Using linguistic phenomena to motivate a set of rhetorical relations. Technical Report HCRC/RP-39, Human Communication Research Centre, University of Edinburgh.

Knox, G. (1978). *Better Homes and Gardens Complete Step-by-Step Cookbook*, page 196. Meredith, Des Moines, IA.

Kukich, K. (1983). *Knowledge-Based Report Generation: A Knowledge-Engineering Approach to Natural Language Report Generation*. PhD thesis, University of Pittsburgh.

Kukich, K. (1988). Fluency in natural language reports. In McDonald, D. D. and Bloc, L., editors, *Natural Language Generation Systems*, chapter 8. Springer Verlag.

Labov, W. (1972). The transformation of experience in narrative syntax. In Labov, W., editor, *Language in the Inner City: Studies in the Black English Vernacular*. University of Pennsylvania Press, Philadelphia, PA.

Labov, W. and Waletzky, J. (1967). Narrative analysis: oral versions of personal experience. In Helm, J., editor, *Essays on the Verbal and Visual Arts: Proceedings of the 1966 Annual Spring Meeting of the American Ethnological Society*, pages 12–44. University of Washington Press, Seattle, WA.

Lampert, J. A. and Lampert, M. D., editors (1993). *Talking Data: Transcription and Coding in Discourse Research*. Lawrence Erlbaum Associates.

LeFevre, J.-A. and Dixon, P. (1986). Do written instructions need examples. *Cognition and Instruction*, 3(1):1–30.

Lehrberger, J. (1982). Automatic translation and the concept of sublanguage. In Kittredge, R. and Lehrberger, J., editors, *Sublanguage: Studies of Language in Restricted Semantic Domains*, Foundations of Communication, chapter 3, pages 81–106. de Gruyter, Berlin.

Liberman, M. and Marcus, M. (1992). Tutorial on text corpora. ACL'92.

Livers, E. (1992). 50 ways to do it better. *Women's Day*, page 100. November 3.

Longacre, R. E. (1976). *An Anatomy of Speech Notions*. Peter De Ridder Press, Lisse.

Longacre, R. E. (1983). *The Grammar of Discourse*. Plenum Press, New York.

LOOM (1991). *LOOM Users Guide*. ISX Corporation, Draft.

Macintosh (1988). *Macintosh System Software User's Guide: Version 6.0*. Apple Computer, Inc.

Malinowski, B. (1923). The problem of meaning in primitive languages. In Ogden, C. K. and Richards, I. A., editors, *The Meaning of Meaning*, pages 296–336. Routledge and Kegan Paul, London.

Mann, W. C. (1985). An introduction to the Nigel text generation grammar. In Benson, J. D., Freedle, R. O., and Greaves, W. S., editors, *Systemic Perspectives on Discourse*, volume 1, pages 84–95. Ablex.

Mann, W. C. and Matthiessen, C. M. I. M. (1985). A demonstration of the Nigel text generation computer program. In Benson, J. D., Freedle, R. O., and Greaves, W. S., editors, *Systemic Perspectives on Discourse: Selected Theoretical Papers from the 9th International Systemic Workshop*, volume 1. Ablex.

Mann, W. C. and Thompson, S. A. (1986). Relational propositions in discourse. *Discourse Processes*, 9(1):57–90.

Mann, W. C. and Thompson, S. A. (1987a). Rhetorical structure theory: a framework for the analysis of texts. *IPRA Papers in Pragmatics*, 1. Also published as USC tech report ISI/RS-87-185.

Mann, W. C. and Thompson, S. A. (1987b). Rhetorical structure theory: Description and construction of text structures. In Kempen, G., editor, *Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics*. NATO Scientific Affairs Division, Martinus Nijhoff.

Mann, W. C. and Thompson, S. A. (1988). Rhetorical structure theory: Toward a functional theory of text organization. *Text: An Interdisciplinary Journal for the Study of Text*, 8(2).

Mann, W. C. and Thompson, S. A. (1989). Rhetorical structure theory: A theory of text organization. In Polanyi, L., editor, *The Structure of Discourse*. Ablex, Norwood, NJ. To appear, currently available as ISI tech. report ISI/RS-87-190.

Mann, W. C. and Thompson, S. A. (1992). *Discourse Description: Diverse Linguistic Analyses of a Fund-Raising Text*. John Benjamins, Amsterdam.

Martin, J. R. (1983). Conjunction: The logic of English text. In Petöfi, J. S. and Sözer, E., editors, *Micro and Macro Connexity of Texts*, pages 1–72. Helmut Buske Verlag, Hamburg.

Matthiessen, C. M. I. M. (1985). The systemic framework in text generation: Nigel. In Benson, J. D., Freedle, R. O., and Greaves, W. S., editors, *Systemic Perspectives on Discourse: Selected Theoretical Papers from the 9th International Systemic Workshop*, volume 1. Ablex.

Matthiessen, C. M. I. M. (1988). *What's in Nigel: Lexicogrammatical Cartography*. USC Information Sciences Institute.

Matthiessen, C. M. I. M. and Thompson, S. A. (1987). The structure of discourse and 'subordination'. In Haiman, J. and Thompson, S., editors, *Clause Combining in Grammar and Discourse*, pages 275–329. Benjamins, Amsterdam.

McCoy, K. F. (1985). *Correcting Object-Related Misconceptions*. PhD thesis, University of Pennsylvania.

McDonald, D. D. (1980). *Natural Language Production as a Process of Decision Making*. PhD thesis, MIT, Cambridge, MA.

McDonald, D. D. (1983). Description directed control: Its implications for natural language generation. *Computers and Mathematics*, 9(1):111–130. Also included in *Readings in Natural Language Processing*, Grosz et. al. editors, Morgan Kaufmann, 1986.

McDonald, D. D. and Pustejevsky, J. (1985). Description-directed natural language generation. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, Los Angeles, CA, pages 799–805.

McKeown, K. R. (1982). *Generating Natural Language Text in Response to Questions About Database Queries*. PhD thesis, University of Pennsylvania.

McKeown, K. R. (1985). *Text Generation*. Cambridge University Press, New York.

McKeown, K. R., Elhadad, M., Fukumoto, Y., Lim, J., Lombardi, C., Robin, J., and Smadja, F. (1990). Natural language generation in COMET. In Dale, R., Mellish, C., and Zock, M., editors, *Current Research in Natural Language Generation*, chapter 5. Academic Press.

McKeown, K. R. and Swartout, W. R. (1988). Language generation and explanation. In Zock, M. and Sabah, G., editors, *Advances in Natural Language Generation - An Interdisciplinary Perspective*, volume 1, chapter 1. Ablex.

MCL (1989). *Macintosh Allegro Common Lisp 1.3 Reference*. Apple Computer, Inc., 20525 Mariani Avenue, Cupertino, CA 95014-6299. APDA #M0067LL/C.

Mellish, C. (1988). Natural language generation from plans. In Zock, M. and Sabah, G., editors, *Advances in Natural Language Generation An Interdisciplinary Prespective*, volume 1, chapter 7. Ablex. Selected readings from the 1st European NLG Workshop, the Abbaye de Rouaumont, 1987.

Mellish, C. and Evans, R. (1989). Natural language generation from plans. *Computational Linguistics*, 15(4):233–249.

Meteer, M. W. (1990). *The "Generation Gap": The Problem of Expressibility in Text Planning*. PhD thesis, University of Massachusetts.

Meteer, M. W. (1991). Bridging the generation gap between text planning and linguistic realization. *Computational Intelligence*, 7(4):296–304.

Meteer, M. W., McDonald, D. D., Andersen, S., Forster, D., Gay, L., Huettner, A., and Sibun, P. (1987). Mumble-86: Design and implementation. Technical Report 87-87, University of Massachusetts.

Meyer, B. J. F. (1975). *The Organization of Prose and its Effects on Memory*. North-Holland Publishing Co., Amsterdam.

Meyer, B. J. F. and Rice, G. E. (1982). The interaction of reader strategies and the organization of text. *Text*, 2(1):155–192.

Moens, M. and Steedman, M. (1988). Temporal ontology and temporal reference. *Computational Linguistics*, 14(2).

Mooney, D. J., Carberry, S., and McCoy, K. F. (1990). The basic block model of extended explanations. In McKeown, K. and an Sergei Nirenburg, J. M., editors, *Proceedings of the Fifth International Workshop on Natural Language Generation*, June 3–6, Dawson, PA.

Mooney, D. J., Carberry, S., and McCoy, K. F. (1991). Capturing high-level structure of naturally occurring, extended explanation using bottom-up strategies. *Computational Intelligence*, 7(4):334–356.

Moore, J. D. and Paris, C. L. (1988). Constructing coherent text using rhetorical relations. In *Proceedings of the Tenth Annual Conference of the Cognitive Science Society*, August 17–19, Montreal, Quebec, pages 637–643.

Moore, J. D. and Paris, C. L. (1989). Planning text for advisory dialogues. In *Proceedings of the 27th Annual Meeting of Association for Computational Linguistics*, June 26–29, Vancouver, B.C. Also available as ISI tech. report ISI/RR-89-236.

Moore, J. D. and Pollack, M. E. (1992). A problem for RST: The need for multi-level discourse analysis. *Computational Linguistics*, 18(4):537–544. Squibs and Discussions.

Nida, E. A. (1984). Rhetoric and styles: A taxonomy of structures and functions. *Language Sciences*, 6(2):287–305.

Nigel (1988). *The Nigel Manual*. USC Information Sciences Institute, Penman Natural Language Group. Draft.

Nogier, J.-F. and Zock, M. (1991). Lexical choice as pattern matching. In Nagle, T., Nagle, J., Gerholz, L., and P., E., editors, *Current Directions in Conceptual Structures Research*. Springer-Verlag, Berlin. to appear.

Panasonic (1987). Operating instructions. Model RC-6065.

Paris, C. (1987). *The Use of Explicit User Models in Text Generation: Tailoring to a User's Level of Expertise*. PhD thesis, Columbia University.

Paris, C. L. (1988). Tailoring object descriptions to a user's level of expertise. *Computational Linguistics*, 14(3):64–78.

Paris, C. L. and Bateman, J. A. (1990). User modeling and register theory: A congruence of concerns. Draft – Submitted to Journal of User Modeling.

Paris, C. L. and Maier, E. (1991). Knowledge resources or decisions. In *Proceedings of the IJCAI-91 Workshop on Decision Making Throughout the Generation Process*, August 24–25, Darling Harbor, Sydney, Australia.

Paris, C. L. and McKeown, K. R. (1987). Discourse strategies for describing complex physical objects. In Kempen, G., editor, *Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics*. NATO Scientific Affairs Division, Martinus Nijhoff. Proceedings of the NATO Advanced Research Workshop on Natural Language Generation, Nijmegen, The Netherlands, August 19-23, 1986.

Patten, T. (1988). *Systemic Text Generation as Problem Solving*. Cambridge University Press.

Pearson, P. D., Barr, R., Kamil, M. L., and Mosenthal, P., editors (1984). *Handbook of Reading Research*. Longman, New York.

Penman (1989). *The Penman Documentation*. USC Information Sciences Institute, Penman Natural Language Group.

Post, E. (1943). Formal reductions of the general combinatorial decision problem. *American Journal of Mathematics*, 65:197–268.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1:81–106.

Quirk, R., Greenbaum, S., Leech, G., and Svartvik, J. (1985). *A Comprehensive Grammar of the English Language*. Longman.

Rambow, O. (1990). Domain communication knowledge. In McKeown, K. and an Sergei Nirenburg, J. M., editors, *Proceedings of the Fifth International Workshop on Natural Language Generation*, June 3–6, Dawson, PA.

Reader's Digest (1973). *Reader's Digest Complete Do-It-Yourself Manual*. The Reader's Digest Association, Inc., Pleasantville, NY.

Reader's Digest (1981). *Reader's Digest Complete Car Care Manual*. The Reader's Digest Association, Inc., Pleasantville, NY.

Reichman-Adar, R. (1984). Extended person–machine interface. *Artificial Intelligence*, 22:157–218.

Reichman, R. (1981). *Plain Speaking: A Theory and Grammar of Spontaneous Discourse*. PhD thesis, Harvard University. This thesis is Bolt Beranek and Newman (BBN) report number 4681, BBN Inc., 10 Moulton Street, Cambridge, MA 02217.

Reiter, E. (1990). A new model for lexical choice for open-class words. In McKeown, K. F., Moore, J. D., and Nirenburg, S., editors, *Proceedings of the Fifth International Workshop on Natural Language Generation*, June 3–6, Dawson, PA, pages 23–30.

Reiter, E. and Mellish, C. (1992). Using classification to generate text. In *Proceedings of the Annual Meeting of Association for Computational Linguistics*, Newark, DE, pages 265–272.

Reiter, E., Mellish, C., and Levine, J. (1992). Automatic generation of on-line documentation in the IDAS project. In *Proceedings of the Third Conference on Applied Natural Language Processing*, pages 64–71.

Rettig, M. (1991). Nobody reads documentation. *Communications of the ACM*, 34(7):19–24.

Robin, J. (1992). Generating newswire report leads with historical information: A draft and revision approach. Thesis proposal, Columbia University.

Rock, S. (1992). Understanding repetition in natural language instructions — the semantics of extent. In *Proceedings of the Annual Meeting of Association for Computational Linguistics*, (Student Session), Newark, DE, pages 279–281.

Rose, A. M. and Cox Jr., L. A. (1980). Following instructions. Technical Report 4, Document Design Project, Washington, D.C. American Institutes for Research.

Rosenberg, S. N. (1985). *The Johnson and Johnson First Aid Book*, pages 17,95,101,129. Johnson and Johnson.

Rösner, D. (1987). The automated news agency: SEMTEX — a text generator for German. In Kempen, G., editor, *Natural Language Generation: New Results in Artificial Intelligence, Psychology, and Linguistics*, pages 133–148. Martinus Nijhoff. Proceedings of the NATO Advanced Research Workshop on Natural Language Generation, Nijmegen, The Netherlands, August 19-23, 1986.

Rösner, D. and Stede, M. (1992a). Customizing RST for the automatic production of technical manuals. In Dale, R., Hovy, E., Röesner, D., and Stock, O., editors, *Aspects of Automated Natural Language Generation*, Lecture Notes in Artificial Intelligence 587, pages 199–214. Springer Verlag, Berlin. Proceedings of the 6th International Workshop on Natural Language Generation, Trento, Italy, April, 1992.

Rösner, D. and Stede, M. (1992b). TECHDOC: A system for the automatic production of multilingual technical manuals. submitted.

Sacerdoti, E. D. (1973). The non-linear nature of plans. In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, Palo Alto, CA.

Sacerdoti, E. D. (1977). *A Structure for Plans and Behavior*. Elsevier, New York.

Sager, N. (1981). *Natural Language Information Processing: A Computer Grammar of English and Its Applications*. Addison-Wesley, Reading, MA.

Sager, N. (1986). Sublanguage: Linguistic phenomenon, computational tool. In Grishman, R. and Kittredge, R., editors, *Analyzing Language in Restricted Domains: Sublanguage Description and Processing*, chapter 1, pages 1–17. Lawrence Erlbaum Associates, Hillsdale, NJ.

Sanders, T. J. M., Spooren, W. P. M. S., and Noordman, L. G. M. (1992). Towards a taxonomy of coherence relations. *Discourse Processes*, 15:1–35.

Sanders, T. J. M., Spooren, W. P. M. S., and Noordman, L. G. M. (1993). Coherence relations in a cognitive theory of discourse representation. *Cognitive Linguistics*. to appear.

Sawyer, M. H. (1991). A review of research in revising instructional text. *Journal or Reading Behavior*, 23(3):307–333.

Scott, D. R. and Souza, C. (1990). Getting the message across in RST-based text generation. In Dale, R., Mellish, C., and Zock, M., editors, *Current Research in Natural Language Generation*, chapter 3. Academic Press.

Sharp (1986). Operation manual. Sharp Plaza, Mahwah, New Jersey 07430. Model 13LM16.

Smadja, F. (1989a). Lexical co-occurrence. the missing link. *Journal of Literary and Linguistic Computing*, 4(3):163–168.

Smadja, F. (1989b). Microcoding the lexicon with co-occurrence knowledge. In *Proceedings of the First International Workshop on Lexical Acquisition*. Detroit, MI.

Smadja, F. and McKeown, K. (1991). Using collocations for language generation. *Computational Intelligence*, 7(4):229–239.

Steinberg, E. R., editor (1991). *Plain Language: Principles and Practice*. Wayne State University Press, Detroit.

Strunk Jr., W. and White, E. B. (1979). *The Elements of Style*. McMillan, New York, third edition.

Tate, A. (1976). *Project Planning Using a Hierarchical Non-Linear Planner*. PhD thesis, University of Edinburgh. Dissertation Abstracts International Report 25.

Thompson, S. A. (1985). Grammar and written discourse: Initial and final purpose clauses in English. *Text*, 5(1–2):55–84.

Thompson, S. A. (1987). "Subordination" and narrative event structure. In Tomlin, R. S., editor, *Coherence and Grounding in Discourse*, pages 435–454. John Benjamins Publishing Co. Outcome of a Symposium, Eugene, Oregon, June, 1984, and published as vol. 11 of the series Typological Studies in Language.

Thompson, S. A. (1988). Information flow and 'dative shift' in English discourse. In Thompson, S. A., editor, *Santa Barbara Papers in Linguistics*, volume 2, pages 147–163. University of Santa Barbara.

Thorndike, E. L. (1921). *The Teacher's Word Book*. Columbia University, Teacher's College, New York.

Ure, J. N. and Ellis, J. (1974). Register in descriptive linguistics and linguistic sociology. In Uribe-Vellegas, O., editor, *Issues in Sociolinguistics*. Mouton.

van Beek, P. and Cohen, R. (1990). Resolving plan ambiguity for response generation. In McKeown, K. and an Sergei Nirenburg, J. M., editors, *Proceedings of the Fifth International Workshop on Natural Language Generation*, June 3–6, Dawson, PA.

van Dijk, T. A. and Kintsch, W. (1983). *Strategies of Discourse Comprehension*. Academic Press, New York.

Vander Linden, K. (1992). The expression of local rhetorical relations in instructional text. In *Proceedings of the Annual Meeting of Association for Computational Linguistics*, (Student Session), Newark, DE, pages 318–320.

Vander Linden, K. (1993). Generating effective instructions. In *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*, June 18–21, Boulder, CO, pages 1023–1028.

Vander Linden, K., Cumming, S., and Martin, J. (1992a). The expression of local rhetorical relations in instructional text. Technical Report CU-CS-585-92, the University of Colorado.

Vander Linden, K., Cumming, S., and Martin, J. (1992b). Using system networks to build rhetorical structures. In Dale, R., Hovy, E., Rösner, D., and Stock, O., editors, *Aspects of Automated Natural Language Generation*, Lecture Notes in Artificial Intelligence 587, pages 183–198. Springer Verlag, Berlin. Proceedings of the 6th International Workshop on Natural Language Generation, Trento, Italy, April, 1992.

Volkswagon (1987). *Volkswagon GTI, Golf, and Jetta Official Factory Repair Manual*. Robert Bentley, Inc., Cambridge, Massachusetts.

Wagenaar, W. A., Schreuder, R., and Wijlhuizen, G. J. (1987). Readability of instructional text, written for the general public. *Applied Cognitive Psychology*, 1:155–167.

Wahlster, W., André, E., Finkler, W., Profitlich, H.-J., and Rist, T. (1993). Plan-based integration of natural language and graphic generation. *Artificial Intelligence*. to appear.

Waller, R. H. W. (1979). Typographic access structures for educational texts. In Kolers, P. A., Wrolstad, M. E., and Bouma, H., editors, *Processing of Visible Language, 1*, pages 175–187. Plenum Press, New York.

Wanner, L. and Bateman, J. A. (1990). A collocational based approach to salience-sensitive lexical selection. In McKeown, K. F., Moore, J. D., and Nirenburg, S., editors, *Proceedings of the Fifth International Workshop on Natural Language Generation*, pages 31–38.

Ward, N. (1990). A connectionist treatment of grammar for generation: Relying on emergents. In *Proceedings of the Fifth International Workshop on Natural Language Generation*, June 3–6, Dawson, PA.

Ward, N. (1991). *A Flexible, Parallel Model of Natural Language Generation*. PhD thesis, University of California, Berkeley. Report Number UCB/CSD 91/629.

Waste King (1979). Operating instructions for the waste king pulverator.

Webber, B. L. and Di Eugenio, B. (1990). Free adjuncts in natural language instructions. In Karlgren, H., editor, *Proceedings of the 13th International Conference on Computational Linguistics, Helsinki, Finland*.

Weiss, E. H. (1988). Breaking the grip of user manuals. *Asterisk – Journal of the ACM SIGDOC*, 14(2):4–11.

Weiss, E. H. (1991). *How to Write Usable User Documentation*. Oryx Press, second edition.

Williams, A. D. (1991). Jigsaw puzzles: Pictures on plywook scroll sawn to pieces. *Fine Woodworking*, 88:52–55.

Winograd, T. (1972). *Understanding Natural Language*. Academic Press, New York.

Winograd, T. (1983). *Language as a Cognitive Process, Volume I: Syntax*. Addison-Wesley, Reading, MA.

Woods, W. A. (1973). An experimental parsing system for transition network grammars. In Rustin, R., editor, *Natural Language Processing*. Algorithmics Press, New York.

Wright, P. (1981). "The instructions clearly state . . ." Can't people read? *Applied Ergonomics*, 12(3):131–141.

Wright, P. and Wilcox, P. (1978). Following instructions: An exploratory trisection of imperatives. In Levelt, W. J. M. and Flores d'Arcais, G. B., editors, *Studies in the Perception of Language*, chapter 4, pages 129–153. John Wiley & Sons, Chichester.

Ying, M. (1992). The holiday roast cookbook. *Good Housekeeping*, page 170. December.

Young, D. (1985). Some applications of systemic grammar to TEFL or whatever became of register analysis. In Benson, J. D. and Greaves, W. S., editors, *Systemic Perspectives on Discourse*, volume 2: Selected Theoretical Papers from the 9th International Systemic Workshop, chapter 18, pages 282–294. Ablex.

Zinsser, W. K. (1985). *On Writing Well: An Informal Guide to Writing Nonfiction*. Harper and Row, New York.

Appendix A

The Source Texts

This appendix contains a listing of all the source texts in the corpus, and the code name used for each one in the analysis. These code names are used in the results analysis discussed in chapter 7. The chart also gives the number of clauses that each of the passages includes.

Description	Code	# of Clauses	Reference
The Airfone Manual	aph†	64	(Airfone, 1991)
An Auto Repair Guide	ar1	15	(Reader's Digest, 1981)
	ar2	23	
	ar3	25	
The Bubble Plane Manual	bpl†	48	(Bubble plane, 1991)
A Clock-Radio Manual	cr1†	149	(Panasonic, 1987)
A First Aid Manual	fa1	23	(Rosenberg, 1985)
	fa2	13	
	fa3	18	
	fa4	7	
A Floor Covering Manual	fc	12	(Reader's Digest, 1973)
Instructions for Raku Pottery	hr1	18	(Gillberg, 1992)
Instructions for Making Puzzles	jp1	55	(Williams, 1991)
	jp2	16	
Macintosh Software Instructions	ms1	42	(Macintosh, 1988)
A Cordless Telephone Manual	ph1†	148	(Code-a-phone, 1989)
Another Telephone Manual	ph2†	113	(Excursion, 1989)
A Beef & Barley Soup Recipe	r1†	30	(Ying, 1992)
A Bagel Recipe	r2†	42	(Knox, 1978)
A TV Operations Manual	tv1†	48	(Sharp, 1986)
Another Auto Repair Manual	vw1	15	(Volkswagon, 1987)
	vw2	14	
Instructions for Using a Hammer	wd1	14	(Livers, 1992)
Instructions for a Garbage Disposal	wp†	17	(Waste King, 1979)

† The procedural and primitive portions of the instructions from these sources are included in their entirety; the corpus entries for the other sources contain only selected passages.

Appendix B

Value Ranges for The Database Fields

This appendix contains a listing of the fields in each of the tables implemented in the corpus database. It also includes a listing of all the possible values for each field that selects its values from a fixed list.

Table B.1: Quick reference of Clause table field types

Field Name	*Field Type*	*Field Size*	*Description*
SOURCE	Character	003	The source text of the clause
INDEX	Number	008	The clause number within the source text
PAGE	Number	005	The page where the example can be found
LINKER	Character	012	Any clause linker used for this clause
ADVERB	Character	012	Any adverb used
TENSE	Character	003	The Tense of the clause
MODAL	Character	010	Any modal used
HAVE_EN	Character	001	Marks the Have-En form
BE_ING	Character	001	Same for Be-Ing
BE_EN	Character	001	Same for Be-En
VERBSTEM	Character	012	The stem of the main verb
POLARITY	Character	001	The polarity of the predicate
SEMCLASS	Character	006	The semantic class of the predication
REFERENCE	Character	012	The referenced predication
LOOKBACK	Number	005	The computed lookback distance
NOUNHEAD	Number	005	The head NP for relative clauses
CLAUSEHEAD	Number	005	The head clause for hypo- and parataxis
ROLE	Character	006	The semantic role
SLOT	Character	001	The slot of the clause
AGENT	Character	001	The agent of the action (reader or non-reader)
ARG_1	Number	005	The Phrase number of the subject
ARG_2	Number	005	The Phrase number of the direct object
ARG_3	Number	005	The Phrase number of the indirect object
TAG	Character	001	A utility field
TEXT	Character	254	The raw text of the clause.
ABSNUM	Number	005	Another utility field

Table B.2: The possible values for TENSE in the Clause table

Possible Value	*Description*
ing	-ing form
ppt	past participle
prs	present participle
pre	simple present
pst	simple past
fut	simple future
tnf	to infinitive
imp	imperative
nom	nominalization
-	no verb, no nominalization

Table B.3: The possible values for SEMCLASS in the Clause table

Possible Value	*Description*
MAT	Material (physical) action
SPEECH	Verbal action
MENT	Mental action - cognition etc.
RELAT	Relational – copulas etc.
EXIST	Existential – there be
?	Don't know

Table B.4: The possible values for ROLE in the Clause table

Possible Value	*Description*
-	not conjoined
ADV	Adverbial
CMP	Complement
CONJ	Conjunction
REL	Relative
?	Unknown

Table B.5: Quick Reference of Phrase Table Field Types

Field Name	*Field Type*	*Field Size*	*Description*
SOURCE	Character	003	The source text of the phrase
INDEX	Number	008	The phrase number within the source text
SEMCLASS	Character	006	The semantic class of the phrase
TYPE	Character	006	The type of the embedded noun phrase
NUMBER	Character	001	The number of the embedded NP
PREP	Character	012	The preposition used (if any)
DET	Character	005	Any determiner used
QUANT	Character	006	Any quantifiers used
ADJ	Character	012	Any adjectives used
REL_PP	Character	001	Marker for relative clauses or PP modifiers
REFERENCE	Character	014	The referent of the noun phrase
LOOKBACK	Number	005	The distance looked back by the reference
CLAUSE_HD	Number	005	The clause to which this is an phrase
NP_HD	Number	005	The noun head adjectival NPs and PPs
ROLE	Character	006	The role of the phrase
SLOT	Character	001	The slot of the phrase
TAG	Character	001	A field used to mark questions or problems
TEXT	Character	100	The raw text of the phrase

Table B.6: The possible values for SEMCLASS in the Argument table

Possible Value	*Description*
HUM	Human
INAN	Inanimate
ORG	Organization
NATION	Nation
BODY	Body part
TIME	Time expression
PLACE	Place expression (location)
PROP	Proposition (verbal/mental)
PART	Part of (a number of...)
DUM	Dummy IT (Subj. or Obj.)
ABS	Abstract entity
EVENT	Event (nominalization etc.)
?	Don't know

Table B.7: The possible values for TYPE in the Argument table

Possible Value	*Description*
?	Can't tell
ADJ	Referring adjective
LEX NP	with lexical head
NAME	Name of person, place etc.
PRN	Personal pronoun
DUMP	Dummy pronoun IT
RPRN	Relative pronoun
DPRN	Demonstrative pronoun
IPRN	Indefinite pronoun
TIME	Time expression: dates etc.

Table B.8: The possible values for NUMBER in the Argument table

Possible Value	*Description*
S	Singular
P	Plural
M	Mass
-	No number (eg adjs)
?	Can't tell

Table B.9: The possible values for REL_PP in the Argument table

Possible Value	*Description*
R	Relative clause
L	Participial clause
T	Infinitival phrase
P	Prepositional phrase
M	More than one p.m.
C	Noun comp clause
-	No postmodifier
?	Not sure

Table B.10: The possible values for ROLE in the Argument table

Possible Value	*Description*
Comp	nominalized complements of nominalizations
A	
S	
O	
SO	For unaccusatives, e.g. open, extend
AS	For unergatives, e.g. attack, eat
AO	e.g. Bob-Bill hostility
1	Logical subject (A or S)
2	Logical object (P/O
3	Indirect object, recipient etc.
Pred	Predicate NP (comp. of "be" etc.)
Loc	Place
Time	Time
Inst	Instrumental
Com	Comitative
Obl	Other obliques
Poss	Possessor (generalized)
Mod	Noun Modifier
Adj	Adjective
App	Apposed NP
Conj	Conjoined NP
?	Can't tell

Table B.11: Quick reference of RST table field types

Field Name	*Field Type*	*Field Size*	*Description*
SOURCE	Character	003	The source text of the phrase
SYNTAX_IDX	Number	008	The current clause or phrase
SYNTAX_CD	Character	001	The type of that entry
CHILD	Number	008	The index of the child of the source
CHILD_CD	Character	001	The type of the child
RELATION	Character	012	The type of RST relation
NUCLEUS	Number	008	The nucleus of the relation
NUCLEUS_CD	Character	001	The type of the nucleus

Table B.12: The possible values for SYNTAX_CD, CHILD_CD, and NUCLEUS_CD in the RST table

Possible Value	*Description*
I	Internal (high level node)
C	Leaf assoc. with a clause
P	Leaf that is an argument
?	To be filled in later.

Table B.13: The possible values for RELATION in the RST table

Possible Value	*Description*
elaboration	elaboration
title	Title elaboration
joint	Joint schema
joint-and	Logical and
joint-or	Logical or
funct	Functional-Unknown order
funct-use	Functional-usefulness
funct-alt	Functional-alternatives
funct-pop	Functional-popularity
sequence	Procedural sequence
enablement	enablement
purpose	purpose
inform	Sat is information given
evidence	Proc. to verify action
background	Background material
precond	RST condition
circ	Circumstance
postcond	RST result (vol. or not)
cocond	Simultaneous condition
motive	Motivation
example	Exemplification (elab.)
contrast	Contrasting information
restate	Restatement

Appendix C

An Annotated Run of IMAGENE

This appendix contains a full, annotated run of the Remove-Phone text. The original text from the manual is repeated here for reference:

Remove-Phone Text:

> When instructed (approx. 10 sec.) remove phone by firmly grasping top of handset and pulling out. Return to seat to place calls.

C.1 The PRL Input File

Here is the Process Representation Language input file that is used for this example:

```
;;; -*- Syntax: Common-lisp; Base: 10; Package: PRL; Mode: LISP -*-
(in-package 'prl)

; Here is the PRL structure for example pp5, "When you are instructed,
;   remove phone by grasping the top of the handset and pulling it.
;   Return to seat to place a call."
;
(defkb current-prl-kb (prl-kb))
(change-kb 'current-prl-kb)

(tell (:about *prl-root* Action
              (subaction instruct-action)
              (subaction remove-action)
              (subaction place-action)))

(tell (:about Instruct-Action Action
              (action-type it::Instruct)
              (actor phone)
              (actee hearer)))

(tell (:about phone Object
              (object-type it::phone)
              (object-status device)))

(tell (:about hearer Object
              (object-type it::hearer)))

(tell (:about remove-action Action
              (subaction grasp-action)
              (subaction pull-action)
              (action-type it::Remove)
              (actor hearer)
              (actee phone)))
```

```
(tell (:about Grasp-Action Action
              (action-type it::Grasp)
              (actor hearer)
              (actee top)))

(tell (:about top Object
              (object-type it::top)
              (object-status device)
              (object-possessor handset)))

(tell (:about handset Object
              (object-type it::handset)
              (object-status device)))

(tell (:about Pull-Action Action
              (action-type it::Pull)
              (actor hearer)
              (actee handset)))

(tell (:about place-action Action
              (subaction return-action)
              (action-type it::Place-Call)
              (actor hearer)
              (actee call)))

(tell (:about call Object
              (object-type it::call-noun)))

(tell (:about return-action Action
              (action-type it::return-movement)
              (actor hearer)
              (destination seat)))

(tell (:about seat Object
              (object-type it::seat)
              (owner hearer)))
```

C.2 The Trace Output

This section includes the complete trace of the IMAGENE run for the remove-phone text. This run includes indications of the realization statements used by IMAGENE. The realization statements used by Penman have been edited out and some of the output has been formated for presentation purposes, no other changes have been made.

```
? (ig)

          ACTION-TYPE-Q: Is IG-TEXT a single project, the root node of the
          text, or a set of primitives?
Enter inquiry answer:
 1. PROJECT
 2. ROOT
 3. PRIMITIVES
Number Of Choice: 2

          CONCURRENCY-STRUCTURE-Q: Is IG-TEXT a procedure with concurrency
          that must be expressed?  If concurrency is involved, is the node a
          concurrent parent (that is does the node have concurrent children)
          or a component of a concurrent structure (that is a node with with
          procedural children)?
Enter inquiry answer:
 1. CONCURRENT
 2. CONCURRENT-COMPONENT
 3. NOT-CONCURRENT
```

```
Number Of Choice: 3

      In CONCURRENCY-STRUCTURE,
Realizing: (ITERATIVE-INSERT ONUS PROCEDURAL-SUBSPAN)

          CHILDREN-COUNT: How many subactions for IG-TEXT will be expressed (0
          means stop)?
Enter inquiry answer: 3

          TEXT-ELEMENT-ID: What is the concept which is associated with the
          subordinate or satellite text elements (in order if applicable) of
          IG-TEXT?  This inquiry will be made once for each subaction if
          necessary.
Enter inquiry answer: instruct-action

          TEXT-ELEMENT-ID: What is the concept which is associated with the
          subordinate or satellite text elements (in order if applicable) of
          IG-TEXT?  This inquiry will be made once for each subaction if
          necessary.
Enter inquiry answer: remove-action

          TEXT-ELEMENT-ID: What is the concept which is associated with the
          subordinate or satellite text elements (in order if applicable) of
          IG-TEXT?  This inquiry will be made once for each subaction if
          necessary.
Enter inquiry answer: place-action

          ACTION-TYPE-Q: Is INSTRUCT-ACTION a single project, the root node of
          the text, or a set of primitives?
Enter inquiry answer:
 1. PROJECT
 2. ROOT
 3. PRIMITIVES
Number Of Choice: 1

          READER-KNOWLEDGE-Q: Is INSTRUCT-ACTION a procedural sequence that
          the reader is assumed to know?
Enter inquiry answer:
 1. KNOWN
 2. UNKNOWN
Number Of Choice: 1

          NODE-TYPE-Q: Is INSTRUCT-ACTION an action coupled with other
          concurrent actions in a concurrent structure or not?
Enter inquiry answer:
 1. NOT-CONCURRENT-NODE
 2. CONCURRENT-NODE
Number Of Choice: 1

          ACTION-ACTOR-Q: Is INSTRUCT-ACTION being performed by the reader or
          some other agent?  The actor is some other agent only if you know it
          is not performed by the reader.
Enter inquiry answer:
 1. READER
 2. NONREADER
Number Of Choice: 2

          ACTION-MONITOR-RESULT-Q: Is INSTRUCT-ACTION to be expressed as the
          result of a monitor action (either by the reader or some other
          agent?
Enter inquiry answer:
 1. MONITOR-RESULT
 2. NOT-MONITOR-RESULT
Number Of Choice: 1

          PRECOND-COORDINATION-STATUS-Q: Is INSTRUCT-ACTION already the
          nucleus of a precondition relation?
```

```
Enter inquiry answer:
 1. PREVIOUS-PRECOND
 2. NOT-PREVIOUS-PRECOND
Number Of Choice: 2


      In ADD-PRECONDITION,
Realizing: (STRUCTURE FOLLOWING-NONTRACE-NODE ONUS PRECONDITION)

      In ADD-PRECONDITION,
Realizing: (UNLINK ONUS JOINT-SCHEMA AS-RANGE)

          PRECOND-RHETORICAL-CONTEXT-Q: Is the combination of INSTRUCT-ACTION
          and the action to which it pertains part of a sequential or
          functional list or not?
Enter inquiry answer:
 1. SIMPLE-CONTEXT
 2. COMPLEX-CONTEXT
Number Of Choice: 2


      In PRECOND-RHETORICAL-CONTEXT,
Realizing: (COPY NEXT-NODE INTERNAL-PRECOND)

      In PRECOND-RHETORICAL-CONTEXT,
Realizing: (UNLINK INTERNAL-PRECOND GRAMMATICAL-FEATURE AS-DOMAIN)

      In PRECOND-RHETORICAL-CONTEXT,
Realizing: (UNLINK INTERNAL-PRECOND SENTENCE-TRACE BOTH)

      In PRECOND-RHETORICAL-CONTEXT,
Realizing: (UNLINK INTERNAL-PRECOND SCHEMA AS-DOMAIN)

      In PRECOND-RHETORICAL-CONTEXT,
Realizing: (UNLINK NEXT-NODE SCHEMA AS-RANGE)

      In PRECOND-RHETORICAL-CONTEXT,
Realizing: (STRUCTURE INTERNAL-PRECOND NEXT-NODE PROCEDURAL-SUBSPAN)

          PRECOND-ACT-TOPIC-Q: Is the action that INSTRUCT-ACTION pertains to
          a local topic or not, that is, is the action the onus of the purpose
          of including this portion of text?
Enter inquiry answer:
 1. LOCAL-TOPIC
 2. NOT-LOCAL-TOPIC
Number Of Choice: 2

          PRECOND-CHANGEABLE-TYPE-Q: Does the action or condition
          INSTRUCT-ACTION come at the beginning of a procedural sequence and
          is it something which the reader is capable of changing at this
          time?
Enter inquiry answer:
 1. CHANGEABLE
 2. NOT-CHANGEABLE
Number Of Choice: 2


      In PRECOND-CHANGEABLE-TYPE,
Realizing: (COMBINE ONUS NUCLEUS-NODE CONTINUE-SENTENCE)

          PRECOND-PREVIOUS-ACT-TYPE-Q: Is the action or condition
          INSTRUCT-ACTION a monitor action, giving action, placing action,
          habitual decision, or effective action?
Enter inquiry answer:
 1. MONITOR-ACT
 2. GIVING-ACT
 3. PLACING-ACT
 4. HABITUAL-DECISION
```

```
 5. EFFECTIVE-ACT
Number Of Choice: 5

          TEMPORAL-ORIENTATION-Q: Is the action INSTRUCT-ACTION one which was
          performed at a temporally remote time in the past?  If you are not
          sure, then it is NOT temporally remote prior.
Enter inquiry answer:
 1. TEMPORALLY-REMOTE-PRIOR
 2. NOT-TEMPORALLY-REMOTE-PRIOR
Number Of Choice: 2

          PROCEDURAL-OBVIOUSNESS-Q: Is INSTRUCT-ACTION procedurally obvious or
          not.  An action is procedurally obvious if the current local
          discourse is about that action and the reader is expected to be
          familiar with it.
Enter inquiry answer:
 1. OBVIOUS
 2. NOT-OBVIOUS
Number Of Choice: 2

          PRECOND-ACT-HIDE-Q: Is the action INSTRUCT-ACTION something that
          should be hidden from the reader?  That is, is it an overly
          complicated, non-reader performed action whose details are not
          relevant to the performance of the process?
Enter inquiry answer:
 1. HIDE
 2. NOT-HIDE
Number Of Choice: 1

          PRECOND-TERMINATION-AVAILABILITY-Q: Does the lexico-grammar contain
          the facility for expressing action INSTRUCT-ACTION in terminating
          condition form
Enter inquiry answer:
 1. AVAILABLE
 2. NOT-AVAILABLE
Number Of Choice: 2


      In PRECOND-TERMINATION-AVAILABILITY,
Realizing: (MARK ONUS TENSE PRESENT)

      In PRECOND-TERMINATION-AVAILABILITY,
Realizing: (MARK ONUS FORM STATIVE-PASSIVE)

          MENTION-STATUS-Q: Has the action INSTRUCT-ACTION or its subactions
          been mentioned in a procedural sequence earlier in the text?
Enter inquiry answer:
 1. MENTIONED
 2. NOT-MENTIONED
Number Of Choice: 2

          ACTION-LOGICAL-TYPE-Q: Is the action INSTRUCT-ACTION conditional or
          not?  An action is conditional if the reader, when following the
          instructions, may or may not have performed it at the time of
          expression.
Enter inquiry answer:
 1. CONDITIONAL
 2. NOT-CONDITIONAL
Number Of Choice: 2


      In PRECOND-WHEN,
Realizing: (MARK ONUS LINKER 'WHEN)

          PROCEDURAL-ACTION-NODE-Q: Is the action INSTRUCT-ACTION the
          culmination of an installation process which results in the device
          being ready for normal use?
Enter inquiry answer:
```

```
 1. GO-ON-CONDITION-NODE
 2. NO-GO-ON-CONDITION-NODE
Number Of Choice: 2

          ACTION-TYPE-Q: Is REMOVE-ACTION a single project, the root node of
          the text, or a set of primitives?
Enter inquiry answer:
 1. PROJECT
 2. ROOT
 3. PRIMITIVES
Number Of Choice: 1

          READER-KNOWLEDGE-Q: Is REMOVE-ACTION a procedural sequence that the
          reader is assumed to know?
Enter inquiry answer:
 1. KNOWN
 2. UNKNOWN
Number Of Choice: 2

          CONCURRENCY-STRUCTURE-Q: Is REMOVE-ACTION a procedure with
          concurrency that must be expressed?  If concurrency is involved, is
          the node a concurrent parent (that is does the node have concurrent
          children) or a component of a concurrent structure (that is a node
          with with procedural children)?
Enter inquiry answer:
 1. CONCURRENT
 2. CONCURRENT-COMPONENT
 3. NOT-CONCURRENT
Number Of Choice: 3


      In CONCURRENCY-STRUCTURE,
Realizing: (ITERATIVE-INSERT ONUS PROCEDURAL-SUBSPAN)

          CHILDREN-COUNT: How many subactions for REMOVE-ACTION will be
          expressed (0 means stop)?
Enter inquiry answer: 2

          TEXT-ELEMENT-ID: What is the concept which is associated with the
          subordinate or satellite text elements (in order if applicable) of
          REMOVE-ACTION?  This inquiry will be made once for each subaction if
          necessary.
Enter inquiry answer: grasp-action

          TEXT-ELEMENT-ID: What is the concept which is associated with the
          subordinate or satellite text elements (in order if applicable) of
          REMOVE-ACTION?  This inquiry will be made once for each subaction if
          necessary.
Enter inquiry answer: pull-action

          ACTION-ACTOR-Q: Is REMOVE-ACTION being performed by the reader or
          some other agent?  The actor is some other agent only if you know it
          is not performed by the reader.
Enter inquiry answer:
 1. READER
 2. NONREADER
Number Of Choice: 1

          TEMPORAL-ORIENTATION-Q: Is the action REMOVE-ACTION one which was
          performed at a temporally remote time in the past?  If you are not
          sure, then it is NOT temporally remote prior.
Enter inquiry answer:
 1. TEMPORALLY-REMOTE-PRIOR
 2. NOT-TEMPORALLY-REMOTE-PRIOR
Number Of Choice: 2

          PROCEDURAL-OBVIOUSNESS-Q: Is REMOVE-ACTION procedurally obvious or
          not.  An action is procedurally obvious if the current local
```

```
          discourse is about that action and the reader is expected to be
          familiar with it.
Enter inquiry answer:
 1. OBVIOUS
 2. NOT-OBVIOUS
Number Of Choice: 2

          MENTION-STATUS-Q: Has the action REMOVE-ACTION or its subactions
          been mentioned in a procedural sequence earlier in the text?
Enter inquiry answer:
 1. MENTIONED
 2. NOT-MENTIONED
Number Of Choice: 2

          ACTION-LOGICAL-TYPE-Q: Is the action REMOVE-ACTION conditional or
          not? An action is conditional if the reader, when following the
          instructions, may or may not have performed it at the time of
          expression.
Enter inquiry answer:
 1. CONDITIONAL
 2. NOT-CONDITIONAL
Number Of Choice: 2

          PURPOSE-RHETORICAL-CONTEXT-Q: Is REMOVE-ACTION either a nucleus in a
          rhetorical relation or an item in a joint structure?
Enter inquiry answer:
 1. SIMPLE-CONTEXT
 2. COMPLEX-CONTEXT
Number Of Choice: 2


      In PURPOSE-RHETORICAL-CONTEXT,
Realizing: (COPY ONUS SUPERORDINATE-PURPOSE)

      In PURPOSE-RHETORICAL-CONTEXT,
Realizing: (UNLINK SUPERORDINATE-PURPOSE GRAMMATICAL-FEATURE AS-DOMAIN)

      In PURPOSE-RHETORICAL-CONTEXT,
Realizing: (UNLINK SUPERORDINATE-PURPOSE SENTENCE-TRACE BOTH)

      In PURPOSE-RHETORICAL-CONTEXT,
Realizing: (UNLINK SUPERORDINATE-PURPOSE JOINT-SCHEMA AS-DOMAIN)

      In PURPOSE-RHETORICAL-CONTEXT,
Realizing: (UNLINK ONUS SCHEMA AS-RANGE)

      In PURPOSE-RHETORICAL-CONTEXT,
Realizing: (UNLINK ONUS RELATION-SCHEMA AS-DOMAIN)

      In PURPOSE-RHETORICAL-CONTEXT,
Realizing: (STRUCTURE SUPERORDINATE-PURPOSE ONUS PROCEDURAL-SUBSPAN)

          PURPOSE-SCOPE-Q: Is the set of actions that REMOVE-ACTION refers to
          singular or multiple?
Enter inquiry answer:
 1. GLOBAL
 2. LOCAL
Number Of Choice: 1


      In PURPOSE-SCOPE,
Realizing: (COPY ONUS INTERNAL-PURPOSE)

      In PURPOSE-SCOPE,
Realizing: (UNLINK INTERNAL-PURPOSE GRAMMATICAL-FEATURE AS-DOMAIN)

      In PURPOSE-SCOPE,
Realizing: (UNLINK INTERNAL-PURPOSE SENTENCE-TRACE BOTH)
```

```
      In PURPOSE-SCOPE,
Realizing: (UNLINK ONUS SCHEMA BOTH)

      In PURPOSE-SCOPE,
Realizing: (STRUCTURE INTERNAL-PURPOSE ONUS PURPOSE)

          PURPOSE-CONDITIONS-Q: Are there any conditions, either propositional
          or adverbial, to the purpose stated by REMOVE-ACTION?
Enter inquiry answer:
 1. CONDITIONS
 2. NOT-CONDITIONS
Number Of Choice: 1


      In PURPOSE-CONDITIONAL-STATUS,
Realizing: (MARK ONUS FORM IMPERATIVE)

       In FRONT-GLOBAL-BY-PURPOSE,
Realizing: (REORDER ONUS NEXT-NODE CONTINUE-SENTENCE)

          PURPOSE-SENTENCE-COMPLEXITY-Q: Does the sentence formed by stating
          the purpose REMOVE-ACTION and all the related actions constitute
          less that 6 propositions?
Enter inquiry answer:
 1. OK
 2. TOO-COMPLEX
Number Of Choice: 1


      In PURPOSE-SENTENCE-COMPLEXITY,
Realizing: (MARK NEXT-NODE LINKER BY)

      In PURPOSE-SENTENCE-COMPLEXITY,
Realizing: (MARK-CHILDREN NUCLEUS-NODE FORM ING)

      In PURPOSE-SENTENCE-COMPLEXITY,
Realizing: (REORDER ONUS NEXT-NODE CONTINUE-SENTENCE)

          SENTENCE-ENUMERATION-Q: Does the style parameter for this manual
          dictate enumeration of long list of sequential actions or not?
Enter inquiry answer:
 1. ENUMERATE
 2. NOT-ENUMERATE
Number Of Choice: 2


      In SENTENCE-ENUMERATION,
Realizing: (REORDER ONUS NEXT-NODE CONTINUE-SENTENCE)

          ACTION-TYPE-Q: Is GRASP-ACTION a single project, the root node of
          the text, or a set of primitives?
Enter inquiry answer:
 1. PROJECT
 2. ROOT
 3. PRIMITIVES
Number Of Choice: 1

          READER-KNOWLEDGE-Q: Is GRASP-ACTION a procedural sequence that the
          reader is assumed to know?
Enter inquiry answer:
 1. KNOWN
 2. UNKNOWN
Number Of Choice: 1

          NODE-TYPE-Q: Is GRASP-ACTION an action coupled with other concurrent
          actions in a concurrent structure or not?
Enter inquiry answer:
```

```
 1. NOT-CONCURRENT-NODE
 2. CONCURRENT-NODE
Number Of Choice: 1

          ACTION-ACTOR-Q: Is GRASP-ACTION being performed by the reader or
          some other agent? The actor is some other agent only if you know it
          is not performed by the reader.
Enter inquiry answer:
 1. READER
 2. NONREADER
Number Of Choice: 1

          TEMPORAL-ORIENTATION-Q: Is the action GRASP-ACTION one which was
          performed at a temporally remote time in the past? If you are not
          sure, then it is NOT temporally remote prior.
Enter inquiry answer:
 1. TEMPORALLY-REMOTE-PRIOR
 2. NOT-TEMPORALLY-REMOTE-PRIOR
Number Of Choice: 2

          PROCEDURAL-OBVIOUSNESS-Q: Is GRASP-ACTION procedurally obvious or
          not. An action is procedurally obvious if the current local
          discourse is about that action and the reader is expected to be
          familiar with it.
Enter inquiry answer:
 1. OBVIOUS
 2. NOT-OBVIOUS
Number Of Choice: 2

          MENTION-STATUS-Q: Has the action GRASP-ACTION or its subactions been
          mentioned in a procedural sequence earlier in the text?
Enter inquiry answer:
 1. MENTIONED
 2. NOT-MENTIONED
Number Of Choice: 2

          ACTION-LOGICAL-TYPE-Q: Is the action GRASP-ACTION conditional or
          not? An action is conditional if the reader, when following the
          instructions, may or may not have performed it at the time of
          expression.
Enter inquiry answer:
 1. CONDITIONAL
 2. NOT-CONDITIONAL
Number Of Choice: 2

          ADD-IMPERATIVE-Q: Has the action GRASP-ACTION already been marked
          for grammatical form?
Enter inquiry answer:
 1. NO-MARKINGS
 2. MARKINGS
Number Of Choice: 2

          IMPERATIVE-ACTION-TYPE-Q: Is the action GRASP-ACTION repetitive,
          continuous, have a side effect or none of the above?
Enter inquiry answer:
 1. REPETITIVE
 2. CONTINUOUS
 3. ACTION-WITH-SIDE-EFFECT
 4. NOTA
Number Of Choice: 4

          SENTENCE-SEQUENCE-TRAILER-Q: Is the action GRASP-ACTION immediately
          followed by another action with no rhetorical satellites?
Enter inquiry answer:
 1. TRAILER
 2. NOT-TRAILER
Number Of Choice: 1
```

In SENTENCE-SEQUENCE-TRAILER,

```
Realizing: (COMBINE ONUS NEXT-NODE CONTINUE-SENTENCE)

          SENTENCE-UNWANTED-CONCURRENCY-Q: Is there the possibility of the
          reader inferring an unwanted concurrency between action GRASP-ACTION
          and the following action in the sequence?
Enter inquiry answer:
 1. POSSIBLE
 2. NOT-POSSIBLE
Number Of Choice: 2


      In SENTENCE-UNWANTED-CONCURRENCY,
Realizing: (MARK NEXT-NODE LINKER AND)

          SENTENCE-PREVIOUS-LINKER-Q: Is the proposition GRASP-ACTION already
          being linked in the current sequence and if so, is the item part of
          a bare procedural sequence (that is not part of a more complex
          purpose or precondition structure)?
Enter inquiry answer:
 1. LINKED
 2. COUPLE-END
 3. NOT-LINKED
Number Of Choice: 3

          PROCEDURAL-ACTION-NODE-Q: Is the action GRASP-ACTION the culmination
          of an installation process which results in the device being ready
          for normal use?
Enter inquiry answer:
 1. GO-ON-CONDITION-NODE
 2. NO-GO-ON-CONDITION-NODE
Number Of Choice: 2

          ACTION-TYPE-Q: Is PULL-ACTION a single project, the root node of the
          text, or a set of primitives?
Enter inquiry answer:
 1. PROJECT
 2. ROOT
 3. PRIMITIVES
Number Of Choice: 1

          READER-KNOWLEDGE-Q: Is PULL-ACTION a procedural sequence that the
          reader is assumed to know?
Enter inquiry answer:
 1. KNOWN
 2. UNKNOWN
Number Of Choice: 1

          NODE-TYPE-Q: Is PULL-ACTION an action coupled with other concurrent
          actions in a concurrent structure or not?
Enter inquiry answer:
 1. NOT-CONCURRENT-NODE
 2. CONCURRENT-NODE
Number Of Choice: 1

          ACTION-ACTOR-Q: Is PULL-ACTION being performed by the reader or some
          other agent?  The actor is some other agent only if you know it is
          not performed by the reader.
Enter inquiry answer:
 1. READER
 2. NONREADER
Number Of Choice: 1

          TEMPORAL-ORIENTATION-Q: Is the action PULL-ACTION one which was
          performed at a temporally remote time in the past?  If you are not
          sure, then it is NOT temporally remote prior.
```

```
Enter inquiry answer:
 1. TEMPORALLY-REMOTE-PRIOR
 2. NOT-TEMPORALLY-REMOTE-PRIOR
Number Of Choice: 2

          PROCEDURAL-OBVIOUSNESS-Q: Is PULL-ACTION procedurally obvious or
          not.  An action is procedurally obvious if the current local
          discourse is about that action and the reader is expected to be
          familiar with it.
Enter inquiry answer:
 1. OBVIOUS
 2. NOT-OBVIOUS
Number Of Choice: 2

          MENTION-STATUS-Q: Has the action PULL-ACTION or its subactions been
          mentioned in a procedural sequence earlier in the text?
Enter inquiry answer:
 1. MENTIONED
 2. NOT-MENTIONED
Number Of Choice: 2

          ACTION-LOGICAL-TYPE-Q: Is the action PULL-ACTION conditional or not?
          An action is conditional if the reader, when following the
          instructions, may or may not have performed it at the time of
          expression.
Enter inquiry answer:
 1. CONDITIONAL
 2. NOT-CONDITIONAL
Number Of Choice: 2

          ADD-IMPERATIVE-Q: Has the action PULL-ACTION already been marked for
          grammatical form?
Enter inquiry answer:
 1. NO-MARKINGS
 2. MARKINGS
Number Of Choice: 2

          IMPERATIVE-ACTION-TYPE-Q: Is the action PULL-ACTION repetitive,
          continuous, have a side effect or none of the above?
Enter inquiry answer:
 1. REPETITIVE
 2. CONTINUOUS
 3. ACTION-WITH-SIDE-EFFECT
 4. NOTA
Number Of Choice: 4

          SENTENCE-SEQUENCE-TRAILER-Q: Is the action PULL-ACTION immediately
          followed by another action with no rhetorical satellites?
Enter inquiry answer:
 1. TRAILER
 2. NOT-TRAILER
Number Of Choice: 2

          PROCEDURAL-ACTION-NODE-Q: Is the action PULL-ACTION the culmination
          of an installation process which results in the device being ready
          for normal use?
Enter inquiry answer:
 1. GO-ON-CONDITION-NODE
 2. NO-GO-ON-CONDITION-NODE
Number Of Choice: 2

          ACTION-TYPE-Q: Is PLACE-ACTION a single project, the root node of
          the text, or a set of primitives?
Enter inquiry answer:
 1. PROJECT
 2. ROOT
 3. PRIMITIVES
Number Of Choice: 1
```

```
          READER-KNOWLEDGE-Q: Is PLACE-ACTION a procedural sequence that the
          reader is assumed to know?
Enter inquiry answer:
 1. KNOWN
 2. UNKNOWN
Number Of Choice: 2

          CONCURRENCY-STRUCTURE-Q: Is PLACE-ACTION a procedure with
          concurrency that must be expressed? If concurrency is involved, is
          the node a concurrent parent (that is does the node have concurrent
          children) or a component of a concurrent structure (that is a node
          with with procedural children)?
Enter inquiry answer:
 1. CONCURRENT
 2. CONCURRENT-COMPONENT
 3. NOT-CONCURRENT
Number Of Choice: 3


      In CONCURRENCY-STRUCTURE,
Realizing: (ITERATIVE-INSERT ONUS PROCEDURAL-SUBSPAN)

          CHILDREN-COUNT: How many subactions for PLACE-ACTION will be
          expressed (0 means stop)?
Enter inquiry answer: 1

          TEXT-ELEMENT-ID: What is the concept which is associated with the
          subordinate or satellite text elements (in order if applicable) of
          PLACE-ACTION? This inquiry will be made once for each subaction if
          necessary.
Enter inquiry answer: return-action

          ACTION-ACTOR-Q: Is PLACE-ACTION being performed by the reader or
          some other agent? The actor is some other agent only if you know it
          is not performed by the reader.
Enter inquiry answer:
 1. READER
 2. NONREADER
Number Of Choice: 1

          TEMPORAL-ORIENTATION-Q: Is the action PLACE-ACTION one which was
          performed at a temporally remote time in the past? If you are not
          sure, then it is NOT temporally remote prior.
Enter inquiry answer:
 1. TEMPORALLY-REMOTE-PRIOR
 2. NOT-TEMPORALLY-REMOTE-PRIOR
Number Of Choice: 2

          PROCEDURAL-OBVIOUSNESS-Q: Is PLACE-ACTION procedurally obvious or
          not. An action is procedurally obvious if the current local
          discourse is about that action and the reader is expected to be
          familiar with it.
Enter inquiry answer:
 1. OBVIOUS
 2. NOT-OBVIOUS
Number Of Choice: 2

          MENTION-STATUS-Q: Has the action PLACE-ACTION or its subactions been
          mentioned in a procedural sequence earlier in the text?
Enter inquiry answer:
 1. MENTIONED
 2. NOT-MENTIONED
Number Of Choice: 2

          ACTION-LOGICAL-TYPE-Q: Is the action PLACE-ACTION conditional or
          not? An action is conditional if the reader, when following the
          instructions, may or may not have performed it at the time of
```

```
          expression.
Enter inquiry answer:
 1. CONDITIONAL
 2. NOT-CONDITIONAL
Number Of Choice: 2

          PURPOSE-RHETORICAL-CONTEXT-Q: Is PLACE-ACTION either a nucleus in a
          rhetorical relation or an item in a joint structure?
Enter inquiry answer:
 1. SIMPLE-CONTEXT
 2. COMPLEX-CONTEXT
Number Of Choice: 2


      In PURPOSE-RHETORICAL-CONTEXT,
Realizing: (COPY ONUS SUPERORDINATE-PURPOSE)

      In PURPOSE-RHETORICAL-CONTEXT,
Realizing: (UNLINK SUPERORDINATE-PURPOSE GRAMMATICAL-FEATURE AS-DOMAIN)

      In PURPOSE-RHETORICAL-CONTEXT,
Realizing: (UNLINK SUPERORDINATE-PURPOSE SENTENCE-TRACE BOTH)

      In PURPOSE-RHETORICAL-CONTEXT,
Realizing: (UNLINK SUPERORDINATE-PURPOSE JOINT-SCHEMA AS-DOMAIN)

      In PURPOSE-RHETORICAL-CONTEXT,
Realizing: (UNLINK ONUS SCHEMA AS-RANGE)

      In PURPOSE-RHETORICAL-CONTEXT,
Realizing: (UNLINK ONUS RELATION-SCHEMA AS-DOMAIN)

      In PURPOSE-RHETORICAL-CONTEXT,
Realizing: (STRUCTURE SUPERORDINATE-PURPOSE ONUS PROCEDURAL-SUBSPAN)

          PURPOSE-SCOPE-Q: Is the set of actions that PLACE-ACTION refers to
          singular or multiple?
Enter inquiry answer:
 1. GLOBAL
 2. LOCAL
Number Of Choice: 2


      In PURPOSE-SCOPE,
Realizing: (STRUCTURE PARENT-NODE CHILD-NODE PARENT-LINK)

      In PURPOSE-SCOPE,
Realizing: (STRUCTURE CHILD-NODE ONUS PURPOSE)

      In PURPOSE-SCOPE,
Realizing: (UNLINK ONUS JOINT-SCHEMA BOTH)

          PURPOSE-OPTIONALITY-Q: Is the purpose stated by PLACE-ACTION an
          action with is optional or required at this point in the
          instructions?
Enter inquiry answer:
 1. OPTIONAL
 2. REQUIRED
Number Of Choice: 2

          PURPOSE-CONTRASTIVENESS-Q: Is the purpose stated by PLACE-ACTION
          expressing a contrast with purposes just expressed in the text?
Enter inquiry answer:
 1. CONTRASTIVE
 2. NOT-CONTRASTIVE
Number Of Choice: 2
```

```
      In PURPOSE-CONTRASTIVENESS,
Realizing: (REORDER NUCLEUS-NODE ONUS CONTINUE-SENTENCE)

          PURPOSE-CONDITIONS-Q: Are there any conditions, either propositional
          or adverbial, to the purpose stated by PLACE-ACTION?
Enter inquiry answer:
 1. CONDITIONS
 2. NOT-CONDITIONS
Number Of Choice: 2

          PURPOSE-AGENT-STATUS-Q: Is the agent of the action PLACE-ACTION a
          non-reader entity that can be ascribed the human quality of
          volitionality?
Enter inquiry answer:
 1. VOLITIONAL-NON-READER
 2. NOT-VOLITIONAL-NON-READER
Number Of Choice: 2

          PURPOSE-GOAL-STATUS-Q: Does the goal of the action PLACE-ACTION have
          more rhetorical importance that the action itself?  That is, is the
          action a generic sort of action whose most important part is its
          goal?
Enter inquiry answer:
 1. METONYMY
 2. NOT-METONYMY
Number Of Choice: 2

          PURPOSE-NOMINAL-AVAILABILITY-Q: Is there a nominalization available
          for the action PLACE-ACTION?
Enter inquiry answer:
 1. AVAILABLE
 2. NOT-AVAILABLE
Number Of Choice: 1

          PURPOSE-NOMINAL-ARGUMENTS-Q: Does the expression of the action
          PLACE-ACTION require more than one propositional argument?
Enter inquiry answer:
 1. OK
 2. TOO-MANY
Number Of Choice: 1

          PURPOSE-NOMINAL-ARGUMENT-COMPLEXITY-Q: Is the single propositional
          argument to action PLACE-ACTION itself complex (a proposition)?
Enter inquiry answer:
 1. SIMPLEX
 2. COMPLEX
Number Of Choice: 2


      In PURPOSE-NOMINAL-ARGUMENT-COMPLEXITY,
Realizing: (MARK ONUS FORM TNF)

          ACTION-TYPE-Q: Is RETURN-ACTION a single project, the root node of
          the text, or a set of primitives?
Enter inquiry answer:
 1. PROJECT
 2. ROOT
 3. PRIMITIVES
Number Of Choice: 1

          READER-KNOWLEDGE-Q: Is RETURN-ACTION a procedural sequence that the
          reader is assumed to know?
Enter inquiry answer:
 1. KNOWN
 2. UNKNOWN
Number Of Choice: 1

          NODE-TYPE-Q: Is RETURN-ACTION an action coupled with other
```

```
          concurrent actions in a concurrent structure or not?
Enter inquiry answer:
 1. NOT-CONCURRENT-NODE
 2. CONCURRENT-NODE
Number Of Choice: 1

          ACTION-ACTOR-Q: Is RETURN-ACTION being performed by the reader or
          some other agent?  The actor is some other agent only if you know it
          is not performed by the reader.
Enter inquiry answer:
 1. READER
 2. NONREADER
Number Of Choice: 1

          TEMPORAL-ORIENTATION-Q: Is the action RETURN-ACTION one which was
          performed at a temporally remote time in the past?  If you are not
          sure, then it is NOT temporally remote prior.
Enter inquiry answer:
 1. TEMPORALLY-REMOTE-PRIOR
 2. NOT-TEMPORALLY-REMOTE-PRIOR
Number Of Choice: 2

          PROCEDURAL-OBVIOUSNESS-Q: Is RETURN-ACTION procedurally obvious or
          not.  An action is procedurally obvious if the current local
          discourse is about that action and the reader is expected to be
          familiar with it.
Enter inquiry answer:
 1. OBVIOUS
 2. NOT-OBVIOUS
Number Of Choice: 2

          MENTION-STATUS-Q: Has the action RETURN-ACTION or its subactions
          been mentioned in a procedural sequence earlier in the text?
Enter inquiry answer:
 1. MENTIONED
 2. NOT-MENTIONED
Number Of Choice: 2

          ACTION-LOGICAL-TYPE-Q: Is the action RETURN-ACTION conditional or
          not?  An action is conditional if the reader, when following the
          instructions, may or may not have performed it at the time of
          expression.
Enter inquiry answer:
 1. CONDITIONAL
 2. NOT-CONDITIONAL
Number Of Choice: 2

          ADD-IMPERATIVE-Q: Has the action RETURN-ACTION already been marked
          for grammatical form?
Enter inquiry answer:
 1. NO-MARKINGS
 2. MARKINGS
Number Of Choice: 1


      In ADD-IMPERATIVE,
Realizing: (MARK ONUS FORM IMPERATIVE)

          IMPERATIVE-ACTION-TYPE-Q: Is the action RETURN-ACTION repetitive,
          continuous, have a side effect or none of the above?
Enter inquiry answer:
 1. REPETITIVE
 2. CONTINUOUS
 3. ACTION-WITH-SIDE-EFFECT
 4. NOTA
Number Of Choice: 4

          SENTENCE-SEQUENCE-TRAILER-Q: Is the action RETURN-ACTION immediately
```

```
          followed by another action with no rhetorical satellites?
Enter inquiry answer:
 1. TRAILER
 2. NOT-TRAILER
Number Of Choice: 2

          PROCEDURAL-ACTION-NODE-Q: Is the action RETURN-ACTION the
          culmination of an installation process which results in the device
          being ready for normal use?
Enter inquiry answer:
 1. GO-ON-CONDITION-NODE
 2. NO-GO-ON-CONDITION-NODE
Number Of Choice: 2

SPL command:
(S114 / PENMAN::CONCURRENT
      :DOMAIN (D114 / PENMAN-KB:DISPOSITIVE-MATERIAL-ACTION
          :LEX REMOVE
          :ACTOR (PENMAN::HEARER / PENMAN-KB:PERSON
                  :LEX PENMAN::HEARER)
          :ACTEE (A116 / PENMAN-KB:DECOMPOSABLE-OBJECT
             :LEX IT:PHONE
             :DETERMINER THE)
          :SPEECHACT PENMAN::IMPERATIVE
          :ENABLEMENT (E131 / PENMAN::CONJUNCTION
                  :EVENT-Q PENMAN::COMPLEX-EVENT
                  :DOMAIN (D131 / PENMAN-KB:DISPOSITIVE-MATERIAL-ACTION
                      :LEX IT:GRASP
                      :ACTOR (PENMAN::HEARER / PENMAN-KB:PERSON
                              :LEX PENMAN::HEARER)
                      :ACTEE (A132 / PENMAN-KB:DECOMPOSABLE-OBJECT
                         :LEX IT:TOP
                         :DETERMINER THE
                         :PART-OF (P148 / PENMAN-KB:DECOMPOSABLE-OBJECT
                              :LEX IT:HANDSET
                              :DETERMINER THE))
                      :CONTRASTIVE-EXTENSION-Q PENMAN::SIMPLE-EXTENDING
                      :TYPICAL-PARTICIPANT-Q PENMAN::NOTTYPICALCLASS)
                  :RANGE (R154 / PENMAN-KB:DISPOSITIVE-MATERIAL-ACTION
                     :LEX IT:PULL
                     :ACTOR (PENMAN::HEARER / PENMAN-KB:PERSON
                             :LEX PENMAN::HEARER)
                     :ACTEE P148
                     :CONTRASTIVE-EXTENSION-Q PENMAN::SIMPLE-EXTENDING
                     :TYPICAL-PARTICIPANT-Q PENMAN::NOTTYPICALCLASS)))
      :RANGE (R114 / PENMAN-KB:DISPOSITIVE-MATERIAL-ACTION
         :LEX IT:INSTRUCT
         :ACTEE (PENMAN::HEARER / PENMAN-KB:PERSON
                 :LEX PENMAN::HEARER)
         :TENSE PENMAN::PRESENT)
      :THEME R114)

SPL command:
(S184 / PENMAN::RST-PURPOSE
      :DOMAIN (D184 / PENMAN-KB:NONDIRECTED-ACTION
          :LEX IT:RETURN-MOVEMENT
          :ACTOR (PENMAN::HEARER / PENMAN-KB:PERSON
                  :LEX PENMAN::HEARER)
          :SPEECHACT PENMAN::IMPERATIVE
          :DESTINATION (D195 / PENMAN-KB:DECOMPOSABLE-OBJECT
                   :LEX IT:SEAT))
      :RANGE (R184 / PENMAN-KB:DISPOSITIVE-MATERIAL-ACTION
         :LEX IT:PLACE-CALL
         :ACTOR (PENMAN::HEARER / PENMAN-KB:PERSON
                 :LEX PENMAN::HEARER)
         :ACTEE (A202 / PENMAN-KB:DECOMPOSABLE-OBJECT
            :LEX IT:CALL-NOUN))
      :THEME D184)
```

```
"
Imagene text output:

   When you are instructed, remove the phone by grasping the top of
the handset and pulling it.  Return to a seat to place a call.

"
?
```

C.3 The Output TRL Commands

Here is a listing of the Text Representation commands that were automatically constructed during the course of executing the IMAGENE system networks.

```
;;; -*- Mode: LISP; Syntax: Common-Lisp; Package: IG; Base: 10. -*-

(in-package "IG")
(let ((savedKnowledgeBase *knowledge-base*)
      (savedWorld *world*))
  (change-kb 'TRL:CURRENT-TRL-KB)

  ;;; Knowledge Base:   TRL:CURRENT-TRL-KB
  ;;; Last Saved On:    03/16/93 13:13:05

(TELL (:ABOUT PULL-ACTION TRL:SPAN (TRL:LINKER 'AND) (TRL:FORM 'TRL::ING)
       (TRL:NEW-SENTENCE RETURN-ACTION)))

(TELL (:ABOUT REMOVE-ACTION TRL:SPAN (TRL:CONTINUE-SENTENCE GRASP-ACTION)
       (TRL:FORM 'TRL::IMPERATIVE)))

(TELL (:ABOUT SUPERORDINATE-PURPOSE156 TRL:SPAN
       (TRL:PROCEDURAL-SUBSPAN INTERNAL-PURPOSE156)
       (TRL:PRECONDITION INSTRUCT-ACTION)))

(TELL (:ABOUT INTERNAL-PRECOND154 TRL:SPAN
       (TRL:PROCEDURAL-SUBSPAN SUPERORDINATE-PURPOSE156)))

(TELL (:ABOUT RETURN-ACTION TRL:SPAN (TRL:FORM 'TRL::IMPERATIVE)
       (TRL:CONTINUE-SENTENCE PLACE-ACTION) (TRL:PURPOSE PLACE-ACTION)))

(TELL (:ABOUT PLACE-ACTION TRL:SPAN (TRL:FORM 'TRL::TNF)))

(TELL (:ABOUT INSTRUCT-ACTION TRL:SPAN (TRL:LINKER 'WHEN)
       (TRL:FORM 'TRL::STATIVE-PASSIVE) (TRL:TENSE 'TRL::PRESENT)
       (TRL:CONTINUE-SENTENCE REMOVE-ACTION)))

(TELL (:ABOUT IG-TEXT TRL:SPAN (TRL:NEW-SENTENCE INSTRUCT-ACTION)
       (TRL:PROCEDURAL-SUBSPAN SUPERORDINATE-PURPOSE157)
       (TRL:PROCEDURAL-SUBSPAN INTERNAL-PRECOND154)))

(TELL (:ABOUT GRASP-ACTION TRL:SPAN (TRL:CONTINUE-SENTENCE PULL-ACTION)
       (TRL:FORM 'TRL::ING) (TRL:LINKER 'BY)))

(TELL (:ABOUT INTERNAL-PURPOSE156 TRL:SPAN (TRL:PURPOSE REMOVE-ACTION)
       (TRL:PROCEDURAL-SUBSPAN GRASP-ACTION)
       (TRL:PROCEDURAL-SUBSPAN PULL-ACTION)))

(TELL (:ABOUT SUPERORDINATE-PURPOSE157 TRL:SPAN
       (TRL:PROCEDURAL-SUBSPAN RETURN-ACTION)))

  (change-kb savedKnowledgeBase)
  (when savedWorld (change-world savedWorld)) )
```

C.4 Discussion

This appendix has given the full output of a session with IMAGENE, as well as a number of its associated input and output command files. A note is appropriate here concerning the large number of inquiries that must be answered in order to produce the Remove-Phone text, a relatively small passage with six action expressions. As was demonstrated in Chapter 4, and discussed in detail in Chapters 5 and 6, IMAGENE is capable of producing a large number of different forms for any one action, on the order of 55 different lexical and grammatical forms. The system network is designed to distinguish the precise communicative context in which each one is appropriate, and must be run once for each action to be expressed. This leads to the large number of inquiries listed in this appendix. A similar phenomenon occurs in Penman's own system network for generating sentences.

Appendix D

Features for Instructional Text

This appendix contains a presentation of each of the functional distinctions that proved useful in the analysis. The presentation includes the name of system that implemented the distinction, a description of the distinction itself, and a specification of the lexical and grammatical consequences of its features. It also includes a list of the possible feature choices for each of the distinctions. The distinctions are grouped by the general function which they perform, namely rhetorical status selection and grammatical form selection. The grammatical form selection distinctions are further divided into the groups based on the rhetorical status which they pertain to, namely purpose, precondition, result, sequence, and concurrent.

In general, this list of distinctions follows the list of systems implemented in the IMAGENE system network. There are, however, two exceptions to this. First, Penman-style system networks include both systems and gates. Systems make relevant functional distinctions that affect the lexical and grammatical form of the output. Gates are simple and-gates or or-gates which make no functional distinctions. Because of this, only systems are included in this list. Secondly, for implementational reasons, some of the systems themselves do not make functional distinctions, but rather perform more procedural tasks. For example, one of the systems determines if a particular action node has already been marked grammatically. Different actions are taken based on this implementational distinction. Because this is not a functional distinction, this sort of system is not included in this list either.[1]

It should also be noted that the system names were used in chapters 5 and 6 were sometimes shortened for inclusion in the diagrams listed there. The system name identifiers are listed in full in this appendix.

D.1 Features for Rhetorical Status Selection

The following features are included specifically for performing rhetorical status selection, although the distinctions may also affect the grammatical form selections.

Action-Type

Distinction: This is the root system of the entire network, and distinguishes between actions which have procedural children and those with have primitive children. Further, the root node of the network is a special case. Currently, on the actions which have procedural children are addressed by IMAGENE.

Feature Names: Procedural-Sequence Root-Node Primitive-Sequence

Reader-Knowledge

Distinction: This system distinguishes between those actions which the readers are expected to know and those which they aren't. Known procedures tend to result in simple imperative commands, while

[1] The reason these non-functional systems exist at all is that the processing required to address some of these procedural issues is available only at the system level in Penman.

unknown procedures require sub-actions to be expressed. These latter actions are called generator actions and tend to be expressed as purposes.

Feature Names: Known-Procedure Unknown-Procedure

Concurrency-Structure

Distinction: This system distinguishes between those action nodes which have concurrent children and those which do not. It also distinguishes concurrent procedure-components, which are nodes in a concurrent structure, which themselves have procedural children. Given these distinctions, IMAGENE is capable of representing and expressing simple sequential actions, simple concurrent execution of two simple actions, and the concurrent execution of an action with a simple sequence of other actions.

Feature Names: Concurrent-Procedure Concurrent-Procedure-Component Not-Concurrent-Procedure

Node-Type

Distinction: As the **Concurrency-Structure** distinction had to do with the children of a particular node, this system addresses the concurrency status of the node itself. Concurrent nodes lead to expressions of concurrency.

Feature Names: Not-Concurrent-Node Concurrent-Node

Procedural-Node-Type

Distinction: This system determines if the current communicative context is appropriate for a Go-on-condition expression. Such expressions are used to delineate between procedural installation instructions and primitive use instructions.

Feature Names: No-Go-On-Condition-Node Go-On-Condition-Node

Action-Actor

Distinction: This system distinguishes between reader and non-reader actions. Reader actions tend to become purposes or sequential commands, while non-reader actions tend to become preconditions and results.

Feature Names: Reader-Action Nonreader-Action

Action-Monitor-Type

Distinction: This system distinguishes between non-reader actions which the reader is expected to monitor and those which are not. Monitor-result actions become preconditions, while non-monitor-result actions become results.

Feature Names: Action-Monitor-Result Not-Action-Monitor-Result

Temporal-Orientation

Distinction: This system distinguishes between those actions which were performs at a temporally remote time in the past and those which are not. Temporally remote actions are typically expressed as preconditions.

Feature Names: Temporally-Remote-Prior Not-Temporally-Remote-Prior

Procedural-Obviousness

Distinction: This system distinguishes between those actions which are considered procedurally obvious to the reader and those which are not. This is a different distinction than is made in the **Reader-Knowledge** system. An action can be procedurally non-obvious and still not require sub-actions to be

expressed. Chapter 5 contains a detailed discussion of actions with are procedurally obvious.
Feature Names: Procedurally-Obvious Not-Procedurally-Obvious

Mention-Status

Distinction: This system distinguishes between those actions which have been prescribed to the reader earlier in the text, either as a purpose or a sequential command. Such actions tend to be expressed as preconditions.
Feature Names: Repeated-Mention Not-Repeated-Mention

Action-Logical-Type

Distinction: This system distinguishes between actions which will always be performed at the current point in the execution of the process and those which are conditional upon some state. Conditional actions tend to become preconditions.
Feature Names: Conditional-Action Not-Conditional-Action

Condition-Probability

Distinction: This system distinguishes between those conditional actions which have a high probability of having been performed and those which do not. High probability preconditions are expressed with "when" and low probability actions with "if".
Feature Names: Condition-Probable Not-Condition-Probable

D.2 Features for Grammatical Form Selection

The following sections present the features that are relevant for the grammatical form selection for each of the procedural relations that were addressed in the current study, purpose, precondition, result, concurrent, and sequence.

D.2.1 Features for Purpose Expressions

The following features are included specifically to deal with grammatical form selection for purpose expressions.

Purpose-Scope

Distinction: This system distinguishes between global and local purpose actions. Global purposes tend to be expressed as fronted "to" infinitives, while local purposes are not fronted (and not in "to" infinitive form).
Feature Names: Purpose-Global Purpose-Local

Purpose-Optionality

Distinction: This system distinguishes between purposes which are options and those which are required. Optional purposes are fronted, required purpose may not be.
Feature Names: Purpose-Optional Purpose-Required

Purpose-Contrastiveness

Distinction: This system distinguishes between contrastive and non-contrastive purposes. Purposes which are expressed in contrast to the preceding material in the text are fronted. Non-contrastive purposes may not be.

Feature Names: Purpose-Contrastive Not-Purpose-Contrastive

Purpose-Conditional-Status

Distinction: This system identifies those purposes which have either explicit, propositional preconditions or restriction conditions, usually stated as arguments or obliques. Such conditions motivate the imperative "by" purpose form.

Feature Names: Purpose-Conditions Not-Purpose-Conditions

Purpose-Sentence-Complexity

Distinction: This system distinguishes those imperative "by" purpose forms that have more than six propositions from those which do not. The former forms are split using the adjoined purpose form.

Feature Names: Not-Purpose-Sentence-Too-Complex Purpose-Sentence-Too-Complex

Purpose-Agent-Status

Distinction: This system distinguishes between those purposes which are to be performed by some non-reader agent that is acting volitionally (even if they are an inanimate) and those which are not. Volitional non-readers lead to the "so that" purpose form.

Feature Names: Purpose-Volitional-Non-Reader Not-Purpose-Volitional-Non-Reader

Purpose-Goal-Status

Distinction: This system identifies those situations where the object of the process (that is the *goal* of the process) is more important that the process itself. This typically happens in contrastive lists. This situation leads to the goal-metonymy purpose form.

Feature Names: Purpose-Goal-Metonymy Not-Purpose-Goal-Metonymy

Purpose-Nominal-Availability

Distinction: This system distinguishes between those actions which have a nominalized lexical item in English and those which do not. If the nominalization exists, it is typically used.

Feature Names: Purpose-Nominal-Available Not-Purpose-Nominal-Available

Purpose-Nominal-Arguments

Distinction: This system distinguishes between those purpose propositions which have more than one argument, and those which do not. Purpose with more than one argument tend not to be expressed in nominalized form in instructional text.

Feature Names: Purpose-Too-Many-Arguments Not-Purpose-Too-Many-Arguments

Purpose-Nominal-Argument-Complexity

Distinction: This system distinguishes between purpose argument that are propositional in nature and those that are not. The nominalized form tends not to be used when there is a propositional argument, even if there is only one. Simplex purpose expressions at this point are expressed with the "for" nominal purpose form.

Feature Names: Purpose-Nominal-Argument-Simplex Purpose-Nominal-Argument-Complex

Purpose-Tnf-Arguments-Status

Distinction: This system distinguishes between those "to" infinitive expressions that require the undesirable restatement of argument already mentioned in the sentence so far. This situation motivates the use of the "for" participial purpose form.

Feature Names: Purpose-Tnf-Arguments-Ok Not-Purpose-Tnf-Arguments-Ok

D.2.2 Features for Precondition Expressions

The following features are included specifically to deal with grammatical form selection for precondition expressions.

Precond-Coordination-Status

Distinction: This system distinguishes between those preconditions which are part of a coordinate precondition structure and those which are not. The coordinate structures give rise "and" conjunctions of precondition expressions.

Feature Names: Precond-Coordinate Not-Precond-Coordinate

Precond-Act-Topic

Distinction: This system distinguishes between preconditions which pertain to actions that concern the current local topic of the text. Such expressions are typically placed before their precondition expressions.

Feature Names: Precond-Act-Local-Topic Not-Precond-Act-Local-Topic

Precond-Changeable-Type

Distinction: This system identifies situations where the precondition is something that the reader is capable of and expected to fix if the condition is not correct. Such situations require the "make sure" precondition form.

Feature Names: Precond-Changeable Not-Precond-Changeable

Precond-Complexity

Distinction: This system distinguishes between those logical preconditions which involve more than one proposition and those which do not. This is closely related to the distinction made by **Precond-Coordination-Status**, but this distinction is used exclusively to address the choice of linker. Simplex conditions lead to either an "only-if" or an "if" expression, while complex conditions lead to either a "whether-or-not" or an "if" expression.

Feature Names: Precond-Simplex Precond-Complex

Precond-Exclusivity

Distinction: This system distinguishes between those simplex preconditions that involve exclusive conditions, expressed with the "only-if" form, and those which are not, expressed with the "if" form.

Feature Names: Precond-Exclusive Not-Precond-Exclusive

Precond-Alternativeness

Distinction: This system distinguishes between those complex preconditions that involve sets of alternatives, expressed with the "whether-or-not"form, and those which do not, expressed with the "if"

form.
Feature Names: Precond-Alternatives Not-Precond-Alternatives

Precond-Previous-Act-Type

Distinction: This system distinguishes between preconditions which are based on a number of different types of actions: (1) Monitor actions, leading to present tense imperative precondition expressions; (2) Giving actions, leading to either present tense having expressions or "is-required" expressions; (3) Habitual decisions, leading to present tense action expressions; (4) Placing actions, leading to locative expressions; and (5) Other types of effective actions, leading to a number of different expressional forms.

Feature Names: Precond-Monitor-Act Precond-Giving-Act Precond-Placing-Act Precond-Habitual-Decision Precond-Effective-Act

Precond-Act-Hide

Distinction: This system distinguishes between those precondition actions that should be hidden if possible in the instructions. Such actions include overly complex actions and those which have excessively long durations.
Feature Names: Precond-Hide-Act Not-Precond-Hide-Act

Precond-Active-Availability

Distinction: This system distinguishes between those actions which have an active form available in the English lexico-grammar. Those which have the form will used it.
Feature Names: Precond-Active-Available Not-Precond-Active-Available

Precond-Inception-Status

Distinction: This system determines if the readers could have witnessed the inception of the process on which the precondition is being based. If so, a present tense sensing action is expressed. If not, either the relational or the passive forms are used.
Feature Names: Precond-Inception-Experienced Not-Precond-Inception-Experienced

Precond-Termination-Availability

Distinction: This system determines if a relational form of the action on which the precondition is being based exists or not. If it does it is used, otherwise the present tense passive form is used.
Feature Names: Precond-Termination-Available Not-Precond-Termination-Available

Precond-Nonchangeable-Act-Nominal-Availability

Distinction: This system determines there is nominalization available in the lexico-grammar for expression the precondition. If there is, it tends to be used. If there isn't, the participial form is used.
Feature Names: Precond-Nonchangeable-Act-Nominal-Available
Not-Precond-Nonchangeable-Act-Nominal-Available

Precond-Nonchangeable-Act-Nominal-Arguments-Status

Distinction: This system distinguishes between those nominal precondition expressions which involve the restatement of material from the same sentence and those which do not.
Feature Names: Precond-Nonchangeable-Act-Nominal-Arguments-Ok
Not-Precond-Nonchangeable-Act-Nominal-Arguments-Ok