INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each original is also photographed in one exposure and is included in reduced form at the back of the book.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.
Compiling pragmatic knowledge for conversational interactions

Becker, Barbara D., Ph.D.
The Ohio State University, 1994

Copyright ©1994 by Becker, Barbara D. All rights reserved.
COMPILING PRAGMATIC KNOWLEDGE FOR CONVERSATIONAL INTERACTIONS

DISSERTATION
Presented in Partial Fulfillment of the Requirements for
the Degree Doctor of Philosophy in the Graduate
School of The Ohio State University

By
Barbara D. Becker, B.A., M.S., M.S.

*****
The Ohio State University

1994

Dissertation Committee:
J. B. Pollack
M. L. Geis
B. Chandrasekaran

Approved by
Adviser
Department of Computer and Information Sciences
Copyright by
Barbara D. Becker
1994
For Leonard
ACKNOWLEDGEMENTS

This work could never have been completed without the support and guidance of my dissertation committee. My committee’s comments have improved this document tremendously. Without advice and encouragement from Jordan Pollack I would not have been able to complete this work. Mike Geis’s theory has been a cornerstone for a lot of the design of the ANITA and ALICE systems; discussions with him always help me clarify my sometimes confused thoughts. I’d like to thank Chandra, for his very helpful comments and also for providing an excellent environment in which to do research.

I would also like to thank Terry Patten for his guidance over many years. His SLANG system is used in both ANITA and ALICE. He was also instrumental in the design of ANITA’s pragmatic stratum, his contributions to it were invaluable. While the planner for ANITA was implemented by me, Terry Patten and Mike Geis contributed many helpful ideas for its design, and some sections of chapter four originally appeared in Patten, Geis and Becker (1992). This work was supported in part by a grant from the Center for Cognitive Science at The Ohio State University.

I’d also like to thank all the graduate students and staff who have helped me over years. In particular, I’d like to thank: Greg Saunders for occasionally letting me use the computer and letting me prattle on about whatever (or whoever) was annoying me on any given day; John Kolen, for knowing more about completing a dissertation and the computer systems in the department than is imaginable; Julie Hartigan and Jim Skon for discussions on systemic grammar, pragmatics, and OPS5 among other things; Deborah
Shands, for having the best critiques of my work for someone who claims to know nothing about computational linguistics; Ed Large, for sharing the misery of the last two months; and everyone else from the LAIR including: Madhu Soundararajan, Pete Angeline, Richard Fox, Arun Welch and everyone else.

I’d like to thank my parents for encouraging me for many long years. They always managed to keep up the appearance that they believed I would eventually finish, which could not have been such any easy task. Thanks to my siblings for helping me hone my debating skills. Thanks to Aunt Lorraine, Uncle Al, and the rest of the Henchecks for making me part of the family while I’ve been in Ohio.

I’d like to thank Lenny for his constant support and love. There are simply no words to describe how much he has done to keep my spirits up when all life’s little problems seemed overwhelming.

Lastly, I’d like to thank my dog, Caddy. She kept me safe from attacks by squirrels while I was distracted by my work; learned to hide dog cookies, so she wouldn’t starve when I was too busy to remember to feed her; and taught me that there are no problems too important that there isn’t time for a good game of fetch.
VITA

September 18, 1961 ......................... Born - Poughkeepsie, NY


1985-1986 .......................................... Research Assistant, State University of New York at Binghamton

1986 .................................................. M.S., Advanced Technology, The State University of New York at Binghamton, Binghamton, NY

1987-1989 .......................................... Teaching Associate, The Ohio State University, Columbus, OH

1988 .................................................. M.S., Computer Science, The Ohio State University, Columbus, OH

1989-1992 .......................................... Research Associate, The Ohio State University, Columbus OH

1992-1994 .......................................... Teaching Associate, The Ohio State University, Columbus OH

PUBLICATIONS

FIELDS OF STUDY

Major Field: Computer and Information Sciences

Studies in:

<table>
<thead>
<tr>
<th>Field</th>
<th>Professor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Artificial Intelligence</td>
<td>Prof. Terry Patten</td>
</tr>
<tr>
<td>Language and Culture</td>
<td>Prof. Keith Walters</td>
</tr>
<tr>
<td>Parallel Computing</td>
<td>Prof. P. Sadayappan</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

DEDICATION .......................................................................................................................... ii  
ACKNOWLEDGEMENTS ....................................................................................................... iii  
VITA ........................................................................................................................................... v  
LIST OF TABLES ................................................................................................................... xi  
LIST OF FIGURES ................................................................................................................ xii  

<table>
<thead>
<tr>
<th>CHAPTER</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>I. INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Introduction</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Syntax, Semantics, and Pragmatics</td>
<td>2</td>
</tr>
<tr>
<td>1.3 Language Variation</td>
<td>3</td>
</tr>
<tr>
<td>1.3.1 Dialect Variation</td>
<td>4</td>
</tr>
<tr>
<td>1.3.2 Style Variation</td>
<td>5</td>
</tr>
<tr>
<td>1.3.3 Politeness Variation</td>
<td>5</td>
</tr>
<tr>
<td>1.4 Register Variation</td>
<td>6</td>
</tr>
<tr>
<td>1.5 Contextual effects on language Variation</td>
<td>6</td>
</tr>
<tr>
<td>1.5.1 Mode</td>
<td>7</td>
</tr>
<tr>
<td>1.5.2 Field</td>
<td>8</td>
</tr>
<tr>
<td>1.5.3 Tenor</td>
<td>9</td>
</tr>
<tr>
<td>1.6 Using language variation</td>
<td>10</td>
</tr>
<tr>
<td>1.7 Sublanguages</td>
<td>11</td>
</tr>
<tr>
<td>1.8 Contributions</td>
<td>12</td>
</tr>
<tr>
<td>1.9 Organization</td>
<td>13</td>
</tr>
<tr>
<td>II. THEORIES OF CONVERSATION</td>
<td>15</td>
</tr>
<tr>
<td>2.1 Introduction</td>
<td>15</td>
</tr>
<tr>
<td>2.2 Grice</td>
<td>15</td>
</tr>
<tr>
<td>2.3 Speech Act Theory</td>
<td>18</td>
</tr>
<tr>
<td>2.3.1 Austin</td>
<td>18</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>4.5.2 Implementing Register Knowledge</td>
<td>57</td>
</tr>
<tr>
<td>4.5.3 ANITA's Grammatical and Lexical Knowledge</td>
<td>61</td>
</tr>
<tr>
<td>4.5.4 An Example</td>
<td>62</td>
</tr>
<tr>
<td>4.5.4.1 Planning the Example</td>
<td>62</td>
</tr>
<tr>
<td>4.5.4.2 Exploiting Register in the Example</td>
<td>64</td>
</tr>
<tr>
<td>4.5.4.3 Grammatical and Lexical Processing in the Example</td>
<td>65</td>
</tr>
<tr>
<td>4.6 Conclusions</td>
<td>67</td>
</tr>
<tr>
<td>V. A GENERALIZED SYSTEM FOR SOCIAL INTERACTIONS</td>
<td>68</td>
</tr>
<tr>
<td>5.1 Introduction</td>
<td>68</td>
</tr>
<tr>
<td>5.2 ALICE's Architecture</td>
<td>68</td>
</tr>
<tr>
<td>5.3 ALICE's Pragmatic Abducer</td>
<td>69</td>
</tr>
<tr>
<td>5.4 Pragmatic Stratum</td>
<td>73</td>
</tr>
<tr>
<td>5.4.1 Style features</td>
<td>74</td>
</tr>
<tr>
<td>5.4.2 Politeness features</td>
<td>75</td>
</tr>
<tr>
<td>5.4.3 Gist Features</td>
<td>75</td>
</tr>
<tr>
<td>5.4.4 An Example of Pragmatic Features</td>
<td>76</td>
</tr>
<tr>
<td>5.5 ALICE's Planning Module</td>
<td>77</td>
</tr>
<tr>
<td>5.5.1 The Relationship Matrix</td>
<td>78</td>
</tr>
<tr>
<td>5.5.2 Communicative Social Interaction Structures</td>
<td>80</td>
</tr>
<tr>
<td>5.6 Generating an Utterance</td>
<td>81</td>
</tr>
<tr>
<td>5.6.1 Selecting style features</td>
<td>81</td>
</tr>
<tr>
<td>5.6.2 Selecting Gist Features</td>
<td>83</td>
</tr>
<tr>
<td>5.6.3 Selecting Politeness Features</td>
<td>84</td>
</tr>
<tr>
<td>5.7 Understanding an utterance</td>
<td>84</td>
</tr>
<tr>
<td>5.7.1 Determining Politeness</td>
<td>85</td>
</tr>
<tr>
<td>5.7.2 Determining Style</td>
<td>85</td>
</tr>
<tr>
<td>5.7.3 Determining Gist</td>
<td>86</td>
</tr>
<tr>
<td>5.7.4 Determining felicity conditions</td>
<td>86</td>
</tr>
<tr>
<td>5.7.5 Determining the Social Interaction</td>
<td>86</td>
</tr>
<tr>
<td>5.7.5.1 A General Method for Determining Social Interaction</td>
<td>87</td>
</tr>
<tr>
<td>5.7.5.2 Determining the Social Interaction in ALICE</td>
<td>90</td>
</tr>
<tr>
<td>5.8 Examples of ALICE's Planning Module</td>
<td>91</td>
</tr>
<tr>
<td>5.8.1 Example 1 - “Do you have hot chocolate?”</td>
<td>92</td>
</tr>
<tr>
<td>5.8.1.1 Generating the Example</td>
<td>92</td>
</tr>
<tr>
<td>5.8.1.2 Understanding the Example</td>
<td>94</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Input to Pragmatic Understander for “Can you leave earlier?”</td>
<td>72</td>
</tr>
<tr>
<td>2. Relationship Matrix for ALICE</td>
<td>79</td>
</tr>
<tr>
<td>3. Pragmatic features for three sample utterances</td>
<td>92</td>
</tr>
</tbody>
</table>
## LIST OF FIGURES

<table>
<thead>
<tr>
<th>FIGURE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Allen's Processing of &quot;how much is a ticket for the train to Rochester?&quot;</td>
<td>33</td>
</tr>
<tr>
<td>2. Two request sequences</td>
<td>37</td>
</tr>
<tr>
<td>3. Compiling a chain of inferences. (A) contains no compilation. (B), (C), and (D) contain compiled links which bypass steps in the inference chain.</td>
<td>41</td>
</tr>
<tr>
<td>4. Compiled Knowledge in the SLANG System</td>
<td>44</td>
</tr>
<tr>
<td>5. A sample conversation for an airline booking</td>
<td>47</td>
</tr>
<tr>
<td>6. A Portion of the Anita's Pragmatic Stratum</td>
<td>50</td>
</tr>
<tr>
<td>7. S-structure for flight-request</td>
<td>52</td>
</tr>
<tr>
<td>8. A simplified logicogrammar fragment (extracted form Mann et al., 1981)</td>
<td>59</td>
</tr>
<tr>
<td>9. Generating &quot;Can you leave earlier&quot; in the ANITA system</td>
<td>66</td>
</tr>
<tr>
<td>10. ALICE's processing</td>
<td>70</td>
</tr>
<tr>
<td>11. Example of a CSI-structure, Commercial-Thing-Request Structure</td>
<td>82</td>
</tr>
<tr>
<td>12. Pragmatic Features for &quot;do you have hot chocolate?&quot;</td>
<td>83</td>
</tr>
<tr>
<td>13. Felicity conditions for three types of social interactions</td>
<td>88</td>
</tr>
<tr>
<td>14. Pragmatic Features for &quot;I would like hot chocolate.&quot;</td>
<td>95</td>
</tr>
</tbody>
</table>
CHAPTER I
INTRODUCTION

1.1 Introduction
There are two significant facts about human use of language. First, everyone, regardless of intelligence, has the ability to engage in productive conversations in everyday life (except for cases of severe brain damage). Second, this use of language is quick, seemingly instantaneous. People do not spend an enormous amount of time planning their contributions in a conversation. Theories of conversation on the other hand, generally have lots of rules about how to decide what to say and how to say it. This leads to an obvious question of how to reconcile the enormous amount of planning that appears to go into conversation with the rapidity of the result.

This work attempts to show how the amount of processing needed to generate and understand natural language utterances in conversations can be significantly reduced by the use of two types of specialized information. The first is a complex mapping between contextual elements and linguistic form and the second is a model of conversation which depends heavily on these contextual elements. These two types of information are pragmatic in nature. They contain information about how we use language in order to help facilitate cooperative activities between conversational participants.
1.2 Syntax, Semantics, and Pragmatics

When an utterance is made in a conversation it generally must go through three levels of analysis in order to be generated or understood by a computer system. These three levels correspond to the syntactic, semantic and pragmatic analysis. (In spoken language phonology and morphology could also be added to this list. Since the scope of this work is limited to keyboard and screen interfaces, morphology and phonology will not be discussed here.) The syntactic level maps the string of words into a parse tree, all words are classified by part of speech and then combined into phrases and clauses. The semantic level of analysis assigns roles to the various parts of the parse tree. This basically involves determining the type of the main verb (action, mental, etc.) and assigning roles (agent, beneficiary, etc.) to the noun phrases and prepositional phrases of the sentences based on the verb type. For example in the sentence: “Mary hit John”, Mary has the role of Actor and John has the role of Theme. The last level of analysis is the pragmatic level, which is the level which my work addresses.

The first step in discussing compiling pragmatic knowledge is to define what is meant by “pragmatics”. In the pragmatic level the emphasis is put on determining the role an utterance performs and the effects it has on the hearers. At this level the analysis expands beyond looking at the “meaning” of the pieces of the utterance to determining what are the implications of what was said, why did the speaker say it, and what does that say about the speaker’s plans, goals and beliefs. The most important aspect of pragmatics is the context of an utterance. The syntax of a sentence can be discussed without knowing who wrote it or when or where they wrote it. Likewise the basic semantics of a sentence remain the same regardless of who said it. There are exceptions to this, such as sentences using definite
reference or indexicals. In “the dog bit the mailman” the actual reference for “the dog” and “the mailman” depends upon the context. But even without a context, anyone can create a mental picture to describe the sentence “the dog bit the mailman”.

The pragmatics of an utterance on the other hand depends solely on the situation in which the utterance occurs. The utterance “can you open the window?” has a multitude of possible functions, depending on the participants and the activity in progress when the utterance is produced in the conversation. If the speaker and hearer are both in an un-air-conditioned office in the summer and the hearer is sitting by the window, the utterance may function as a request that the hearer actually open the window. If the speaker is a prospective tenant in an office building and the hearer is the landlord, the utterance may be merely an information question, designed to determine whether the windows in the room are capable of opening at all. If the speaker of the utterance is a physical therapist and the hearer is a patient, the utterance may serve either or both purposes: as an information question and/or a request for a verification that the patient can perform the action.

It is possible with a little imagination to think up several uses for any utterance. The interaction between the utterance and the context to produce a particular meaning is central to the work presented here.

1.3 Language Variation

Given that pragmatics involves the interaction of language and context, what does that mean to conversations?

At the simplest level it means that people will choose to say different things in different ways depending who they are, where they are, and what they are trying to do. For
example, a person at a dinner table could communicate the desire for the salt in a number of ways including:

1) Pass the salt.
2) Could I have the salt?
3) I need the salt
4) Give me the salt.
5) Would you mind passing the salt?

The speaker’s selection of one of the above to depends on who they are, who they are with and how formal the dinner is. For example, a king could choose option 4 while having dinner with a group of subjects, but a small child at thanksgiving dinner could not without resulting in correction from adults. The king has the right to give orders because of his social position. Of course there are even some situations where it would be inappropriate to select option 4. If he where at a dinner with other heads of state he would probably be considered rude for demanding the salt and would have to fall back on an utterance more like option 5.

Geis (forthcoming) has identified four types of variation that can occur: dialect, style, politeness, and register.

1.3.1 Dialect Variation

Dialect variation refers to variation based on geographic location or social group. Examples of dialect variation include the difference in pronunciation found between speakers in Texas and New England; or use of different words in different locations. In New York a person can offer a soft drink by saying “do you want a soda?” in the midwest the acceptable term would be “pop”.

1.3.2 Style Variation

Style variation refers to instances where the literal meaning of utterances stays the same but the formality changes. Geis claims this is a reflection of the variation due to social distance and power relationships of the people engaged in the conversation. Examples of this type of variation can be seen in utterances (6) through (9). While there has been no change in the literal meaning or even the form of the utterance, (7) is obviously a more formal choice than (6) or (9).

6) Wanna beer?
7) Do you want a beer?
8) How about a beer?
9) Ya want a beer?

1.3.3 Politeness Variation

Unlike style variation, utterances which shown politeness variation do not have the same literal meaning. Utterances (10) through (13) show politeness variation:

10) I’ll have a chocolate doughnut.
11) Give me a chocolate doughnut!
12) Could I get a chocolate doughnut?
13) I’d like a chocolate doughnut.

These utterances differ in their syntactic form, all three of the major sentence type are represented. They also differ in their main verb. Utterances (10) and (13) are similar in form but their main verb is different. This difference is important to the meaning since (13) implies merely a preference, making it appear more polite. The similarity between these utterances is in the significance or role in a conversation, which Geis terms s-meaning.
1.4 Register Variation

The fourth type of language variation Geis identifies is register knowledge. Like politeness, register variation is often a variation of literal meaning, or l-meaning as opposed to s-meaning.

The difference between politeness and register variation lies in the cause of this variation. In politeness the cause of the variation is due to the efforts of participants to do facework (Goffman, 1967). In register variation the difference is caused by the current activity and by the social role which the speaker chooses to use.

For example, in the case of a travel agent register, an agent can say to a customer, “I can put you in Boston by noon.” If it were two friends talking it would they would not be able to use the verb “put” in this sense. A friend will instead have to say “I can get you to Boston by noon” or “I can drive you to the airport.”, but friends do not “put” friends in places.

1.5 Contextual effects on language Variation

Many researchers in the sociolinguistics have noted that syntactic and semantic structures appear with varying frequency in different instances of language. Halliday(1978) proposes that context could be divided into three distinct dimensions each of which effects the type of language used. These dimensions are: mode, field and tenor. In this section I will use these distinctions as a starting point for looking at how this language variation can be used in conversation.
1.5.1 Mode

There is a wealth of research which shows that the language that people use varies tremendously depending on the medium of expression. A simple example of how mode effects language is in telegrams. Telegrams have a language all their own, regardless of the topic being communicated or the relationship between sender and receiver. In telegrams words cost money. This leads to a very simple syntax, which generates messages such as: “Uncle Fred dead. Funeral Monday.” In telegram talk only content words appear, function words such as “the” or “is” are elided.

Another example of language variation as a function of mode is in the distinction between spoken and written language. Many researchers have studied the differences between written and spoken text (Biber, 1986; Chafe and Tannen, 1987; Halliday, 1989). While the interpretation of the results varies from researcher to researcher, there are patterns which appear again repeatedly. Perhaps one of the most thorough analyses of these patterns was done by Biber. Biber claimed the problem with much of the other work was that it was limited to a small number of texts and various researchers looked at different criteria for comparing speech and written text.

Biber separated texts into more than just spoken or written. He defined three dimensions on which texts could be ranked: Interactive vs. edited, abstract vs. situated, and reported vs. immediate style. He then showed two things. First, that each of these dimensions represented a different set of syntactic features. Second, that given a collection of texts they would be ordered differently depending on the dimension chosen.

The importance of these dimensions is that they reflect the function of the particular text more than the mode of communication. What Biber shows is that the important factors
in the type of language used is not whether it is written or spoken but why it is being used. It just happens that spoken text is often used for purposes that are more interactive, situated and immediate than the uses for written text. This means that the mode of the text is not a simple two state function, but rather a complex interaction of several different factors.

In terms of Geis’s types of variation discussed earlier, the most frequent type of variation reflected in mode is style variation. Face to face conversations are usually less formal than phone conversations, which are in turn usually less formal than written forms of communication.

1.5.2 Field

The field of a text refers to the activity in which the participants are engaged or the activity they are discussing. Several researchers have pointed out the differences in language used in different fields such as radio DJs (Montgomery, 1986), medicine (Cicourel, 1981), sports announcing (Ferguson, 1983), law (O’Barr, 1981), etc.

The most obvious difference in language related to field involves the use of lexical items. In medicine it is normal to see terms such as “laceration, pulmonary system, etc.” but it would be very strange to see these terms in sports announcing. Exceptions to this will occur in situations where the domains overlap. For example, after a fight in a hockey game the announcer might use the word “laceration” to describe an injury. Even in these cases however, the terminology will be at a less technical level than the would be in a discussion between doctors. An announcer is more likely to say that a player has “suffered a heart attack” than that player “is suffering from a myocardial infarction”.
Although lexical differences between fields is often the most striking difference, there are also syntactic and semantic differences. For example O’Barr noted that in jury instructions, there are a large number of left branching sentences (sentences in which there are complex constructions before the verb). Left branching sentences are uncommon in most other areas. Semantic difference can be seen in the use of different senses of words in different fields. When a radio DJ uses the term “hit”, the term refers to a popular record or song, whereas when a baseball announcer uses “hit”, the term refers to the action of connecting a ball with a bat.

The most common type of variation related to field is register variation. Politeness and formality may also be affected though since particular activities may constrain the roles which the participants choose, and those roles in turn effect the social distance and power balance.

1.5.3 Tenor

The tenor of a conversation refers to the relationship between the participants. Participants who are close to each other socially (friends, lovers, etc.) are able to speak to each other in ways strangers can not. It is acceptable for a wife to say to her husband “bring me a beer” when he is heading to the kitchen, but it would be considered rude to say that to a stranger at a party, unless he or she had already made some kind of an offer.

The tenor of a conversation also reflects the relative social status of the two people engaged in the conversation. A teacher can say to a student “you must turn in your exam” but a student can not tell a teacher “you must grade my exam by Wednesday”. In fact, in the
case where one person enjoys a higher status than the other, the person of lower status can rarely use imperatives at all.

Examples of work which has found changes in language based on social relationships includes: Labov (1970) which looks at both phonetic and syntactic differences in language based on the participants; and Circourel (1981), which noted that the relationship between doctors and patients often causes them to frame statements in ways which result in misunderstandings.

Variation in the tenor of the context often reflects variations in the style and politeness of the utterances. People can be less formal when there is less social distance between the participants. The relationship between the participants affects the politeness since the relationship defines both the distance and the power balance.

1.6 Using language variation

While the concept of language variation is fairly well accepted in the linguistic community, there is no real consensus about how this knowledge can be used in the generation or understanding of utterances. A possible use of this type of knowledge is suggested by the work of Halliday (1978). Halliday proposes that the combination of field, tenor, and mode features of a particular situation in which language occurs determines the ranges of possible meanings and structures which can be used in the situation. In other words these features define the subpart of the language of the participants which can be used in that particular situation.

In his work, Halliday provides what he call a “semantic stratum”. The particular semantic stratum designed by Halliday represents the possible language functions available
to a two year old child. In this representation, Halliday separates the possible meanings that
the child can generate from the syntax rules he can use to generate a sentence. The
relationship between the semantics and the syntax is represented by links in the semantic
stratum. Particular functions entail particular utterances. Of course there is a wide gap in
the number and complexity of issues that influence language of a two year old child and the
language of an adult. In this work I will use a set up similar to Halliday's semantic stratum
to show how contextual features can affect language use.

1.7 Sublanguages

Some researchers in computer science have used this knowledge of language variation as
support for the use of sub-languages (Sager, 1982; Hirschman and Sager, 1982). In these
systems a semantic grammar is used for the grammar. Semantic grammars use a subset of
the syntax of English that is commonly found in a particular domain. Semantic grammars
also use grammatical categories that include information about the types of activities which
are engaged in that particular domain. The syntax is combined with these particularized
grammatical categories to create a set of grammatical rules which describes the possible
utterances within that domain.

There are several drawbacks to the sublanguage method of text generation. First, it
does not look at the contextual features responsible for the different types of syntactic
structures which are found. Therefore it may miss regularities that are not immediately
apparent in the data, and find others that are just coincidental. Second, since it does not look
at the functions behind the patterns found there is no way to carry information from one
domain to another. Whenever a system needs to be developed for a new domain the system
developers must start from scratch developing a new sublangauge for the second domain.
These two problems mean that the systems which are built in this manner will eventually turn out to be brittle. They will work fine under the limited domain for which they were built, but will fail as soon as the context switches, even slightly.

1.8 Contributions

In this work I will show how combining pragmatic knowledge with a theory of conversation as social interaction and some standard language planning mechanisms can be used to help build a conversation machine which uses a wide range of uncanned utterances.

The pragmatic knowledge includes a pragmatic stratum which is a hierarchical representation of context whose organization is suggested by the work of Halliday. This hierarchical representation is divided into separate sections representing field, tenor, and mode. The separation of parts allows for a representation that would enable system designers to save large sections of the stratum while moving form one domain to another.

The organization of conversations is based on a theory of language as social interactions developed by Geis (1989a). I have combined these two into a planning mechanism for a conversation machine which, given a certain goal and contextual situation can plan an appropriate utterance for the part of the conversation. I will also discuss how that same knowledge can be used to allow an language understanding system to fully interpret an utterance in a conversation.
1.9 Organization

In this section I will lay out the organization of the dissertation, which consists of six chapters.

Chapter 2 provides the linguistic background for the work. In this chapter I will discuss several theories of conversation. I will also look at the type of knowledge and processing that conversational participants might use by exploring the current opinion within the field of linguistics. In this chapter I will explain Geis's theory of social interaction which is used as a basis for the systems discussed in chapters 4 and 5.

Chapter 3 provides the computational background for the work. In it I will discuss other approaches to building natural language systems. In this chapter I will also discuss work by Patten on compiling information in natural language generation.

Chapter 4 presents the ANITA system. ANITA is a natural language generation system for the Travel agent register. This chapter explains some of pragmatics behind generating a simple conversation and also discusses how this part of the system interacts with the semantic and semantic/syntactic components.

Chapter 5 discusses the extensions made to Anita's knowledge structure in order to use this knowledge for natural language understanding. In order to do this, the extended system now functions within the more general context of commercial encounters, rather than being limited to travel agents. This extended system, called ALICE, is a bidirectional pragmatic component for natural language processing. Chapter 5 explains the processing the system goes through in the course of interpreting and producing utterances in a
conversation. This chapter also includes some examples of how ALICE processes utterances.

Chapter 6 summarizes the work presented in chapters 2 through 5 and then presents conclusions of this work and shows possible future extensions to the research.
CHAPTER II

THEORIES OF CONVERSATION

2.1 Introduction

Since conversation is linguistic in nature there is a natural tendency to look at many of the features of conversation and consider them part of the knowledge of language a person has at their disposal. This can be problematic since conversation is not a single user activity, it is a multi-agent activity. Often the features of conversation which are caused by the cooperative nature of conversation are perceived as necessary to conversation, when in fact they are merely necessary to cooperative activities. In order to understand what is really occurring in a conversation it is important to look the issues involved in engaging in a cooperative conversation.

The notion of conversation as a cooperative activity is certainly not new. Grice (1957) and Austin (1962) both mentioned the importance of understanding language within the context of a multi-agent activity.

2.2 Grice

Grice identified language as cooperation when he put forth the cooperative principle and a set of maxims which follow from the cooperative principle. The cooperative principle states: “make your contribution such as is required, at the stage at which it occurs, by the
accepted purpose or direction of the talk exchange in which you are engaged” (Levinson, 1983, pg. 101). He proposed that in order for both parties to follow the cooperative principle, they must both adhere to the following maxims:

1. Maxim of Quantity: Provide as much information as is needed in a context, but not more information.
2. Maxim of Quality: Speak true information and have evidence for what you are saying.
3. Maxim of Relation: Make your contribution relevant to the context in which you are speaking.
4. Maxim of Manner: Speak as clearly as possible, avoid ambiguity, say things as simply as possible.

In order for both participants to understand what has been said, and to believe that their message has been understood, each must believe that the other is aware of and generally adhering to these rules. There are instances when one or more of the participants in a conversation may choose to flaunt these rules, for instance when telling a lie or making a joke. Knowing when and why these rules are broken is itself a significant part of producing or understanding the utterance.

While Grice’s rules are reasonable, they are too general and inexplicit to provide an effective model for determining what to how or what to utter in a conversation. Given a particular situation, they do not provide any usable information about how to plan an appropriate utterance for that situation. These rules can not be credibly used by people to analyze and interpret every utterance they hear. It would take far too long apply each of these maxims to an utterance to derive its implications. To interpret every utterance, the hearer would have to apply each one of these maxims to the utterance and develop all implications of the utterance for each maxim.
They are even less useful from the standpoint of generation. For instance, if you are in a store and you would like to know if they have a particular brand of toothpaste, Grice's rules do not provide enough guidance to produce a sentence. The maxim of quantity says that you should say enough but not too much. How do you know what is enough? Does the clerk need to know what size tube the customer wants? Does he need to know if the customer has any cavities? The maxim of quality restricts the customer to only saying what is true. Unfortunately at any given time there are an enormous number of things that are true. The maxim of relevance says make the contribution relevant to the context, but without a strong theory of context how do you know what is relevant and What does it mean to be relevant to a context? Sperber and Wilson (1986) present a theory of relevance which links relevance to its “effects” within a context. These effects consist of implications which can be drawn by combining new (a new utterance) and old information (old utterances and beliefs, and their implications). Their theory relies heavily on prepositional logic which does not provide rapid enough reasoning for conversations. Lastly the maxim of manner says to be clear, avoid ambiguity, and make it simple. But all three of these terms are themselves ambiguous. What is clear and unambiguous depends on the situation and the participants of the conversation.

What is needed is some mechanism which links the goals of the speakers with the actual syntactic and semantic structures of an utterance. A clue towards providing this mapping is provided by Austin.
2.3 Speech Act Theory

2.3.1 Austin

Austin (1962) was one of the first to note that there is more to an utterance than just its logical form. You also have to take into account its “force” or significance within a particular context. This force corresponds to the role the utterance performs in the context in which it is used. This multi-faceted approach to meaning can provide a method for connecting syntax, semantics and pragmatics. While the literal meaning is closely related to the syntax of the utterance, the force of the utterance is closely tied to the reason for its use in a particular context. Austin pointed out that there were a number of utterances which he called performatives which by the very act a speaking them caused changes in the world. Examples of these are:

14) I now pronounce you man and wife.
15) I declare war on France.
16) I bequeath you all my worldly possessions.

As soon as a preacher utters (14) the couple he or she is speaking to is married. Likewise in the case of a declaration of war, as soon as the statement is uttered the war is in fact in progress. Austin also noted that there are particular conditions which are essential to the use of these utterances in performing actions. For example, although anyone can say “I declare war on France”, it only has the effect of causing a war if the person uttering it is the head of a country and he or she says it publicly. If a person off the street was to say “I declare war on France” nothing would happen, and if the queen of England, mutters the phrase to herself when no one else can hear it there is no war. Austin termed these conditions on the use of these utterances “felicity conditions”. Felicity conditions are conditions which must be true in order for the illocutionary act to occur.
Exploring these types of utterances led Austin to propose three kinds of acts that are performed simultaneously while speaking: locutionary acts, illocutionary acts, and perlocutionary acts. The locutionary act refers to the uttering of a sentence that has meaning and reference. The illocutionary act refers to the force associated with an utterance, for example a promise, a threat or a request. The perlocutionary act refers to the effects the utterance has on the hearers when it is uttered.

2.3.2 Searle

The notion of the illocutionary force of an utterance was developed further by Searle (1969). Searle proposed a more systematic approach which further developed the notion of felicity conditions. According to Searle, the illocutionary force of promising, for example, has the following felicity conditions (from Levinson, 1983):

- The speaker said he will perform a future action
- The speaker intends to do it
- The speaker believes he can do it
- The speaker thinks he wouldn’t do it anyway, in the normal course of action
- The speaker thinks the addressee wants him to do it
- The speaker intends to place himself under an obligation to do it by uttering U
- Both the speaker and addressee comprehend U
- They are both conscious, normal human beings
- They are both in normal circumstances (not acting in a play)
- The utterance U contains some illocutionary force indicating device which is only properly uttered if all the appropriate conditions obtain.
The first six of these conditions are specific to promises; the last four hold for all speech acts. By setting up a set of felicity conditions, Searle hoped to determine a taxonomy of actions based on these conditions. Using this method, he divided illocutionary forces into five basic groups: representatives, directives, commissives, expressives and declarations. Representatives are actions which commit the speaker to the truth of the expressed proposition, such as assertions and conclusions. Directives are attempts by the speaker to get the addressee to do something as in requests, and questions. Commissives are actions which commit the speaker to some future course of action, as in promises or threats. Expressives are actions which express a psychological state, as in thanking or apologizing. Declarations are actions which effect immediate changes in the institutional state of affairs and which tend to rely on extra-linguistic institutions, for example declaring war or christening.

He then classified felicity conditions into four categories: preparatory conditions, sincerity conditions, propositional content conditions and essential conditions. Preparatory conditions define what is implied by the speech act. Searle states (Searle, 1969, pg. 65):

To put it generally, in the performance of any illocutionary act, the speaker implies that the preparatory conditions of the act are satisfied. Thus, for example, when I make a statement I imply that I can back it up, when I make a promise, I imply that the thing promised is in the hearer’s interest.

Sincerity conditions state what is expressed in the act. In the case of a promise this is and intention. In other illocutionary acts it could be a belief or desire, or an expression of gratitude. Propositional content conditions specify restriction on the action or object involved in the illocutionary act. For example, the propositional content for promises is: Future Act A of Sp (where Sp is the speaker), this condition restricts A to be an action,
which has not already occurred. The essential condition expresses the goal of the act. In a request the goal is to get the hearer to perform and action, in the case of a promise the goal is to get the hearer to believe that the speaker will perform an action.

The main drawback to Searle's work from the standpoint of a computational theory is that he only looks at single utterances. While it is not hard to see questions and assertions as actions pertaining to a particular utterance, many of the other examples mentioned are not necessarily single utterance actions. Take promises for example. Certainly, a single utterance can constitute a promise. For example, a teacher may walk into a classroom and say:

17) I promise to have the midterms graded by tomorrow.

in this case there is no previous discussion, the teacher is willingly entering into an obligation towards the students without being asked; but more often promises do not come out of the blue. They are a response to some expressed need by the other party, for example (a instructor is talking to a grader for a course):

18) Teacher: I need to hand back these exams as soon as possible.
19) Grader: I promise to have the midterms graded by tomorrow.

So the question now becomes: if (19) is a promise, what is (18)? According to Searle's definitions it would probably be an assertion, but it isn't clear. This points out a major flaw in Searle's work. It does not provide explicit criterion for determining which speech act applies to a particular utterance. But that analysis somehow misses the fact that these two utterances together form one single transaction.
2.4 Adjacency Pairs

An adjacency pair (Schegloff and Sacks, 1973) refers to two turns in a discourse each from a different speaker which form a single action. For example, the following dialogue consists of an adjacency pair. The first utterance is a yes/no question which opens the adjacency pair. The second utterance is an answer to the question. This answer closes the adjacency pair.

20) Jerry: Are you busy friday?
21) Chris: No.

Schegloff and Sacks argue that conversations consist of series of adjacencies pairs like the one in (20) - (21). The adjacencies pairs create a structure for the conversation and provide participants with useful information about what to say or expect next in a conversation. If a speaker utters a question, he or she can expect to hear an answer. If a speaker utters an offer, he or she can expect an acknowledgment. Adjacency pair theory also provides for “insertion sequences”, which are cases where a question is followed by another question. In these cases a second adjacency pair is nested inside of the first one. An example of this is seen in (22)- (25).

22) Peg: Do you want to come over for dinner friday?
23) Rob: Could we make it saturday instead?
24) Peg: Sure.
25) Rob: Ok.

I argue that looking at conversation as a series of pairs of this sort does not assist us in accounts of either utterance generation or utterance understanding and therefore merely adds extra processing to the system. What is important about this pairing of utterance in conversation is not that it occurs, but why it occurs. In both of these conversations, the
utterances occur in pairs because these are negotiations of Searle's felicity conditions on speech acts. It is necessary that both participants in the conversation acknowledge that the condition is valid in order for the act to be properly conducted.

Another problem with adjacency pair theory is that there are many conversations in which the attempt to organize all the utterances into adjacency pairs becomes difficult. For example in the following conversation (Jacobs and Jackson, 1983, pg 299) it is not clear what adjacency pairs utterances (31) and (32) are in.

26) C: Hey, Debbie.
27) Are you free from 1:30 to 2:30?
28) D: Yeah, I think so.
29) Do you want me to watch him?
30) C: Yeah.
31) D: I'd love to.
32) It'd be a pleasure
33) C: Okay.
34) Thanks.
35) I'll bring him around then.

While (27) and (28) could form one pair, and (29) and (30) could together form an insertion sequence, it's not clear what to do with (31) and (32). If (31) is in a pair with (27) then (28) is left out. In either case (32) doesn't go with anything. One possible answer is that not all utterances must be in pairs, but if they don't how does a person know which should or shouldn't be in pairs. Regardless it turns out that the notion of adjacency pairs does not help us in the processing of the utterance, and in the next chapter will discuss a planner that works just fine while ignoring the adjacency pair issue completely.
2.5 Language as a social system

One of the most thorough explorations of language as a cultural system can be found in the work of Halliday (1978). Halliday views language from a functional viewpoint. He approaches it as a task of determining what are the functions being performed by the group. He then expects the language being used to be a direct attempt to communicate these functions to others in the group. This implies that there will be a natural tendency in language to try make it easy for the hearer to distinguish which of the functions is being communicated at that moment.

We would expect that in language evolution there would be this constant battle to maintain a balance between complexity and simplicity. The simpler the language the easier it is to understand the meaning in each communication. If a group only needed to communicate two different messages, then the language needed to encode these messages could be simple and there would be little if any miscommunication. As the number of functions the members of the group wish to communicate increases the number of types of messages increases and the encoding method becomes more complicated and the distinctions between any two messages becomes much finer. This in turn leads to a more complex decoding mechanism which could induce more errors, since a small error in decoding could lead to a different interpretation.

One method around this complexity problem is to organize the information to be processed into hierarchies. Hierarchies can be searched quickly. Choices made at the top level of search can eliminate large section of the data information from consideration. Halliday employs this methodology in the design of his “systemic grammar”. In systemic grammar, the syntax of language is arranged hierarchically by function. Halliday organizes
his grammar into separate systems such as: the mood of the clause, the type of main verb for the clause, the type of subject, etc. Each of these systems represents a section of the hierarchy which can be searched independently. Choices in all the sections combine to form a clause.

Halliday uses these hierarchies for all levels of language: phonology, morphology, syntax, and semantics. Choices in any level constrain the choices made at lower levels. This allows a clause to be generated in layers. First, select the semantic function which are to be communicated (this will constrain at least some of the syntactic, morphological and phonological choices). Next, fill in the syntactic choices not determined by the semantic section. Third fill in any morphological choices. Lastly fill in the phonological choices.

Halliday’s systemic grammar has been used as the basis for several natural language systems (Mann and Mathiessen, 1983; Winograd, 1972; Davey, 1979; McKeown, 1982; McDonald, 1980; Patten, 1988).

2.6 Defects in Speech Act Theory
An alternative to Searle’s approach to speech acts is offered by Geis (forthcoming). Geis’s work is built on three particular criticisms of Searlian speech act theory. First, illocutionary force does not necessarily have to be associated with an utterance. Second, some primary acts such as requests often require multiple turns in a conversation in order to be completed. Lastly, these activities are social in nature, not linguistic, therefore there is no reason to associate them with particular linguistic forms as Searle does.
2.6.1 Validity of Illocutionary Force

While at first glance it may appear obvious that an utterance like "Can you pass the salt?" is a request. Speech act theorists have not produced any evidence that it must be seen as a request. Geis points to work by Good (mss.) which argues that there is no psychological evidence that "Can you pass the salt" needs to be mapped into the act type "request" in order to be properly understood. It also could be construed as an attempt to get the hearer to pass the salt or as expressing the need for salt and a desire that the hearer provide it. These alternative interpretations involve what Geis terms the "transactional effect" and the "initial state condition", respectively of interactions. There may be no added benefit to interpreting that question as a request.

2.6.2 Multi-turn Primary Acts

As noted previously sometimes Searle's speech acts require that some previous communication to have occurred in order to be felicitous. Promises rarely come without some prior expression of need by the promisee. Threats often follow statement which provokes them threatener. Invitations frequently require several turns to negotiate, as in (36) - (41). In this conversation the invitation can not be completed until A and B are able to establish B's ability to go.

36) A: Are you doing anything Saturday Night?
37) B: No, why?
38) A: I have a couple tickets to the ball game. Would you like to go?
39) B: What time does it start?
40) A: 8:30.
41) B: I get off work at 7. I'd love to go. Thanks.

As Geis points out the only difference between single and multi turn interactions concern how many of the felicity conditions are satisfied before the conversation starts.
Therefore single turn interactions are merely special cases of multturn interactions. Those utterances still function in the same capacity as they do in multi-turn conversations, which is merely to establish one or more felicity conditions. The primary act itself is not attached to any one condition but to all of them.

2.6.3 Non linguistic Primary Acts

The third point Geis makes is that not only is there no reason to associate primary acts with single utterances, there is no reason to associate them with any utterances at all. Requests, promises, threats etc. are really social interactions. They involve attempts by participants to effect the thoughts or behaviors of others.

These same types of effects can be performed without ever uttering a single word. A kidnapper can make a threat merely by pointing a gun and motioning to a car, thereby communicating “Get in or I'll shoot”. While a receptionist is on the phone, he or she can still offer chairs to people walking in the door with a wave of his or her hand. Even my dog makes requests for walks (or perhaps these are invitations) by dropping a leash at my feet. In essence these actions are not linguistic in nature at all, they are social. Therefore by concentrating on single utterances and the syntactic forms which correspond with particular acts, speech act theory is missing out on the most important aspects of these interactions.

2.7 Communicative Social Interaction Theory

These points have led Geis to develop what he calls a theory of “social interaction”. In this theory, the primary acts used in speech act theory are recast as social interactions. These social interactions involve negotiations of felicity conditions by the participants which generally results in multi-turn sequences of utterances. They also provide a general frame
for classes of social interactions. These interactions are adapted into more specific forms to be used in situations that routinely occur in human interactions.

2.7.1 Components of Communicative Social Interactions

Geis's communicative social interactions are associated with communicative social interaction structures which specify the components of the interaction. This structure consists of four types of components: the initial state condition, the transactional effect of the interaction, the interactional effects, the satisfaction conditions and the domain specifications.

The initial state condition specifies the necessary state for invoking the interaction. This is often equivalent to Searle's sincerity condition. The transactional effect defines the effect that this interaction will have on the participants. This is similar to Searle's essential condition. The interactional effects correspond to politeness effects which the interaction may produce. The satisfaction conditions correspond to Searle's preparatory conditions and usually refer to willingness and ability obstacles which must be overcome in the course of the interaction. Domain specifications specify the type of acceptable parameters for the action. These domain specifications are usually not available at the abstract level of request or offer. Geis's theory involves further refinement into more specific subclasses of interactions such as a ride-request or a commercial-service-request where there is a limit to the acceptable actions or objects to which the interaction can refer.

2.8 Another approach to conversational sequences

Geis is not the only one to notice that some primary acts require multi-turn sequences. Another approach to the solution has been posed by the field of conversation analysis, this solution adds the notion of presequences for primary acts. Presequences are sequences of
turns in conversation which occur before an actual action. Examples of presequences include pre-closings, pre-invitations, and pre-requests. An example of a pre-invitation is (Atkinson and Drew 1979):

42) A: Whatcha doin’?
43) B: Nothin’
44) A: Wanna drink?

Where (44) is considered to be an invitation and (42) is a pre-invitation. The role of the pre-invitation is to flag what is going to occur later. While this makes sense from a post-conversation analysis it offers no useful information during the processing. There is nothing significant about the utterance “Watcha doin’?” which signals it as the precursor to an invitation. It could just as easily be the precursor to a request as in:

45) A: Whatcha doin’?
46) B: Nothin’
47) A: Could ya give me a ride home?

In this sequence this theory would claim that (45) is a pre-request since it signals the future occurrence of a request in (47).

This approach misses one very clear similarity between the use of (42) and (45). In both conversations the utterance is addressing the felicity condition: “hearer is able to do x”. Both an invitation and a request would be impossible if the hearer was occupied with some other activity.

So we are left with the sequences of utterances that are involved in performing an action. Clearly, the pre-sequence analysis approach to solving the problem appears to ineffective. An alternative solution is that these illocutionary forces which Searle associates
with single utterances should really be associated with sequences of utterances. There is no such utterance as a request, but there are sequences of utterances which can be seen as requests. It just happens that in some cases those sequences are only one utterance long, but just because they can be performed in one utterance does not mean that they are a different sort of action than a multi-turn action. The real difference between the two is that in the case of multi-turn sequences, some of the felicity conditions on the action have not yet been completed. In the single turn action all of the felicity conditions are somehow accounted for by context in which the utterance occurs.

**2.9 Summary**

In this chapter, I have discussed the major approaches to conversation structure and utterance meaning which are currently adhered to. Grice’s theory on conversational maxims, provides a general framework for appropriate contributions in conversations. Unfortunate it is too general to be of any practical use. Austin and Searle’s approaches provide a little more guidance since they actually attempt to look at the reasons behind the use of particular utterances. Those theories fail unfortunately because they are too specific. By limiting themselves to specific utterances, Austin and Searle find themselves too tied to particular types of utterances and miss out on how this utterances fit into conversations as a whole. The work of conversation analysts on the other hand has focussed on explaining sequences of utterances but are hampered by their text-oriented non-cognitive approach. An alternative to these approaches is provided by Geis who uses interactional theory of language use to provide motivation for the structure and content of conversations. Geis’s theory also accounts for politeness diecisions which are made in conversations. It is this
theory of Geis's that will be used in chapters 4 and 5 as a motivation for choices in building conversational systems.
3.1 Introduction

In this chapter I will look at related work in the field of natural language systems. In doing this I will look at two particular areas. First is the work on conversation systems. These traditionally take a plan-based approach to natural language processing, and have tried to account for some of the regularities found in conversations. Second I will look at the issue of compiling information in natural language systems and in particular look at the SLANG system (Patten, 1988) which is the basis for the ANITA system described in the following chapter.

3.2 Conversation Systems

The most widespread approach to creating conversation systems has to look at conversation as planning. This allows the system builders to make use of all of the work in AI that has looked at how systems can build plans to achieve particular tasks (Sacerdoti, 1974; Schank and Abelson, 1977; Wilensky, 1983; Stefik, 1980).

The most thorough development of the plan based natural language systems comes from the work which has attempted to look at the intentions behind utterances (Allen and Perrault, 1980; Allen, 1983). In Allen’s system, TRAINS, utterances are mapped into steps
“how much is a ticket for the train to Rochester?”

REQUEST (JACK1, CLERK,
INFORMREF(JACK1,CLERK,x,EQ(PRICE(TICKET(TR1)),x)))

WANT(Jack1,INFORMREF(Jack1,CLerk,x,EQ(Price(ticket(TR1)),x)))

(INFORMREF(JACK1,CLERK,x,EQ(Price(Ticket(TR1)),x))

KNOWREF(Jack1,x,EQ(Price(Ticket(TR1)),x))

Give(Jack1,Clerk,Price(Ticket(TR1))))

Buy (Jack1,Clerk,Ticket(TR1))

Take-Trip(Jack1,TR1,ROC)

Figure 1: Allen’s Processing of “how much is a ticket for the train to Rochester?”
in a plan in process of being understood. An example of the processing which Allen’s system goes through to analyze one utterance can be seen in Figure 1. This example shows the steps required to analyze the utterance “How much is a ticket to Rochester?” The first step consists of parsing the utterance and then mapping it into a basic class of utterances based on its sentence type (declarative, interrogative, etc.). The system then uses rules to determine that if someone requests a ticket price that implies that they want to know it, which would require that they be informed of the price by the system. Informing them of the price would result in them knowing the price. Knowing the price would enable them to give the money to the clerk. Giving money to a clerk is part of buying a ticket action, which is in turn part of the take a trip plan. So by the end of reasoning through these steps the system can determine that the user’s question implies that they want to take a trip.

The problem with this analysis is the length of the reasoning path. Understanding this sentence requires six steps of logical deduction using inference rules. There are several clues in this sentence which should allow the system to bypass the long reasoning path. First, there is the context. The fact that the hearer is in a ticket booth at a train station should predispose he or she to interpret utterances as pertaining to train trips. Second, the fact that it is a “wh” question automatically tell us that the questioner has some need for either information or action from the clerk. Third, the occurrence of particular words like “train”, and “ticket” should prime the trip plan. The combination of the context and the components of the utterance should provide overwhelming evidence for the Take-trip plan without having to reason about plans and goals.

Another problem with Allen’s approach is that the utterance analysis relies heavily on traditional speech act theory. In his first step of analysis, Allen claims that the sentence
is mapped into a request. As discussed in the previous chapter this single utterance approach to speech acts is simplistic at best. There are many utterances which could be mapped into any number of speech acts and it misses out some of the regularities of conversations.

This type of system can be improved by giving the system access to the clues in the syntax and the ability to access social interactions. Those clues would enable the system to avoid having to resort to reasoning about plans and goals which could become expensive if there were a large number of plans in the plan library.

### 3.2.1 Discourse Strategies

An important part of the design of a conversation machine is the discourse strategies involved in conversation. In this section I will look at the different strategies used by other systems and determine which are truly necessary for conversation processing and which are merely artifacts of planning. By the term “discourse strategies” I mean the methods used to examine the organization that is assumed to be relevant to the analysis of a conversation. I will explore exactly what it is that researchers consider to be relevant to the flow of conversation and how this has affected their systems.

### 3.2.2 Litman and Allen

In their 1987 paper on plan recognition in for subdialogues, Litman and Allen attempt to separate plans which are specific to the task from plans which are specific to discourse processing. They identify three major classes of plans within the scope of discourse plans: continue, clarification, and topic shift.
The continue class refers to plans that extend the dialogue in an expected manner either by executing the next step in the plan or by inserting an utterance which talks about the next step in the plan in the case where the next step is non-linguistic. The clarification class involves plans that either identify a parameter to a plan or identifies a correction which must be made to the partially completed plan. The third class, topic shift, involves changes of topics in the conversation. These plans include introducing a new topic and modifying a previous topic so that it becomes a new plan.

It is useful to look at what these discourse plans add to the processing of conversation. As an example I will look at clarifications. From the point of view of my work, the fact that a utterance is a clarification must be recoverable from either its syntax, semantics, or some combination of these two with the current state of the conversation. In the two conversations in Figure 2 the same utterance appears, but it is performing a different action in each case. The utterance “I’ll take a chocolate doughnut” appears as the second turn in the first conversation and as the fourth turn in the second conversation. According to Litman’s analysis of conversations these two utterances would be mapped into separate discourse plans. The utterance in the first conversation will be mapped into plan continuation whereas utterance in the second conversation will be mapped into plan-clarification. This of course leaves us with the question: how does the system decide which one it should be? Litman’s system passes an utterance through a series of analyses on the way to complete understanding of the utterance. The first step is to map the utterance into it’s surface form as described in Allen and Perrault (1980). From a surface form it is
mapped into speech act, then into a domain plan in the manner of Allen and Perrault and into discourse plans.

<table>
<thead>
<tr>
<th>Short Conversation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clerk: Can I help you? T1</td>
</tr>
<tr>
<td>Customer: I’ll take three chocolate doughnuts. T2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Long conversation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clerk: Can I help you? T1</td>
</tr>
<tr>
<td>Customer: I’d like three cinnamon rolls. T2</td>
</tr>
<tr>
<td>Clerk: We’re all out of cinnamon rolls. T3 Can I get you something else?</td>
</tr>
<tr>
<td>Customer: I’ll take three chocolate doughnuts. T4</td>
</tr>
</tbody>
</table>

Figure 2: Two request sequences

3.2.3 Attention and Intention

Grosz and Sidner (1986) identified three major components of discourse structure: the sequence of utterances that make up a discourse, the purposes or intentions of the utterances, and the focus, or attentional space. Grosz and Sidner see intentions as composed of two parts: the purpose of the discourse as a whole and the purpose of the each segment of the discourse. The focus or attentional space determines which topics are currently being addressed. Both intentions and attentions appear in the systems explained here, although the are not explicitly designed into the system. Intention and attention are used in designing conversational stacks. The goal of the whole conversation my work is what Geis calls the
transactional effect of the social interaction. Completing a social interaction consists of establishing all of the felicity conditions for that interaction. This forces the conversation to be composed of subgoals, corresponding to felicity conditions, which are addressed either by a single utterance or by a series of utterances. The focus, at any given time is on the current subgoal and any knowledge structures used in satisfying that particular goal.

While this set up is useful since it provides a motivation for organizing topics and handling topic shift, it does not contribute any information about how to actually plan the conversation. In order to plan a conversation a system still needs some theory of how interactions are processed during a conversation.

3.2.4 Houghton and Isard

Another example of strategies used in discourse is found in Houghton and Isard (1987). In this model of conversation the authors include the notion of interaction exchange in their discourse strategies. This is based on the idea notion of adjacency pairs by Sacks, Schegloff and Jefferson which was discussed earlier. As I showed earlier this tracking of adjacency pairs complicates the processing without adding any important information to the conversation.

Interaction exchanges include an opening, a response and an optional closing. Their system uses this three part exchange as the basis for all the utterances in the discourse. The three-part exchange structure to discourse leads to a lot of excess and useless processing by any system using it. A reasonable discourse processing strategy must exhibit two criteria: first this organization must be easy to use in all conversations (or there must be an
explanation of when it will or will not hold), and second it must not add information at an unreasonable computational cost.

3.2.5 Sidner

Sidner (1983) looks at how to reduce the amount of processing that Allen uses by using what she calls situation-specific rules. These rules tell the computer that whenever the speaker says they cannot do something and they know the computer can do it, that they want the computer to do it for them. This is fine within the context that Sidner creates, but her rules are not flexible enough to account for different situations in which the computer might be engaged. In fact she holds all relationships factors between the system and the user as stable and therefore are not reflected in the system at all. Her method also involves creating models of the speaker which include what the system knows or assumes about them. This methodology allows the system to reason about why the speaker may have said certain things at certain times, by reasoning about possible goals and intentions. Like Allen’s work, Sidner’s can lead to a long inference path since there are no specific links between utterances and goals.

3.2.6 Relating Pragmatic Features to Planning

There has also been work in natural language system which looks at factors affecting syntactic and semantic choices that has not dealt with conversations. In PAULINE, Hovy (Hovy, 1986; Hovy, 1988) used what he called pragmatic features to generate several different versions of news reports. Possible features include: highfalutin and colloquial (which identify the general formality), pressured, unplanned, highly planned (depending on relative social status), and particular types of rhetorical plans such as convince and relate. Based on the features selected the system selects the plan and then generates the
text. One of the drawbacks of this system is its reliance on a phrasal lexicon, which includes not only words but also common phrases. This results in an unnecessarily large grammar since it must include many different types of phrases and suffers from one of the drawbacks of sublanguages: it is not easily expandable. Whenever a new word is inserted the system builder must verify which phrases it can and can not be used in, in order to ensure a consistent grammar.

3.3 Compiling Knowledge in Natural Language Systems

Compilation in knowledge based systems refers to the use of knowledge to bypass long chains of inferences. This type of knowledge has been widely used within the artificial intelligence community. The MDX system in medical diagnosis (Chandrasekaran and Mittal, 1983) showed a representation of diagnostic knowledge that allows a system to bypass excessive reasoning. Macro-operators in planning systems (Fikes, Hart and Nilsson, 1972); and the chunking mechanism in SOAR systems (Laird, Rosenbloom and Newell, 1986) have been widely used to reduce the necessary path for planning solutions.

In all of these cases compilation refers to applying knowledge to a problem solving task to skip reasoning stages. Figure 2 demonstrates the effect of compilation. In (A) the reasoning must proceed through four steps to reach from the start state (a) to the end state (e). In (B) the link between steps (c) and (e) has been compiled, now given start state (a), the final state (e) can be reached in only three inferences. Compilation can occur anywhere along the reasoning chain, in (C) a compiled link between (a) and (c) is added. In a highly compiled situation the entire inference chain is compiled, shown in (D), and the system moves straight from state (a) to state (e). Of course, one can not always just compile out intermediate states in reasoning chains. Usually the intermediate states could lead to
Figure 2: Compiling a chain of inferences. (A) contains no compilation. (B), (C), and (D) contain compiled links which bypass steps in the inference chain.
different final state depending on the conditions. It is important to know when perform this compilation at exactly what must occur to allow for the jumping over intermediate steps.

Compilation in this same sense has also been used in natural language generation. A major concern in natural language systems is what is the proper level of compilation necessary, or as stated above how many step can you compile out. An analysis of levels of compilation in natural language systems is provided in Patten, Geis, and Becker (1992).

3.3.1 Scripts

The notion of scripts (Schank and Abelson, 1977) is a form of highly compiled knowledge. Scripts refer to specific situations where the activities and the roles of the participants are well established. In these situations the conversations can follow very well established patterns almost to the same level of predictability as a script in a play (hence the term "scripts"). This knowledge can be compiled into a script which allows a computer system to merely following a list of meanings to communicate, in order, in the course of a conversation. A script for going to a movie would involve: the customer goes to the ticket booth, orders the ticket, gives the clerk the money, gets the ticket, goes to the food counter, orders food, gives the clerk money, gets the food, sits down in the theater. These actions are based on a stereotypical instance of going to a movie and are ordered within the script by when they must occur. Script are usually too highly compiled for normal situations. In the case of going to a movie, the customer may already have a pass, allowing them to skip the ticket buying part; or they may not want food, allowing them to skip the food purchasing section.
3.3.2 Frames

A less highly compiled form of language generation is seen in GUS (Bobrow et al., 1977). GUS is a dialogue system for booking airline flight. It uses frames which contain slots which must be filled in, in order to complete the booking activity. Using a frame based approach frees the system from the ordering constraints inherent in a script mechanism. The slots can potentially be filled in, in any order, which allows for mixed initiative dialogues. This type of frame based system is similar to the one used in the ANITA system described in the next chapter.

3.3.3 Compiled Mapping Between Context and Syntax

Another example of compilation in natural language systems is the SLANG system (Patten 1988). SLANG is based on Halliday’s theory of language as a social system. Patten uses a systemic grammar to represent the syntax in a “grammatical stratum”, and a “semantic stratum” to represent contextual elements. Utterance generation in this system is compiled by the links between the two strata. Both the semantic and the grammatical stratum represent information in hierarchies. Features in the semantic stratum hierarchy are linked directly to features in the grammatical stratum. Using this type of compilation Patten is able to generate utterances by simply selecting the relevant situational features in the semantic stratum the system then uses the links to determine the grammatical features which apply to that utterance.

A major drawback of the SLANG system is that it looks at utterances in isolation. It does not have a theory of conversation, and in it Patten makes no attempt to explain how a semantic stratum could be developed or used within a conversation system. The ANITA
system discussed in the next chapter attempts to offer solutions to this limitation of the SLANG system.

3.4 Summary

This chapter summarized two areas of research in the field of natural language systems. First are systems that attempt to account for utterances in conversations. This includes using planning as a basis for utterance. Primarily these systems suffer from two major drawbacks. First they do not have a theory of conversation as it relate to actions. Second they do not have any interactional components. They fail to take into account the relationships between the participants and the effect that has on the conversation. The second area of research I have looked at discusses methods of compiling information in
natural language systems. Compiled approaches to natural language include the use of scripts and frames for determining what to say at a particular point in the conversation. A second use of compilation involves creating a mapping between the choice of what to say and the syntactical decisions which determine how to say it. This second use of compilation was used in the SLANG system which will be used as a component to a travel agent interface discussed in the next chapter.
CHAPTER IV
A GENERATION SYSTEM FOR THE TRAVEL AGENT DOMAIN

4.1 Introduction
In this chapter I discuss the Implementation of the ANITA system (Patten, Geis, and Becker, 1992). ANITA is a natural language generation system which uses the theory of social interactions developed by Geis and a theory of compiling natural language generation used by Patten as the basis for building a language generator for the travel agent domain.

4.2 ANITA’s Conversation
Figure 4 shows a sample conversation for the travel agent domain (from Patten, Geis and Becker 1992). This conversation was adapted from real travel agent conversations, and it is the conversation which the ANITA system generates. In generating the conversation, ANITA plays the role of the travel agent and generates those utterances marked A. The client responses are entered into the system as fillers for slots.

4.3 ANITA’s Knowledge Structures
In order to process a travel agent booking ANITA requires two major pieces of compiled knowledge. First is the pragmatic stratum. The pragmatic stratum is a hierarchical
Setting: It is the 23rd of December and the Client wishes to travel on the 26th of December, which is a difficult time to get bookings. Conversation based on actual agent talk.

A: May I help you? T1
C: Do you have any flights to Miami on the 26th? T2
A: How many seats are you looking for? T3
C: One. T4
A: What time can you leave? T5
C: Some time in the afternoon. T6
A: Let me look.... I’m not finding anything then.... Can you leave earlier? T7
C: If I have to. T8
A: I’ve got a seat on an 11:00 a.m. flight on Treetop Airlines T9
C: That’ll be good. T10
A: When can I bring you back? T11
C: On the morning of the thirtieth. T12
A: Well, all I’m showing is a 10:00 p.m. flight. T13
C: Do you have anything the night before? T14
A: I can put you on that 10:00 p.m. flight. T15
C: That’ll be okay. T16
A: The round trip fare will be $295 T17
C: Okay. T18

Figure 4: A sample conversation for an airline booking
representation of the mapping between the context features and the logicogrammatical features.

4.3.1 ANITA's Pragmatic Stratum

ANITA's Pragmatic stratum allows the system to choose between options along several different dimensions. First, the system can select the style of the conversation, whether the conversation is going to be a formal or an informal conversation. The selection of formal or informal, determines which type of verbs can be selected from the dictionary for the main verb of the sentence. This allows the verbs in the dictionary to be marked with either a formal or informal feature and different verbs could appear based on the selection in the pragmatic stratum. For example, selecting informal might produce the utterance “when can you leave” whereas selecting formal might produce “when can you depart”. “depart” and “leave” are classified in ANITA’s dictionary as having the same base meaning but having a different formality.

Another dimension available in ANITA’s pragmatic stratum is that of politeness. In ANITA's limited range the only politeness choices available is the orientation of the utterance. The user could select either addressee-oriented which would make the user the subject of the utterance or it could choose subject-oriented which would make the system the subject of the utterance.

The third dimension available in ANITA's pragmatic stratum involves the actual gist of what is being communicated. Since ANITA is designed to be a interface for a travel agent the only action the system is aware of was booking airline flights. thus, the pragmatic stratum includes such activities as determining the users needs and desires: dates, times,
destinations and informing the user about what the system has available in the flight database.

4.3.2 ANITA's S-Structures

ANITA is based on Geis's social interaction theory. As mentioned in chapter 2, not only does Geis's theory use the abstract concept of request, offers, etc., it also provide for the refinement of these into more specific types of interactions. ANITA is designed for one specific type of request interaction, a flight booking interaction. ANITA was designed based on an earlier version of Communicative Social Interaction theory so the structures for ANITA do not contain all of the conditions in the current theory. ANITA structures, called significance-structures or S-structures, address only the initial state condition and the satisfaction conditions. The initial state condition on requests is “Init wants Resp to do A” (the requester wants the requestee to perform some action). The satisfaction conditions on requests are: “Resp is willing to do A” and “Resp is able to do A”. A request can not occur unless the requestee is both willing and able to perform the action the requester would like performed.

Since ANITA is restricted to flight booking requests and in the real world it is safe to assume that travel agents frequently book flights, this system uses a flight-request as a compiled form of knowledge which the agent has available. The difference between a general request and a flight-request is that in a flight request the action has been restricted to be a airline flight. The system also knows what the components to airline flight reservation are. Namely it has a depart time, an arrival time, an airline, the number of seats reserved, etc. The system also knows that when someone is engaged in the role of an clerk or agent they have waived their right to refuse a request based on willingness. Therefore
Figure 5: A Portion of the Anita’s Pragmatic Stratum
the ANITA system an S-structure which already has the willingness condition marked as true, and has the value of the ability condition is dependent on finding a flight that both matches the clients wishes and is available in the flight database. The S-structure representing flight-requests is shown in Figure 6. The initial state condition does not appear in the S-structure itself in ANITA. The initial state condition is used instead in the system to determine which S-structure to instantiate. This works out fine in ANITA it is highly unlikely that client would express the desire for a flight and not actually want to make a request. Of course, people contact travel agents to find out flight information, such as price, without intending to purchase a ticket. Clients normally signal this information early in the conversation. Instead of saying “Do you have any flights to Miami on the 26th?” they are likely to start with “I’d like to know how much it would cost to fly to Miami?”.

4.4 ANITA’s Planner

Anita’s planner exploits the S-Structures described above to tightly constrain the inferencing it must engage in. These S-Structures are a form of compiled knowledge that allows the planner to quickly perform its task. This section will illustrate the role of this kind of compiled knowledge in planning.

The planner’s activities are divided into three basic phases: a general need determination phase; a specific need determination phase; and a need satisfaction phase. The first phase, general need determination, involves determining which specific SA the speaker wishes to engage in. This really amounts to a greeting (ANITA asks “How may I help you?”) which the planner achieves by choosing the pragmatic stratum feature GENERAL-NEED-DETERMINATION. At present, ANITA is designed only to book air flights, but in a wider domain, it must be able to distinguish a need for hotel space or space
Flight-Request:

Agent-willing = TRUE
Agent-able = ?

Flight =

Origin: ?
Destination: ?
Leave-time (hour, date, AM/PM): ?
Arrive-time (hour, date, AM/PM): ?
Seats (number of seats): ?
Cost: ?
Carrier: ?
Direction (to flight or return flight): ?

Figure 6: S-structure for flight-request
on a cruise from a need for space on a flight. So at present, a request for an airline flight always becomes the target for the next two phases. It knows it should be an airline flight from the meaning of the clients first response which expresses the desire to fly to Miami. This addresses the initial state condition of the request and the word fly constrains the type of request to be a flight.

The specific need determination phase is activated when the planner receives (as input) a representation of the gist or S-Meaning of a client’s utterance. It begins the task of booking an airline flight by creating the S-Structures associated with airline bookings. The planner then addresses each of the slots in the S-Structures and attempts to get the agent and the client to agree about the values that will fill these slots.

The planner works with two main S-Structures. The first contains slots that the agent fills with information from the data base, the second contains slots filled with values learned from the client. The second S-Structure represents desires of the client: desired number of seats, desired destination, desired departure time etc. I distinguish negotiable desires (departure time, for instance) from non-negotiable desires (the destination, for instance). Each slot encodes a felicity condition (e.g., “Client X is willing and able to leave at time T”) that must be satisfied (or renegotiated and then satisfied, where the desire is negotiable). The Agent S-Structure encodes felicity conditions as well, in this case representing what ANITA is able to provide to the client in the way of destinations, departure times, prices, etc. During the specific need determination phase, ANITA inquires about slot values until it has enough information to commence an efficient search. The planner first attempts to determine the values for those slots involving non-negotiable conditions that do not yet have values, and continues until sufficient information has been
obtained to begin an efficient search. For instance, this may involve an attempt to determine
the value for the slot SEAT-QUANTITY, a slot involving a non-negotiable client desire.
The pragmatic stratum features chosen by the planner in this case are SEAT-QUANTITY
and CHANCES-NOT-AT-ISSUE—the latter because the chances of succeeding are not
relevant to a non-negotiable desire. It also outputs pragmatic stratum features
_corresponding to the fact that this is a new inquiry (ORIGINAL-QUESTION), and style
(INFORMAL) and politeness (CLIENT-ORIENTED) features (see Figure 5).

Once the client responds, the planner enters the need satisfaction phase in which it
attempts to match flights in the database with client needs. If the agent finds flights that
satisfy the client’s needs, then the planner outputs features that result in an assertion that
confirms its success and supplies information to the client concerning values for all
remaining slots. So, if the planner has found one seat on a flight leaving at the desired time,
it will output an assertion that conveys this fact and information about the flight number and
the price. This affords the client the opportunity to accept or reject the price. On the other
hand if the agent is unable to satisfy the client’s needs and desires, the planner backtracks
into the need determination phase, and outputs features resulting in follow-up queries about
those desires that are negotiable, such as departure time, in an attempt to set new values for
these slots.

When both the agent and client agree about the values of all the slots in the client’s
S-Structure, the goal of booking a flight is achieved.

The planner is implemented straightforwardly in OPS5 production rules that match
specific elements in the S-Structures. The planner is thus at the “partial pattern matching”
level of compilation. Given the degree of structure imposed on the planning by the domain
(and the S-Structures in particular), this level of compilation is justified. In cases where this kind of compiled text-planning knowledge is not available, a text planner would want to fall back on planning that is less compiled. Planning, like classification, has been identified by Chandrasekaran (e.g. 1987) as a generic problem-solving task. This means that planning could be attempted at the generic problem-solving level of compilation before resorting to first principles.

4.5 ANITA’s Register Knowledge

4.5.1 The Travel Agent Register

The special features of ANITA’s language in Figure 5 derive from special features of the Travel Agency Context of Situation (CoS), including, in particular, the fact that it is engaged in what might be called the “Booking a Flight Social Interaction,” the fact that ANITA enjoys the socially defined status as Agent, the activities it engages in determining the needs of the client and in attempting to satisfy them, and the tools that it employs in the process.

Many of the features of ANITA’s language reflect the nature of ANITA’s business. ANITA must engage in determining what the customer’s needs are and will normally employ questions to do so. Reports on successes and failures to satisfy these needs will normally take the form of assertions. A failure to satisfy an expressed need will typically result in a question to determine if the customer is willing to redefine his or her needs. Since I am speaking here of literal questions and assertions, these activities also determine the sentence type employed in making the utterance.
Other high-level grammatical features are determined by the specific tasks the agent is engaged in doing. Thus, while attempting to determine if there is a flight that meets the clients needs, if ANITA meets with failure, it can report that it doesn’t have such a flight, that it can’t find such a flight, or that it isn’t finding such a flight. The third case will occur, I believe, while ANITA is still actively engaged in the search for a flight, for it reports on an inability to find a flight so far. The first two report on completions of failed searches. Hence, “I’m not finding anything then” reflects the pragmatic stratum features SEARCH-IN-PROGRESS and UNSUCCESSFUL-SO-FAR (see Figure 5). This is, at least, how we understand human productions of utterances like “I’m not showing anything then.” The use of “show” reflects an interesting fact about this CoS. Computers are used by agents, and if an agent is to find a flight it must show up on the screen which makes salient the thesaurus feature !443vb (Roget’s “visibility”) and affords a travel agent the opportunity to lay the failure to find a desired flight onto the system, rather than himself or herself. If while engaged in a search (SEARCH-IN-PROGRESS), a travel agent is UNSUCCESSFUL-SO-FAR, and wishes to attribute its failure to find a desired flight to the system (SYSTEM-RESPONSIBILITY), the utterance will be a declarative sentence with a first person subject, a negation, the progressive aspect, and the verb “show”, giving rise to “I’m not showing anything then” or the travel agent might choose to accept responsibility (AGENT-RESPONSIBILITY), in which case the thesaurus feature will be !484vb (Roget’s “discovery”), and the resulting sentence will be “I’m not finding anything then”.

The distinction between the choice of verb here is of great importance. There seems to be a “logic of discovery” here according to which if the client is to have a flight, then the agent must be able to find a flight and if an agent is to find a flight, it must show up on the
screen. Instead of employing axioms expressing these relationships that ANITA must resort to in determining how to express itself, the thesaurus features for “show” and “find” are attached to features in the pragmatic stratum as part of the register mapping. This has the effect of compiling the lexical consequences implicit in this “logic of discovery.”

4.5.2 Implementing Register Knowledge

Register knowledge is implemented in ANITA by a pragmatic stratum. This pragmatic stratum contains a mapping between pragmatic features used by the planner and the logico-grammatical features used to generate the utterances. The method for processing the pragmatic stratum and generating the actual utterances was implemented by Patten (1988). The implementation used in ANITA is based on the SLANG (Systemic Linguistic Approach to Natural-language Generation) system developed by Patten. The version used for ANITA was reimplemented using a declarative representation of the grammar as opposed to the procedural implementation used in the original version. These differences are not relevant to the current problem and are discussed further in Patten, Geis and Becker (1992).

The system consists of several hierarchies. One for the pragmatic stratum and several others for the logico-grammar. Patten uses inheritance within the hierarchies as an important compilation mechanism for processing the hierarchies. By selecting a set of “leaf nodes”, which are nodes at the bottom of the hierarchy, the system uses inheritance to retrieve all of the information corresponding to the nodes on the path between the leaf nodes and the root of the hierarchy.

These hierarchies consist of two sections: the top section of the hierarchies containing *system features* and the bottom section containing *gate features*. The system
feature portion of the network is an And/or graph. At the bottom of the system feature section of the hierarchy are the leaf nodes. The gate features are features whose truth is determined by a combination of system features. These can be thought of like logic gates in computer hardware. The value of a particular gate is determined by the features feeding into it. Gate features come in two types: AND-gates and OR-gates. AND-gates are true if all their inputs are true. OR-gates are true if at least one of their inputs are true.

Figure 7 shows an example of a section of the clause hierarchy used by ANITA. The leaf nodes shown here are the marked and unmarked versions of NEGATIVE, DECLARATIVE-THEME and WH-THEME. Typically these would be the only features from the figure chosen by the planner. The reason is that all the others follow from these. If, for instance, the register knowledge preselects UNMARKED-DECLARATIVE-THEME and UNMARKED-NEGATIVE, then the following features can be chosen automatically: First, DECLARATIVE, INDICATIVE, FINITE and CLAUSE will be chosen since they are the ancestors of UNMARKED-DECLARATIVE-THEME. Similarly, NEGATIVE is chosen as an ancestor of UNMARKED-NEGATIVE (CLAUSE has already been chosen). Now since both NEGATIVE and INDICATIVE have been chosen, the gate NEGINDICATIVE-FINITE fires; and since both MARKED-NEGATIVE and INDICATIVE have been chosen, the gate REDUCED-NEGFINITE fires as well. Thus by choosing only the leaf nodes, the entire path (and hence all the syntactic constraints) is easily determined.

ANITA's implementation of register simply involves mapping from nodes in the pragmatic stratum hierarchy to the nodes of the logicogrammar. This is accomplished by attaching properties to the representation of each pragmatic stratum feature that indicate
Figure 7: A simplified logicogrammar fragment (extracted from Mann et al., 1981)
that certain features in the logicogrammar should be preselected (see Figure 5). These preselections are then performed any time the pragmatic stratum feature is chosen. The current implementation assumes that all the necessary leaf nodes in the logicogrammar will be preselected, and this allows the complete logicogrammar description to be computed without resorting to the establish-refine strategy described above. This implementation thus illustrates only the cases where the pragmatic stratum contains enough knowledge to completely determine all the logicogrammatical features.

Currently, many of the logicogrammatical features that need to be preselected are of the form NON-X, where X is some interesting property. It is possible that these features, among others, can be treated as defaults, and could be selected automatically by the grammar unless the feature X is chosen. This could make the register mapping even more concise than it is at present.

Most of the register knowledge exploited by ANITA takes the form of direct association of features at the “partial pattern matching” level of compilation. Associated with GENERAL-NEED-DETERMINATION, however, is a piece of canned text “How may I help you?” that bypasses the grammar altogether. This highly-routine greeting is thus at the “table-lookup” level of compilation. ANITA thus illustrates the two highest levels of compilation, and the hierarchical classification structure it uses for its knowledge would facilitate the generic problem-solving although this has not been implemented.

In summary, this implementation of register knowledge takes the form of a highly-compiled mapping from pragmatic stratum features to logicogrammatical features. For a more precise description of the notation and implementation details, see Patten (1988).
4.5.3 ANITA's Grammatical and Lexical Knowledge

The grammatical and lexical knowledge contained in this implementation is in the form of a large systemic grammar of English. The grammar itself is domain-independent, but the lexicon/thesaurus contains very few entries at the moment. The syntactic knowledge in a systemic grammar is encoded as constraints attached to the logico-grammatical features. These constraints are expressed in terms of a small number of structural primitives (adjacency, classification, unification, and a few others) where the primitives take thematic roles (and features in the case of classification) as arguments. For instance, the logico-grammatical feature DECLARATIVE has the constraint that the Subject is adjacent to the Finite verb, and the feature UNMARKED-DECLARATIVE-THEME as the constraint that the Agent is (is unified with) the Subject of the clause (see Figure 7). Building syntactic structures from these types of constraints is a trivial operation.

The lexicon/thesaurus is implemented in exactly the same way as the grammar--as a network of features and attached realization rules. Here the realization rules are, for the most part, assigning lexical tokens to terminal nodes in the constituent tree. As an illustration, I will describe the processing necessary to assign the lexical token “finding”. As described above, the register knowledge indicates that if the travel agent is taking responsibility for a failed search, the thesaurus feature will be !484vb (Roget’s “discovery”). The clause feature that indicates progressive aspect preselects the main verb to be the present participle (the feature is !ING). The entry for “finding” is thus the following (lexical/thesaurus features and constituents are preceded by “!” by convention, and “\{-\}” denotes conjunction following systemic network notation):

\[ ((!484vb \text{!ING} =\{-\}) !\text{FINDING}! (!\text{Verb} = !\text{finding})) \]
If the features !484vb and !ING have been chosen, an OPS5 rule will notice that the entry conditions for !FINDING! are satisfied, and it will assign the token !finding! to the verb. As the lexicon/thesaurus is developed, more features will have to be added to distinguish between “finding”, “locating”, “discovering” and so on, but these features will also be determined by the register.

Many of the items that must be generated in this domain (such as prices and times) will not appear in the lexicon/thesaurus. In these cases the pragmatic stratum preselects special features that are pointers to memory locations that contain the desired item. That is, rather than preselecting features such as !809n, features such as *TOTAL-COST or *DEPARTURE-TIME are preselected, where there are memory locations labeled TOTAL-COST (containing a dollar value) or DEPARTURE-TIME (containing a time descriptor).

4.5.4 An Example

T7: “Let me look.... I’m not finding anything then.... Can you leave earlier?”

4.5.4.1 Planning the Example

Once the planner has received sufficient information from the client to begin an efficient search of the database for a flight, the planner enters the searching phase. The database was designed by using flight schedules from airlines. The database contains a frame for each flight. The frame contains: the airline, number of available seats, origin, destination, flight number, etc. It does not contain actual seat number information, that type of information would only add to the size of the database without adding any interesting features. The first time the planner does this it also initializes a pre-search phase. The purpose of the pre-search is to notify the client that the search is beginning and that this could take some time.
At this point the planner outputs three features: SEARCH-ABOUT-TO-BEGIN—since that is the phase the planner is in, PAUSE-FILLER—since this is the S-Meaning of the utterance the planner wants to communicate, and INFORMAL—which just states the style of the conversation. These features result in the generation of “Let me look”.

Next the planner attempts to find a flight which satisfies the client’s needs and desires. In this case there is no flight to Miami on the afternoon of the 26th that has any unbooked seats. The goal of the planner is to notify the client of the problem, and then try and find alternatives that might be acceptable. The notification of the failure is done by outputting the following features: SEARCH-IN-PROGRESS—since the search has not yet been abandoned completely, UNSUCCESSFUL-SO-FAR—since no flight was found which fits the expressed desires of the client, AGENT-ORIENTED—the orientation which this agent chooses most often during the search phase, and INFORMAL—which again just states the style of the conversation. These features result in the generation of “I’m not finding anything then”.

When looking for alternative flights the planner uses its knowledge about what slots of the S-Structure are negotiable and which are non-negotiable. It knows that time is a negotiable field, and so it searches for flights that are near to the time expressed by the client. In this case the planner finds that there are flights that leave in the morning which have openings. The planner now must determine whether or not these flights are viable alternatives. This requires going back to the need determination phase in order to determine whether the client is willing to accept an earlier flight. The features output for this utterance are: QUESTION-TIME-BACKWARD—since the planner wants to know if an earlier time would be acceptable; DEPARTURE-INFO—since the question relates to the time the flight
leaves, rather than arrives; CLIENT-ORIENTED—since this is the default orientation for the planner in the need determination phase; NEED-SATISFACTION-UNKNOWN—the planner is asking if the client can leave earlier, which may not satisfy the client’s needs; CHANCES-NOT-AT-ISSUE—since the probability of need satisfaction is not at issue; and INFORMAL—the style of the whole conversation. These features combine to generate “Can you leave earlier?”.

4.5.4.2 Exploiting Register in the Example

To illustrate the exploitation of register in this example, I will consider the final utterance, “Can you leave earlier?” of T7. Taking each of the planner’s selections in turn, the first feature to be processed is QUESTION-TIME-BACKWARD. Attached to this node in the pragmatic stratum is a constraint that preselects the thesaurus feature !135adj (Roget’s “earliness”) for the Head of the Temporal element. QUESTION-TIME-BACKWARD has the parent TIME-FOLLOWUP (a follow-up question about times), which preselects COMPARATIVE-ADJ-HEAD for the Temporal element, and the thesaurus feature !296vb (Roget’s “departure”) for the Process (main verb). TIME-FOLLOWUP is a subclass of FLYING-TIMES which preselects the logicogrammatical feature OPERATIVE (one type of active voice). TIME-FOLLOWUP’s other parent FOLLOWUP makes no interesting preselections. The next ancestor of QUESTION-TIME-BACKWARD that has notable preselections is NEED-DETERMINATION which preselects the logicogrammatical feature POSITIVE.

The next pragmatic stratum feature selected by the planner is DEPARTURE-INFO, which in this case has no significant effect because it makes no preselections on its own and its ancestors are the same as those for QUESTION-TIME-BACKWARD. The planner then
selects CLIENT-ORIENTED, which preselects ADDRESSEE-SUBJECT and has no further effect as its only ancestor (SERVICE) makes no preselections. NEED-SAT[ISFACTION]-UNKNOWN is the next planner selection. This feature makes no preselections, and neither do any of its ancestors, but together with FOLLOWUP it triggers a gate that preselects UNMARKED-YES/NO-THEME, the clause feature MODAL, and the auxiliary verb feature !X-CAN for the Modal element. Finally, the planner selection INFORMAL selects the lexical feature !INFORMAL for the Process.

4.5.4.3 Grammatical and Lexical Processing in the Example

A complete description of how utterances are generated in the logicogrammatical stratum can be found in Patten (1988). Here I will just explain some of the features used and the realizations which result from their selection.

The first feature considered is UNMARKED-YES/NO-THEME. This feature has a realization rule which constrains the Finite (the verb carrying tense and aspect) to be placed at the beginning of the clause. The next feature MODAL conflates, or unifies the finite verb and the modal verb, which forces the modal to the front of the clause. The feature !X-CAN for the modal combines with the feature POSITIVE to lexicalize “can” as the first word of the utterance. Using the inheritance mechanism, the ancestors of UNMARKED-YES/NO-THEME are also marked true. They are YES/NO INTERROGATIVE, INDICATIVE, FINITE and CLAUSE. YES/NO places the Subject immediately after the Finite verb, and the feature ADDRESSEE-SUBJECT selects “you” for the Subject. Several other preselections combined with OPERATIVE place the process after the subject. !296vb and !INFORMAL where selected for the process in the pragmatic stratum, they are combined with the feature !STEM (selected by MODAL), to lexicalize the process as “leave” Then
(a) UNMARKED-YES/NO-THEME places Finite at front of clause

(b) MODAL conflates (unifies) Modal and Finite

(c) !X-CAN and POSITIVE lexical Finite as “can”

(d) YES/NO and ADDRESSEE-SUBJECT lexicalize “you” the subject and place it after the Finite

(e) OPERATIVE, !296 vb, !STEM and !INFORMAL, place the Process after the finite, and lexicalize it as “leave”

(f) COMPARATIVE-ADJ-HEAD and !135adj lexicalize the Temporal as “earlier” and place it after the Process at the end of the clause.

Figure 8: Generating “Can you leave earlier” in the ANITA system.
the feature COMPARATIVE-ADJ-HEAD and !135adj select the word “earlier” for the temporal and place it after the process. At this point the processing is complete.

The total real time taken to generate this utterance, including planning and processing the grammar, is approximately 4 seconds on a SPARCstation SS1.

4.6 Conclusions

ANITA shows a system which uses social interaction theory to develop compiled structures which can be used in generating appropriate utterances in a conversation. By using this type of knowledge in the planning phase the system was able to plan a coherent conversation which appears at face level to use adjacency-pair theories. In reality the structure of the conversation is determined not by rules on adjacency pairs by rather by the simple goal of filling in the s-structure within the system.
CHAPTER V
A GENERALIZED SYSTEM FOR SOCIAL INTERACTIONS

5.1 Introduction
One of the biggest drawbacks to ANITA is that the pragmatic stratum and planning system were designed with the conversation already in mind. The goal was to show that this type of knowledge could be used to generate appropriate utterances in context. Unfortunately, the result is that the pragmatic stratum can generate only a limited number of utterances, which in turn limited the range of possible contextual situations available to the planner. For these reasons, in this chapter I will describe a second system called ALICE (A Language Interface for Commercial-service Encounters). ALICE is a bidirectional pragmatic component for a language processing system which has been implemented in OPS5.

5.2 ALICE's Architecture
ALICE uses an extended pragmatic stratum designed by Geis(forthcoming). Using this extended stratum requires that the planning system be augmented to deal with a much wider range of possible contextual features. ALICE, as shown in Figure 9, consists of three major components: a pragmatic feature processor for generation, a pragmatic feature abducer for understanding, and planner/plan understander. These components deal with the processing
of the pragmatic knowledge used in a conversation. The pragmatic feature processor for 
generation has the role of taking a set of pragmatic features and generating a set of 
logicogrammatical features for an utterance. To do this, ALICE uses the same component 
described in the previous chapter, which is based on the SLANG system (Patten, 1988). The 
processing for this is described in the previous chapter.

The Pragmatic Stratum Abducer is responsible for mapping between 
logicogrammatical features and pragmatic features during the understanding phase of 
conversations. It uses a simple abduction algorithm which is described later in this chapter.

The Planner/Plan Understander is responsible for generating the pragmatic features 
during the generation phase and processing the pragmatic features during the understanding 
phase.

**5.3 ALICE’s Pragmatic Abducer**

The mapping from syntactic/semantic features to pragmatics features is done by 
performing a simple abduction algorithm on the pragmatic stratum. Abduction refers to 
reasoning from data to a best explanation for that data (Peirce, 1955; Josephson and 
Josephson, 1994).

In this case the data consists of features of syntactic and semantic features of the 
utterance. The input to this system is the output of a parser using a feature grammar. The 
pragmatic stratum for ALICE was designed around a systemic grammar, but any feature 
grammar such as HPSG (Pollard and Sag, 1987) would also work. Table 1, shows the input 
to the pragmatic understander for the utterance “Can you leave earlier?” The input consists 
of a list of features which are associated with each of the components of the utterance. For
Figure 9: ALICE’s processing
example the syntactic/semantic processing has determined that the utterance is operative. In the grammar this corresponds to having the subject proceed the finite in the utterance. There are also features associated with the subject, predicate, and adjunct of the utterance.

The goal of running an abduction processor on this data is to determine what subset of the features in the pragmatic stratum would account for these features in the utterance. The first step for the abducer is to find essential pragmatic features. These are pragmatic features which must be present in order to account for particular features in the input. In ANITA's pragmatic stratum this essential phase finds the pragmatic feature "departure-info" to be essential. It is essential because it is the only feature in ANITA's pragmatic stratum which can account for the predicator of the utterance being in the thesaurus class 809 (departure). This is reflected in the input features as the feature !809n.

Once all the essential pragmatic features are found the system then eliminates any other features in the input which are accounted for by the essential pragmatic features. The system then eliminates any pragmatic features from consideration which would conflict with the current set of candidate feature and adds to explanatory set the pragmatic features which are implicated be the features already in the explanatory set.

The system then repeats this cycle, by looking to see if any of the remaining possible pragmatic features are now essential and processing the results of these choices. This cycle is repeated until no more essential features are found. At this time it chooses the pragmatic feature which accounts for the most logico-grammatical features left in the data set, computes it's implications. This continues until there are no logico-grammatical left to account for. At this point the system stops and outputs the list of pragmatic features which were found.
Table 1: Input to Pragmatic Understander for “Can you leave earlier?”

<table>
<thead>
<tr>
<th>Utterance Component</th>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>UTTERANCE:</td>
<td>OPERATIVE</td>
</tr>
<tr>
<td></td>
<td>MODAL</td>
</tr>
<tr>
<td></td>
<td>MATERIAL</td>
</tr>
<tr>
<td></td>
<td>NO-RESIDUAL</td>
</tr>
<tr>
<td></td>
<td>POSITIVE</td>
</tr>
<tr>
<td></td>
<td>ADDRESSEE-SUBJECT</td>
</tr>
<tr>
<td></td>
<td>NON-EXTENT</td>
</tr>
<tr>
<td></td>
<td>NON-PAST-IN</td>
</tr>
<tr>
<td></td>
<td>NON-PRESENT-IN</td>
</tr>
<tr>
<td></td>
<td>TIME-ADJUNCT</td>
</tr>
<tr>
<td></td>
<td>NON-TEXTUAL-THEME</td>
</tr>
<tr>
<td></td>
<td>UNMARKED-YES/NO-THEME</td>
</tr>
<tr>
<td>UTTERANCE&gt;LOCATIVE/ADJUNCT/TEMPORAL:</td>
<td>NON-PRE-MODIFIED-AP</td>
</tr>
<tr>
<td></td>
<td>NON-POST-MODIFIED-AP</td>
</tr>
<tr>
<td>UTTERANCE&gt;LOCATIVE/ADJUNCT/TEMPORAL&gt;HEAD:</td>
<td>!135ADJ</td>
</tr>
<tr>
<td></td>
<td>!COMPARATIVE-ADJUNCT</td>
</tr>
<tr>
<td>UTTERANCE&gt;PROCESS/PREDICATOR/MODALSTEM:</td>
<td>!296VB</td>
</tr>
<tr>
<td></td>
<td>!INFORMAL</td>
</tr>
<tr>
<td></td>
<td>!NONREduced</td>
</tr>
<tr>
<td>UTTERANCE&gt;MODAL/FINITE/TOPICAL:</td>
<td>!X-CAN</td>
</tr>
<tr>
<td></td>
<td>!INFORMAL</td>
</tr>
<tr>
<td></td>
<td>!EN</td>
</tr>
<tr>
<td>UTTERANCE&gt;SUBJECT/AGENT:</td>
<td>PLURAL</td>
</tr>
<tr>
<td></td>
<td>NON-CLASSIFIED</td>
</tr>
<tr>
<td></td>
<td>NON-POSSESSIVE-NOM</td>
</tr>
<tr>
<td>UTTERANCE&gt;SUBJECT/AGENT&gt;HEAD:</td>
<td>!SECOND</td>
</tr>
<tr>
<td></td>
<td>!OBJECTIVE</td>
</tr>
</tbody>
</table>
The final result of running this abduction algorithm on the input in Table 1 is the following pragmatic features: QUESTION-TIME-BACKWARD, DEPARTURE-INFO, CLIENT-ORIENTED, NEED-SATISFACTION-UNKNOWN, CHANCES-NOT-AT-ISSUE, and INFORMAL.

5.4 Pragmatic Stratum

The pragmatic stratum that the system uses has its roots in the semantic stratum first introduced by Halliday (1978). He created a description of the possible functions available in an interaction between a mother and a small child. In his semantic stratum he identified a group of actions which a mother could take, including threats, and promises, and then associated with each of the functions the particular grammatical rules that would be used to implement that function. An utterance would then be created by selecting the relevant functions of an utterance and collecting the grammatical constraints which would then be passed to a systemic grammar and the result would produce an utterance. Patten’s SLANG system (Patten, 1988) used a systemic grammar and a small semantic stratum to produce utterances in isolation. A limitation of this system was that it made no attempt to explain how the features in the semantic stratum would or could be created.

ANITA attempted in part to address this problem of where the contextual features come from. The semantic stratum in ANITA was renamed the “pragmatic stratum” since by the time ANITA was complete it was clear that the information encoded in the stratum really contained pragmatic, not semantic information. As stated in the previous chapter, ANITA’s pragmatic stratum had serious limitations. ALICE’s pragmatic stratum offers a greater variety of utterance types.
ALICE's pragmatic stratum is much more complex than ANITA's. Instead of limiting the context to that of travel agents, ALICE's context is widened to include generalized service encounters. This gives the system the opportunity to engage in a wider range of conversations which allows for more contextual variations. The features available in ALICE's pragmatic stratum can be divided into three major types: style, politeness and gist. I will look at each of these types of features separately.

5.4.1 Style features

Where as ANITA only had the distinction between formal and informal, ALICE's formality is divided into five distinct levels: intimate, casual, neutral, formal, and frozen. The starting value for the this is determined by looking at the relationship between the conversational participants. The combination of the relative social power and social distance of the participant along with level of imposition created by particular interactions combine to determine the exact features to be chosen. The style of the conversation is more formal when the participants are more distant than when they are close. The style also is more formal when the speaker is at a lower power level than the hearer.

ALICE adjusts this formality as the conversation progresses by attempting to match the formality of the other participant. If the user follows the formality level as predicted by ALICE then that level will remain for the whole conversation. If the user selects a different formality level, then ALICE will mimic that level unless the user changes it again. The assumption is that the style of the conversation will remain relatively steady throughout the conversation.
5.4.2 Politeness features

ALICE uses the face-threat values to adjust the politeness in the conversation. ALICE tries to maintain a balance in the relationship during the course of the conversation. The notion of face threats come from work of Brown and Levinson (1987). They propose that politeness consists of two components: positive politeness and negative politeness. Positive politeness is politeness that accents a person's role as a helpful member of society. Negative politeness deals with effects that interactions have on a person's ability to exert control over their own environment. Offers boost the offerer's positive face, while requests decrease the both participants negative face since it puts an imposition on the requestee, and create a debt for the requester.

ALICE also has orientation features which determine the subject and topic of an utterance. Depending on what is being communicated, it might be most polite to orient the utterances towards the speaker, the hearer or the domain.

5.4.3 Gist Features

Gist features in the pragmatic stratum represent two types of information. First they can reflect the felicity conditions addressed by utterances. They can also address the actual values for parameters in the social interaction. For example, if the social interaction is a commercial request the actual item being requested could have several components which must be negotiated. In the sample dialog used by ANITA a significant number of utterances are used to deal with the actual components to a flight (e.g. date, time, number of seats) as opposed to interactional conditions on the interaction.
5.4.4 An Example of Pragmatic Features

In Figure 13, the pragmatic features corresponding to the utterance “Do you have hot chocolate?” are listed. There are six pragmatic features corresponding to this utterance, and one frame value. The first feature INIT-SPEAKER merely identifies the speaker of the utterance. In this case the speaker is the person actually doing the requesting, this is the person who is considered the initiator of the social interaction. The next feature, THING-REQUEST, comes from the type of social interaction being performed. In this case, the interaction is a commercial-thing-request, where the thing being purchased is a physical item, as opposed to a service. The third feature OBJECT-FOCUS is a politeness choice. The fourth feature, VALUE-REQUESTING implies that the value for a particular felicity condition or domain object is unknown. The alternative feature is VALUE-POSITING which means that the value for a condition or object is known and the speaker is merely informing the hearer about the status of that condition or object. These two features usually distinguish between interrogative and declarative utterances. The next feature TOKEN-VALUE means that the system is either looking for or proposing a true/false value for the current object or felicity condition. TOKEN-VALUE corresponds to yes/no questions. The alternative feature is TYPE-VALUE type value is used to refer to subparts of an object or preconditions on a felicity condition. TYPE-VALUE is used for Wh-questions such as: “how many would you like?” or “what colors do you have?”. The last feature NEUTRAL-FORMALITY means that the style of the conversation is neutral, it is neither casual nor formal.
5.5 ALICE's Planning Module

ALICE's planning module is responsible for using the CSI-structures (communicative social interaction structures), its knowledge about interpersonal relationships, and its knowledge of the pragmatic stratum to generate utterances and interpret utterances made by others. Generating an utterance entails selecting the pragmatic features which correspond to the particular state of the planner. Understanding an utterance consists of making changes to the current state of the planner based on the pragmatic features of the input utterance. In order to either generate an appropriate utterance in a conversation or understand the significance of an utterance, a planner must have the right type of knowledge.

A planning system can divide its knowledge into two main types. First there is the information that it needs internally to process the conversation: how to manipulate its data-structures in order to reach its ultimate goal. Second, it needs to know what information must be communicated with agents other than the system; these correspond to the pragmatic features in ALICE's pragmatic stratum. These pragmatic features involve all of the types of things that are important to communicate to others, as well as features the system would like to receive from others. The pragmatic features are those particular features which will be reflected in the utterances that a system generates.

Not all internal features are pragmatic features. For instance, in order for any system to succeed in reaching its goal it must have some function which measures whether or not its goal is considered attainable. Yet this is information that is only important to the system. In terms of a conversation, people never actually communicate that we have a goal we feel is attainable. If we didn't feel it was even remotely attainable we would discard it.
While information about the goal the system is trying to achieve may be central to the processing of the system there are other features, communicative-features, which while not necessary to attaining the system's immediate goal are necessary to the long term goals of maintaining social standing and good will with others. In order for a system to engage in truly "natural" conversation it must be aware of and capable of using the same types of features which humans might use in cooperative conversation.

For this reason this system has access to not only the traditional domain goal-subgoal hierarchies of other systems it also has access to information about interpersonal relationships and how to in some sense maintain these relationships.

5.5.1 The Relationship Matrix

The first part of this type of interpersonal knowledge is information about the roles which the system and users can engage in during a conversation. For instance, in the case of a sales clerk, the clerk really has access to several different roles. The clerk can be a clerk for a customer, in this role she has the responsibility of trying to please a customer, being friendly so the customer is likely to contact her again, and showing competence so the customer is likely to return. The clerk could also engage in a conversation with a colleague at which point the necessity of friendliness is not as rigid, although she is still in a business setting and expected to be professional she is not in the job of pleasing the colleague in the same sense that she is responsible for pleasing the client. The clerk may also in the course of the work day engage in conversations with friends and family who call or stop by, introducing the ability to engage in more intimate conversational patterns; or she may engage in conversations with bosses which will cause a more formal conversational style. A simple list of roles for the sales clerk would include: clerk, boss, employee, friend, parent, spouse,
etc. Likewise anyone she talks to has a range of roles which they could also choose from. Granted not all roles are compatible. The clerk can not choose “friend” while her manager chooses “boss”, but there is some matrix of possible choices. This “relationship matrix” forms the first piece of knowledge that the system needs in order to engage in conversations. The relationship matrix used by ALICE is shown in Table 2. The left side of the table shows the roles which ALICE can select. The top of the table shows the roles the system user may select.

<table>
<thead>
<tr>
<th></th>
<th>Clerk</th>
<th>Boss</th>
<th>Customer</th>
<th>Friend</th>
<th>Son or Daughter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clerk</td>
<td></td>
<td>3,Eq</td>
<td>3,&lt;</td>
<td>3,&lt;</td>
<td>3,&gt;</td>
</tr>
<tr>
<td>Parent</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>2,&gt;</td>
</tr>
<tr>
<td>Friend</td>
<td>2,Eq</td>
<td>3,Eq</td>
<td>*</td>
<td>2,Eq</td>
<td>3,Eq</td>
</tr>
<tr>
<td>Spouse</td>
<td>1,Eq</td>
<td>*</td>
<td>*</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>Customer</td>
<td>3,&gt;</td>
<td>3,&gt;</td>
<td>*</td>
<td>3,&gt;</td>
<td>3,&lt;</td>
</tr>
<tr>
<td>Trainer</td>
<td>3,&gt;</td>
<td>*</td>
<td>*</td>
<td>2,&gt;</td>
<td>2,&gt;</td>
</tr>
</tbody>
</table>

Each cell in the table contains a pair of variables which defines the default politeness measurements for the speaker/hearer role pair. For example in the case of clerk-clerk conversations the variable pair is (3,Eq) the 3 stands for a social distance of 3 on a scale of 1 to five, with 1 being very close and 5, very distant. This distance is a default only, the agent can choose to use either a closer or further distance under certain circumstances. The “Eq” in the matrix stands for an equal power relationship between clerks. In cases where the clerk speaks to her boss the power relationship changes to “<“, signalling that the boss
has a higher power position than she does. A “§” in the matrix indicates that role combination is not available. The numbers are meant to demonstrate a feasible matrix and will be used to demonstrate how the system works. They are not intended to necessarily be a valid description of sales clerk relationships. The question of what is actually the social distance between clerks and other is an important question but not with the scope of this work.

These two variables correspond to what Brown and Levinson (1987) call D and P respectively. They describe “D” as the ‘social distance’ of the speaker and hearer and “P” as the relative “power” of S and H. Brown and Levinson use these two along with a third measure to compute the effect of interaction by the initiator on the respondent.

The third measure Brown and Levinson use is “R” the absolute ranking of imposition in a particular culture. “R” would not appear in this matrix because it is not affected by the relationship between particular individuals, but rather on the interaction presently being performed. This measure would be the same for all members of society.

5.5.2 Communicative Social Interaction Structures

The second major type of knowledge which the system must have at its disposal is the information about how one goes about performing a particular social interaction. This involves knowing what are the particulars of an act, what parts must be agreed to by the participants, and what parts can be assumed to be given in a particular situation. This type of knowledge is contained within the system in the “CSI-structures”. CSI-structures are contains “slots” of information that must be filled in and agreed to by the parties in order for the social interaction to be completed successfully. CSI-structures are plans for
performing the social act. Social-interactions are complex cooperative activities such as requesting, promising, and inviting. In many cases these interactions require multi-turn conversations in order to be properly executed. An example of the CSI-structure of a commercial-request interaction is shown in Figure 10 (adapted from Geis (forthcoming)).

There are several different components to the CSI-structure. First is the Initial state condition. The initial state condition specifies the state which the participants must be in order to complete a successful instance of the social act. In the case of a commercial service request the initial condition is Desire (Init, O), which means that the initiator of the requests desires the object “O". The interactional effect of the commercial service request is that the object “O" will be transferred to the customer (Init). The interactional effect is the goal of the interaction. The satisfaction conditions specify other conditions which must be satisfied in order to complete the interaction successfully.

5.6 Generating an Utterance
Generating an Utterance consists of determining the style politeness and gist features for a particular context.

5.6.1 Selecting style features
Selecting the style value depends on what has previously occurred in the conversation. If a style level has already been selected, ALICE uses that level. If no style has yet been chosen, ALICE's selects the style features by looking up the relationship of the conversation participants in the relationship matrix. The social distance is the style level. That level is then converted into style features from the pragmatic stratum by rules which match the
Commercial Service Request:
Init = requester (customer)
Resp = requestee (clerk)
O = object (product)
M = money (price of product)

Initial-State-Condition: Desire(Init, O)
Interactional Effects: transfer(Resp, Init, O)
Satisfaction-Conditions:
   Able(Resp, transfer(Resp, Init, O))
   Precond: Possess (Resp, O)
   Willing(Resp, transfer(Resp, Init, O))
   Precond: Commit(Init, transfer(Init, Resp, M)

Figure 10: Example of a CSI-structure, Commercial-Thing-Request Structure
"Do you have hot chocolate?"

Pragmatic-Stratum Features:
- INIT-SPEAKER
- THING-REQUEST
- OBJECT-FOCUS
- VALUE-REQUESTING
- TOKEN-VALUE
- NEUTRAL-FORMALITY
*OBJECT = "hot chocolate"

Figure 11: Pragmatic Features for "do you have hot chocolate?"

level with the pragmatic features. A distance of 1 = intimate, 2 = casual, 3 = neutral, 4 =
formal, 5 = frozen.

5.6.2 Selecting Gist Features

Selecting gist features consists of determining the social interaction and the determining what felicity condition should be addressed. ALICE will initiate social interactions if the system has a a state which matches the initial condition of an interaction. ALICE will initiate a commercial-thing request if the system has a desire for some object and that object is marked as a sellable item. Hot chocolate is a sellable item so desiring hot chocolate causes ALICE to initiate a request. The felicity conditions are satisfied in the order in which they appear in the social interaction frame. Whenever a felicity condition has a precondition on it, the precondition is addressed before the condition. Gist features also address the clarification of parameters for a particular social interaction. For example in a commercial request, the actual item being purchased must be agreed upon by both parties. They must both
know the number of items, along with others attributes such as size, color or flavor. Clarifying the object is done in the process of negotiating the ability condition in ALICE, unless the ability condition is already established by the context.

5.6.3 Selecting Politeness Features

Selecting politeness features consists of determining what the current state of the face values of for the speaker and hearer and the effects the current utterance will have on them. Each participant in the conversation has two faces, a positive and a negative face. ALICE can increase the value of the positive face for the other participants by selecting the pragmatic feature PRODRESS. ALICE can increase the value of the hearers negative face by selecting REDRESS. The system can also try to minimize the effect of a statement it makes by selecting features such as RESP-ORIENTED or DOMAIN-ORIENTED which put the focus on the hearers abilities and desire as opposed to ALICE’s.

5.7 Understanding an utterance

Another extension of this system over the ANITA system is an understanding module, which shows how this same knowledge used in generation can also be used in utterance understanding. Like generation, understanding an utterance requires mapping between planner goals and pragmatic features. In understanding, the planner gets the pragmatic features as input, it also has some information about the participants and the current interaction. It uses this information to determine the effect of the speakers utterance on its internal state. Understanding an utterance consists of three tasks: determining the politeness effects of the utterance, determining any necessary change in the style of
conversation, and determining the gist of the utterance, which entails determining the social interaction, felicity condition and/or object being addressed by the utterance.

5.7.1 Determining Politeness

Politeness in an utterance is reflected in two separate types of features, face-threat features and orientation features. ALICE uses the face threat features to adjust her own face levels. If an utterance has the feature prodress, which means it pays respect to her positive face, she increases her own positive face value by one. If the utterance contains the feature redress then it pays respect to her negative face. In this case the system ups its own negative face value by 1.

The orientation features effect a general politeness level of the conversation. The general politeness level is effected by the interaction which is being engaged in and the felicity condition on that interaction which is being addressed. If the initial state condition is being addressed then and INIT-ORIENTED is chosen as the orientation. If a willingness condition is chosen and the power balance places the speaker lower than the hearer or the distance between them is more than 2, a domain oriented utterance is chosen. If the condition is a willingness condition and the participants are initiate and the speaker is equal or higher then a respondent oriented selection is made. Ability conditions require a domain oriented politeness choice.

5.7.2 Determining Style

Since there are only five style distinctions available to ALICE, determining the style of the utterance is fairly straight forward. ALICE selects a default style based on the social relationship between the conversational participants. She continues to use this as the style of
the conversation until she processes an utterance which is inconsistent with that style (it's analysis implies other pragmatic features.

### 5.7.3 Determining Gist

An utterance can perform one of two basic roles in a conversation. It can provide information any objects involved in the conversation. For example if the conversation is a request, an utterance may define the item which is being requested. The second basic role is to negotiate the felicity conditions involved in the exchange. Sometimes an utterance may perform both simultaneously, as in the case of “may I have a jelly doughnut?” This utterance if it occurs provides filler for the product slot in a commercial request, while at the same time addressing the willingness portion of a request.

### 5.7.4 Determining felicity conditions

Deciding which felicity condition a particular utterance pertains to is done by simply matching the features coming from the pragmatic stratum to a database which the planner has access to. This database explicitly states which features pertain to which felicity conditions. Naturally all features do not relate to felicity conditions. Those that are not relevant are ignored at this point in the processing.

### 5.7.5 Determining the Social Interaction

It is important to be able to determine which social interaction the speaker trying to engage the hearer in since that allows the hearer to recognize the speaker’s goals. One of the advantages of using compiled pragmatic knowledge is that it makes this recognition of the social interaction relatively simple.
5.7.5.1 A General Method for Determining Social interaction

Since social interactions consist of instantiating felicity conditions, one of the obvious determiners of what to say and when to say it is what felicity conditions need to be accounted for. Figure 12 shows the felicity conditions for three types of abstract CS-interactions: requests, suggestions, and invitations. These were created by adapting Searle’s (1969) felicity conditions to Geis’s (forthcoming) CS-interaction theory.

In this analysis, these interactions each have three conditions attached to them (invitations also have a fourth constraint, that the action benefits both Resp and Init). In the absence of any contextual knowledge, determining which interaction is being engaged in amounts to process of elimination. At the start of an interaction all three are possibilities, and if the speaker addresses a felicity condition he or she has an option of six different felicity conditions:

1. Init wants Resp to do A
2. Resp is willing to do A
3. Resp is able to do A
4. Init believes Resp should do A
5. Resp wants to do A
6. Resp wants to do A with Init

While some of these specify particular interactions, others specify a set of interactions. A mapping from felicity conditions to social interactions would appear as follows:

1. Init wants Resp to do A -> Request, Invitation
2. Resp is willing to do A -> Request
3. Resp is able to do A -> Request, Suggestion, Invitation
4. Init believes Resp should do A -> Suggestion
5. Resp wants to do A -> Suggestion
6. Resp wants to do A with Init -> Invitation
Requests =
Initial State Condition:
Init wants Resp to do A
Satisfaction Conditions:
Resp is willing to do A
Resp is able to do A

Suggestions =
Initial State Condition:
Init believes Resp should do A
Satisfaction Conditions:
Resp wants to do A
Resp is able to do A

Invitations =
Initial State Condition:
Init wants Resp to do A
Satisfaction Conditions:
Resp wants to do A with Init
Resp is able to do A
Domain(A):
action benefits Resp and Init

Figure 12: Felicity conditions for three types of social interactions
At the first step in a conversation, a system needs to set up a list of all possible social interactions, then as each utterance comes into the conversation, the system can compare its list of possible social interactions with the list of social interactions of which this particular utterance could be a part. If there is a social interaction on the list for which the current utterance can not be a part that, that social interaction is removed from the list of possibilities. When the list is narrowed down to one, the system has identified the social interaction. This is also useful for determining what utterance to generate at any given point. The system can select ones which could possibly whittle down the list.

As an example of how this works can be seen by looking at the conversation in (53) - (56). The first utterance can be mapped into the felicity condition “Resp is able to do A”. While it does not specify what A is, Bob is clearly attempting to find out if there are any obstacles to prevent Phyllis from doing something. Assuming Phyllis only knows about requests, suggestions and invitations, she would have to look at all three as possibilities, since “Resp is able to do A” is a felicity condition for all three interactions, so the set of possible interactions is \{request, suggestion, invitation\}. In this conversation Phyllis takes no initiative to determine which it is, but she does grant that she has no major plans. Then in (55), Bob addresses the felicity condition “Resp wants to do A with Init”. At this point Phyllis can determine that this new felicity condition can only imply an invitation, so requests and suggestions are removed form the list of possible interactions and she can now be sure that it is an invitation.

53) Bob: What are you doing Saturday afternoon?
54) Phyllis: Nothing.
55) Bob: Do you want to go to a movie with me?
56) Phyllis: Ok.
As I noted earlier, the invitation interaction also contains a restriction on the type of action with can be involved in the invitation, this action must be one which benefits both the initiator and the responder. So in the example above, going to a movie is a mutually beneficial action, but asking someone to come over and help move furniture is not. So another important part of determining the interaction involves looking at the object or action being specified and determining is which type of interaction that might imply. In the next section I will explain how these ideas have been implemented in ALICE.

This approach of mapping into felicity conditions and using felicity conditions to control the planner is desirable since there are a much smaller number of felicity conditions than literal meanings. A planner which needs to be able to handle all possible literal meanings would become very complex. A planner that limits itself to felicity conditions has a much simpler problem to solve.

5.7.5.2 Determining the Social Interaction in ALICE

ALICE has CSI-structures for several types of interactions. The system has structures for flight-requests (adapted from ANITA), commercial-service-requests, general-requests, threats, and offers. ALICE uses the general framework described in the previous section to determine which type of interaction is in progress. If a social interaction has already been determined at a previous point in the conversation then ALICE’s first goal is to see if the current utterance is consistent with that interaction. This is accomplished by checking the CSI-structure of the current interaction. If the felicity condition of the current utterance is part of the interaction then ALICE assumes the CSI-interaction is still in progress. If the utterance is not mapped into the one of the conditions on the current interaction then ALICE knows a new CSI-interaction is being engaged and updates the situation to that effect.
If there is no current CSI-interaction then ALICE’s goal at this point is to try to
determine what CSI-interaction the speaker is attempting to perform. ALICE does this by
first checking to see if she has a list of possible CSI-interactions. If this utterance is not the
first one in the conversation she will have created this list previously. If it is the first
utterance in the conversation then there will be no list.

In the case of no list, ALICE creates a high level list of possible CSI-interactions
based on the context. If ALICE is using the clerk role and the other participant is a stranger,
she assumes the role they are taking is “customer” which restricts the exchange to some
kind of request, either a commercial-service-request or an information-request. If the other
participant is a friend then the number of possible interactions is expanded to include
invitations and offers.

The system then updates this list by removing any CSI-interactions which do not
have the current felicity condition in their list of conditions. If the list is of possible
interactions is narrowed down to one, that interaction becomes the current interaction.
Whenever the interaction has been determined, ALICE can then start attempting to fill in
values for felicity conditions during her part of the conversation, instead of just replying to
the other participant’s utterances.

5.8 Examples of ALICE’s Planning Module
In this section I will explain the steps which the planning part of ALICE uses to map
between a particular situation and the pragmatic features of an utterance. I will discuss how
ALICE uses pragmatic features for three example utterances:

1. “Do you have hot chocolate?”
2. “I would like hot chocolate.”
3. “I want hot chocolate.”

A comparison of the pragmatic features for these three utterances can be seen in Table 3.

Table 3: Pragmatic features for three sample utterances

<table>
<thead>
<tr>
<th>PRAGMATIC-FEATURE</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>THING-REQUEST</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>OBJECT-FOCUS</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VALUE-REQUESTING</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VALUE-POSITING</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>TOKEN-VALUE</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>INIT-SPEAKER</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>INIT-ORIENTED</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PRODRESS</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NEUTRAL-FORMALITY</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

5.8.1 Example 1 - “Do you have hot chocolate?”

The state of the planner is that the initiator desires a hot chocolate, but does not know if hot chocolate is available (e.g. it is not on the menu). The initiator is the customer and responder is a clerk. These roles instantiate the social distance to 3, and the power balance to customer is higher than the clerk.

5.8.1.1 Generating the Example

In generation ALICE would be a customer in a store who is talking to a clerk.

INIT-SPEAKER falls out of the fact that ALICE is the initiator of the interaction in this example, the alternative to INIT-SPEAKER would be RESP-SPEAKER. In this
situation ALICE has as part of her current state $\text{DESIRE}(ALICE, \text{hot-chocolate})$. This is matched up with the CSI-structure for commercial-thing requests which then equates ALICE with the initiator.

OBJECT-FOCUS comes out of politeness choices. Since that current interaction is a request which has the potential for lowering the negative face of the hearer and the system is addressing the felicity condition $\text{POSSESS}(\text{Resp}, \text{hot-chocolate})$, the system chooses to focus on the object as opposed to themselves. This is a more polite form and mitigates the effect of the request on the hearer.

THING-REQUEST is determined by a rule which looks for requests interactions. Since commercial requests are requests, it then looks at the value of the requested item. The item is hot chocolate which is a sellable object (as opposed to an action or a object that can not be sold such as bathroom). So the feature THING-REQUEST is produced.

VALUE-REQUESTING comes from the fact the current felicity condition $\text{POSSESS}(\text{Resp}, \text{hot-chocolate})$ has no current value in ALICE’s CSI-structure. She does not know if the clerk has hot chocolate.

TOKEN-VALUE is a pragmatic choice that signals that there is a value for the variable "object" which is being requested. The value for that item will be supplied in the object variable.

NEUTRAL-FORMALITY is selected by the social distance in the relationship matrix. The social distance between clerks and customers is 3 which is mapped into neutral-formality.
5.8.1.2 Understanding the Example

If these features are input to the system, the processing is similar, except in the opposite direction.

INIT-SPEAKER tells the system that the user in this case is the initiator of the interaction. When that fact is combined with THING-REQUEST the system can infer that a request is being made of the system and that the system is expected to satisfy the users requests. This maps ALICE into the Resp role in the request interaction.

OBJECT-FOCUS is interpreted as a face saving effort by the hearer. The request interaction, places a burden on the hearer, so the negative face value for ALICE will be decreased, the choice of object focus will mitigate the effect of this to some extent.

THING-REQUEST combined with the context, tells the system the interaction is a commercial request. This is done by using ALICE's default role of clerk along with the fact that the speaker is a stranger.

VALUE-REQUESTING tells the system that the user is merely looking for confirmation of a fact. ALICE only needs to check for the presence of the token in her database. She does not have to provide any other information about it.

TOKEN-VALUE tells ALICE that the user is providing a value for the item requested. She can then look for the variable *OBJECT for the actual item being requested.

NEUTRAL-FORMALITY tells ALICE the formality is neutral, since this the formality the system expects in a client-clerk relationship nothing needs to be done to the
style level. If the client had used a different formality level, ALICE would have adjusted to match that formality level.

The result of this is that ALICE will create a frame for a commercial request with hot chocolate as the requested item.

5.8.2 Example 2 - "I would like hot chocolate."

A second example of how ALICE processes an utterance can be seen with the utterance "I would like hot chocolate." The same context holds for this example as for example one except that in this case the initiator (the customer) knows that the clerk possesses hot chocolate (e.g. it is on the menu). This allows the customer set both the ability and willingness conditions on the action to true. The only thing left for the customer to communicate is the initial state of the interaction.

```
"I would like hot chocolate."

Pragmatic-Stratum Features:

INIT-SPEAKER
THING-REQUEST
INIT-ORIENTED
PRODRESS
VALUE-REQUESTING
TOKEN-VALUE
NEUTRAL-FORMALITY
OBJECT = "hot chocolate"
```

Figure 13: Pragmatic Features for "I would like hot chocolate."
Some of the features in this example are the same as for “do you have hot chocolate?”. Both utterances share the features INIT-SPEAKER, THING-REQUEST, NEUTRAL-FORMALITY and TOKEN-VALUE. These occur in this utterance for the same reason they occur in the previous one. The speaker of this utterance is doing the requesting so they are the initiator, the request is still for an item not an action, the social distance between the two is the same, and since it is addressing the condition “init desires O” which can be either true or false it uses TOKEN-VALUE.

The three other features INIT-ORIENTED, PRODRESS, and VALUE-POSITING are different. VALUE-POSITING comes from the fact that the speaker in this case knows the value for this condition, it is true that the speaker wants hot chocolate. So in this example the speaker is entering this fact into the conversation in order to inform the speaker of this fact. The choice between VALUE-POSITING and VALUE-REQUESTING is made in the planner by looking at the slot for that particular condition. If that slot is empty, the utterance is VALUE-REQUESTING, if that slot is full the utterance is VALUE-POSITING.

INIT-ORIENTED is chosen because of the felicity condition being addressed, in this case it is “Init desires O” which is the initial state condition for requests. The planner has a link between this felicity condition and the INIT-ORIENTED feature.

The last feature PRODRESS is a politeness choice. Although customers are considered to have higher relative power than clerks, the two still have some social distance between them this distance requires some measure of politeness on the part of the customer. Since the initial state conditions implies a request and requests are impositions, the system
determines that it should minimize the effect by adding some politeness to the utterance, so it selects PRODRESS.

5.8.3 Example 3 - "I want hot chocolate."

Of course politeness is only an option, there are many times when this option may not be chosen. The relationship-matrix (Table 2) really represents a stereotypical version of the relationship between different people in society. Since it is unrealistic to suppose people are born with knowledge of social relationships, we can assumed they are learned. This means we can all learn different matrices. There are people in society who do not appear to use much politeness when talking to clerks. Some cases may be that they weigh the power balance as more significant than the social distance, in others they may see the social distance between clerks and customers as smaller.

This situation can be produced in ALICE by selecting different roles for the participant. If we choose the role of friend for both participants of a request then the distance is minimized and ALICE will choose all of the features from the previous example except PRODRESS.

The features chosen: INIT-SPEAKER, THING-REQUEST, INIT-ORIENTED, VALUE-POSITING, and TOKEN-VALUE, will produce the utterance “I want hot chocolate” instead of “I would like hot chocolate”.

5.9 Summary

ALICE has access to a wide range of pragmatic features which are mapped into particular contextual features within the planner/plan understander. These features allow the system
to draw a large number of conclusions without a long chain of inferences and to quickly generate appropriate utterances since they are linked directly to contextual situations within the system. It contains several types of compiled knowledge, the pragmatic stratum, the relationship matrix, and the interaction-structures which allow for a mapping between interactions and felicity conditions.
6.1 Summary of Work

On the surface generating appropriate utterances in conversation appears to be a formidable task. There appears to be a wide number of choices on what to say and how to express those thoughts at any given point in a conversation. In actuality, research on language variation has shown that language variation tends to follow particular patterns.

6.1.1 Language Variation

Changes in the activity the participants are engaged in or the topic can effect the words and the syntax used in conversation (Halliday, 1978; Montgomery, 1986; Cicourel, 1981; Ferguson, 1983; and O'Barr, 1981). Changes in the relationship between the participants (Labov, 1970; Halliday, 1978; and Cicourel, 1981) or the medium of the conversation (Biber, 1986; Chafe and Tannen, 1987; Halliday, 1989) can also effect the syntax used. Halliday (1978) termed these three types of contextual changes: field, tenor, and mode, respectively. Unfortunately Halliday's approach is too broad to be of much help in designing a computer system.

Geis (forthcoming) suggests an alternative approach to looking at language variation. His theory suggests four types of variation: dialect, formality, politeness and
register. Of these, only the last three are addressed in this work. Formality and politeness variation are effected by the relationships between the participants. Utterances which vary in formality share the same literal meaning, but may use slightly different words than each other. Utterances which vary in politeness will have different literal meanings but will share the same role in a conversation. Register variation refers to the use of particular phrases or words when talking about a particular activity.

6.1.2 Social Interaction Theory

Geis also recasts traditional speech act theory into a theory of social interaction. In this interaction theory, primary acts (Searle, 1975) are considered features of a conversational sequence as opposed to a single utterance. The utterances in a conversation address the felicity conditions on the interaction. In the case of a request interaction, the utterances may have to address the requestee’s ability and willingness to help the requester. This theory provides an explanation for what is said at any given turn in a conversation and it allows for the use of politeness considerations in determining the form of the utterance generated.

6.1.3 ANITA

The first system discussed in this work is the ANITA system. ANITA is a natural language generation for the travel agent domain. The system uses a hierarchical network of contextual features called a “pragmatic stratum” to link between contextual features and syntactic and semantic features. This stratum provides a method for accessing formality, politeness and register variation. The system also used Geis’s theory of social interaction for planning what needed to be said in a conversation. Using the pragmatic stratum allowed the planner to limit its concerns to the task that needed to be accomplished without having to account
for how what it wanted communicated would be realized in actual utterances. This allows for a clean separation of pragmatic information from semantic and syntactic information.

6.1.4 ALICE

The second system, ALICE is designed to work on utterances within a general commercial service context. It can handle both generation and understanding across a range of utterances, but has not yet been used on an entire conversation.

ALICE also contains other types of compiled knowledge not found in ANITA. ALICE’s relationship matrix creates a direct link between roles which participants select in a conversation and social distance and power values, these matrix is based on Brown and Levinson’s theory of politeness (Brown and Levinson, 1987). The power and distance values in turn allow for a direct connection to specific pragmatic features which in turn determine syntactic and semantic structures in utterances. Another form of compilation within ALICE is the use of social interactions structures. By looking at interactions as negotiations of felicity conditions, the system is provided with information about how to determine what interaction is being performed and how to determine what else needs to be said in the conversation. These structures can reduce the amount of processing needed during the course of a conversation since they provide a guide for select what to say and how to say it at any particular point in the conversation. The relationship matrix and the interaction structures determine the selection formality, politeness and register features which appear in the pragmatic stratum.
6.2 Future Work

This work shows how knowledge about relationships between participants and social interactions can be used by a planner or planner understander to help bridge the gap between the current situation and an appropriate utterance. There are several possible ways in which this work can be continued.

6.2.1 Further Pragmatic Expansions

An obvious next step in this work is to increase the size of the pragmatic stratum, and in doing so clarify which parts the pragmatic stratum will be usable in different contexts. The pragmatic stratum for ANITA was designed for travel agent scenarios, the stratum for ALICE was designed for general service encounters. The work will prove to be extremely useful if it can be shown that pragmatic strata can be developed for other contexts while being able to keep large sections of the current ones.

One step in extending the pragmatic stratum is to combine the strata of ALICE and ANITA into one stratum. This would enable ALICE's planner, which is more sophisticated than ANITA's, to be tried out on the ANITA conversation. I feel this would provide an interesting test of ALICE's politeness mechanism. Combining the two strata would also allow for developing strategies for working with multi-topic pragmatic stratum. If a computer system is going to be used in more than one domain it needs to know how to select determine which section of a pragmatic stratum are or are not relevant at any time.

Another area of research in extending the pragmatic stratum involves developing mechanisms for learning new pragmatic strata. I believe that a lot of the information contained in ALICE's pragmatic strata will be transferable to other domains. In particular,
the effect of formality and politeness should remain stable regardless of the activity. If not, it would imply that people do not care their knowledge of language use from domain to domain. Recreating a relationship between function and syntax for every situation would be so burdensome that language would not be a usable tool. Therefore, I believe that while changing domains will require adding domain specific knowledge it will not require a complete redesign of either the pragmatic stratum or the planning systems method of interacting with the pragmatic stratum.

Currently, the strata used by ALICE and ANITA have been developed by hand. Even though large sections of the stratum should remain unchanged in new domains, developing the stratum by hand could turn out to be a tedious process. A better method would be to automate it using some learning mechanisms.

Currently ALICE and ANITA can only really engage in one side of a conversation. ANITA has the ability to generate sentences in a travel agent domain, but has no ability to understand them. The client portion of the ANITA conversation is currently input into the system as fillers for frame slots. In essence the understanding has been done by the user. ALICE can either understand or generate, but when she generates she is a customer, when she understands she is a clerk. The reason for this is the limited pragmatic strata that the systems have to deal with. An extended pragmatic stratum would include the types of utterances both participants in a conversation would be expected to make.

6.2.2 Predicting Pragmatic Features for Understanding

Another way in which the processing can be reduced in understanding, would be by generating the expected pragmatic features for the next utterance. Since the system has access to
the current activity, and has some knowledge of the style and politeness factors involved, the system could make predictions about what the user is going to say next in the conversation. As the system stands now the first step in this type of extension would be easy. Some predictions of what to expect during understanding is for ALICE to simulate the role of the other participant in the conversation. Then ALICE could use pragmatic features that she would choose in that situation to predict semantic and syntactic features which might occur in the incoming utterance. Examples of use of pragmatic knowledge to reduce processing time can be found in Hartigan (1994) and Skon (1993).

6.2.3 Testing Theories of Conversation
The biggest problem with most theories of conversation is that they are just theories. It is very difficult in the real world to control all the factors which could possible affect utterance selection and understanding. In the ALICE system, many of these factors are easily manipulated. It is possible to adjust the relationship matrix to test effect of different social distance and power relationships on conversation. It is also possible to adjust the systems knowledge to make predictions about what would be said or understood given various interactions and background knowledge. An interesting way to test out the system and perhaps provide feedback for theories of conversation is to test the system on a range of users while manipulating the systems knowledge. Manipulating the formality and politeness factors in a conversation with people would allow for better refinement of the effect of these factors on real conversations.

6.3 Conclusions
From the small children to great scholars, just about every human being is capable of en-
gaging in conversation. The pervasiveness of conversation in life tends to make it appear simple, but research in computational linguistics has shown the opposite to be true. As easy as it is for a child to have a conversation with her parent, it still remains too complicated for computer systems to communicate at even a preschool level. Part of reason for the difficulty lies in the enormous amount of information which a conversational participant has to use in order to determine what would be an appropriate utterance at any given time. In order to engage in rapid conversation, humans cannot be engaging in a lot of run-time processing. The only alternative is that they have access to a large amount of pre-compiled information about conversation which enables them to perform quick run-time processing.

This precompiled knowledge must contain several parts. It must have knowledge of the functions of language. Human language is a tool. People employ it when we are attempting to perform actions which require interactions with others. It is reasonable to predict that since its major use is in communicating with others, that over time many of the reasons for its use have been embedded in its structure. As such this compiled knowledge needs to take into account the factors which are important to human interaction. These factors include the activities being engaged in as well the relationships between the participants.

In this work I showed how a possible design of this compiled information can be based on a linguistic theory by Halliday. I also showed how that type of compiled knowledge can be combined with a strong theory of conversation, such as the one proposed by Geis, to create a computational approach to conversational interactions. Using this type of pragmatic knowledge a system can use not only domain knowledge to engage in conversation, but can also use knowledge of rules of interactions with others in society.
This ability to access the cultural reasons behind utterance generation is necessary in order for computer systems to eventually engage in normal sounding conversations, which should be the ultimate goal of any natural language system. We should not have to learn to communicate with computers, computers should have to learn how to communicate with us.
LIST OF REFERENCES


21 Geis. M. L. (Forthcoming.) Speech acts and conversational interaction. (book ms.)


23 Good, D. (Mss.) The psychology of requests: the illocutionary assumption.


