Constraint-based Text Generation

Using Local Constraints and Argumentation to Generate a Turn in Conversation

Michael Elhadad

Department of Computer Science
Columbia University
New York, NY 10027
Elhadad@cs.columbia.edu

CUCS-003-90

February 1990

Abstract

In this paper, I present a natural language generation system aimed at constructing a turn within an ongoing conversation. Three main points are advanced: (1) previous discourse determines the production of a new turn; this paper describes techniques to take the influence of previous discourse into account when generating text. (2) the connection between previous discourse and a new turn is described as the interaction between five local constraints; local constraints are viewed as relations between one discourse segment of the previous discourse and the new turn. (3) argumentation is one class of local constraints which has important effects on the form of the language produced and interacts closely with the other types of local constraints used in this work. The approach is applied to the implementation of an explanation module for the ADVISOR expert system.

This work is supported by DARPA under contract #N00039-84-C-0165

1 Introduction

I propose working on the problem of generating text in the context of a larger discourse. I consider a discourse to be a segment of natural language, made up of several sentences or utterances and possibly produced by more than one author. I will specifically focus on the production of a conversation involving several participants.

Early work in text generation was concerned with generating a single sentence; more recent work has tackled the much more difficult problem of producing connected discourse. In this work, I address the problem of incrementally adding to an existing discourse. This is a small step towards the full problem of connected discourse generation. In the context of conversation, this means contributing a new move[1] to an ongoing conversation.

The problem is addressed by considering the task of text generation as a type of constraint satisfaction. The claims of the work center around the following points:

- When generating text, the influence of previous discourse must be taken in account. This influence is built incrementally and the effects of previous discourse on the following text cannot be planned in advance for a whole discourse. (This is particularly true in the case of a conversation, since one participant alone cannot control the development of a conversation *a priori*. It is therefore difficult to prepare a plan of what is going to happen in a conversation.) In particular, the form of an utterance is constrained by its position in a larger discourse.
- The relation between a new move and the preceding conversation is a complex interaction among several local constraints. In this work, I propose a classification using five categories of local constraints and show how the interaction between these five categories of influence can determine the production of new text. In contrast, previous approaches to the problem tend to describe such a relation by simply giving it a label from a set of possible ‘‘rhetorical’’ relations.
- Argumentation is an important aspect of language production. In this work it functions as one type of local constraint. I show how argumentation can effectively determine several aspects of text production - both in selecting content and in determining linguistic form - that are not easily accounted for by more conventional descriptive tools.

In the rest of this introduction, I state more precisely the problem addressed in this work by illustrating each one of these three points with examples.

1.1 Generating Text within a Larger Discourse

The production of text within a larger discourse is clearly constrained by the previous discourse. There are many different types of influences that discourse can exercise however. The following three examples illustrate three types of influence from the beginning of a discourse on the production of its last turn.

The two sentences of sequence (1) do not appear coherent. In (2) however, the heading mentioning ‘‘ski’’ is sufficient to establish an implicit connection between Harry’s fall and the snow. If the idea of ‘‘snow’’ is not otherwise introduced in the previous discourse, the connection between the snow and the fall must be expressed explicitly, as it is in (3). If a text generator has to generate a sentence expressing the content that the ‘‘snow was cold and wet’’ after the first sentence mentioning Harry’s fall, it must be able to choose the right way to refer to objects. Example 1 shows that the proper way the generator must refer to objects or concepts in the text is determined by the way they have been introduced into the discourse. We want to make the text generator sensitive to these different types of constraints so that it will generate a sequence like (2) or (3) but not like (1).

---

[1] A precise definition for ‘‘move’’ will be given later in the paper.

Example 1:
(1)* Harry fell several times.
The snow was cold and wet.

(2) **Learning to ski.**
Harry fell several times.
The snow was cold and wet.

(3) Harry fell several times.
The snow on which he fell was cold and wet.[2]

Example 2 illustrates another type of influence. The preparation utterance A1 can be seen as imposing an orientation to the question A2. By mentioning his distaste for broccoli, A makes it clear that he does not want the quiche to contain broccoli. The question A2 is now ‘‘polarized.’’ In her answer to A2, B takes this orientation into account. Instead of B1, a simple ‘‘yes’’ would simply answer A’s question literally, and would ignore A’s concerns. It could therefore create the inference that whether A eats the quiche or not is indifferent to B. When producing her answer, B selects the content of B1 to satisfy the constraints imposed by the orientation of A1, and chooses a way of phrasing this content which indicates how she is sensitive to this constraint. Note that if the question is not otherwise polarized by a preparation move, a simple ‘‘yes’’ is appropriate, as in (5).

Example 2:
(4) *A1. I hate broccoli.*
A2. Is there broccoli in the quiche?
B1. Yes, there is, but not much.

(5) *A2. Is there broccoli in the quiche?*
B2. Yes.

Example 3, finally, exemplifies yet another type of constraint from the position of an utterance within a larger conversation. In (6), the ‘‘thanks’’ in A3 closes the segment of the conversation. It is the way of closing the interaction between the customer and the salesman. After this closure, special linguistic devices must be used to ‘‘re-open’’ the conversation (such as ‘‘oh, also’’). In (7) in contrast, if the same request is expressed before using a closing device, such devices are not needed, and ellipsis can be used (there is no need to repeat ‘‘do you have’’).

Example 3:
(6) *A1. Do you have the Times?*
B2. Here you go.
A3. Thanks.
A4. Oh! also, do you have Newsweek?

(7) *A1. Do you have the Times?*
B2. Here you go.
A3. and Newsweek?

These examples point out the diversity of the constraints that one must address when generating a contribution within a larger discourse: in example 1, the connection between the entities evoked in the new turn and the previous discourse constrained the form of referring expressions; in example 2, the attitude of the speaker vis a vis the information she gave explained the addition of further information and constrained the choice of a connective; finally, in example 3, the position of a question before of after the closing of a sequence constrained the choice of

---

[2] Examples (1) and (2) are taken from [McCoy & Cheng 88]

cue words and determined the use or non use of ellipsis. I consider a situation where a previous discourse is already produced and one has to contribute a new turn to this discourse. I assume that a set of possible relevant contents has already been determined by another program, and is represented in some knowledge representation formalism.

The problem I address is the generation of one move in the context of an ongoing conversation. The task is, therefore, to develop a *dynamic model* of conversation construction. Such a model must describe how the influence of previous discourse constrains both the selection of ‘‘what to say’’ from the set of possible contents at a given point of the conversation, and ‘‘how to say’’ it in a proper way.

1.2 Generation as Constraint Satisfaction: The Notion of Local Constraint

In this work, I distinguish between *local* and *global* aspects of discourse organization. In a global perspective on discourse, one tries to identify the ‘‘outline’’ of the discourse: some hierarchical structure underlying its progression. The goal is to create a summary of the discourse that captures some of its properties. In contrast, a local perspective has a narrower scope. The focus is on the relation between two discourse segments. A *local constraint* determines properties of each segment, independently of their location in a larger structure. I call the segment of the previous discourse creating the local constraint the *source* of the constraint, and the segment of text being created under the influence of the source the *target*. A local constraint is therefore a relation between a source and a target, represented as `C<Source, Target>`.[3]

The previous examples have shown that the type of influence from the previous discourse can vary. Accordingly, there are different classes of local constraints. Example 4 illustrates how several local constraints of different nature interact to determine the production of one turn in the conversation.

Example 4

(8) *A1. do you want to go out?*
B2. it’s too cold out there.
A3. it is not very cold.
B4. too cold for me.

(9) *A5. It is not very cold.*
B6. <>

In (8), A3 is an argumentative opposition to B2. It is interpreted as an expansion of the exchange A1-B2: by refusing the answer that B gave, A is in some way ‘‘re-posing’’ the question A1. A can decide to use an utterance which is on surface a simple negation to repose a question because in this position within the exchange, the argumentative contradiction is interpreted as an initiative turn, and obliges B to answer. Note that the use of this form in A3 is determined by the position of A3 in the structure of the conversation and by its argumentative orientation with regard to B2. In (9), in contrast, nothing makes A5 an initiative turn. B cannot form a complete interpretation of the turn, and does not have to react to it. A cannot use the same form to pose a question

The form of A3 must therefore be determined using structural factors (where is A3 positioned), argumentative factors (what is the orientation of A3, and how is it related to the previous turns), and illocutionary factors (is A3 a simple statement, or is it an initiative move requiring B to react). The important point of this example is that all these factors are extremely interdependent: the illocutionary status of A3 is in part determined by its structural position, and in part by its argumentative orientation. All the constraints taken together can determine A3’s form; if one of these is missing, the task of deciding how to phrase A3 is largely underspecified. The connection between

---

[3] This notation is derived from [Moeschler 85].

the new turn and the previous discourse must therefore be explained along several dimensions.

In general, I maintain that the task of text generation is a problem of satisfying a set of interacting constraints. Each class of local constraints corresponds to a separate perspective on the discourse. It is the interaction between these separate constraints which determines the production of new discourse.

Such a multi-dimensional approach to discourse analysis has been used in previous work. In section 3, I will discuss the division in three meta-functions proposed by the systemic school in linguistics, Grosz and Sidner's theory of discourse structure which also distinguishes between three levels of analysis. I use a classification in five categories derived from these two approaches and from [Moeschler 85]: structural, topic, illocutionary, interactional and argumentative.

The main focus of this work is to describe a mechanism to handle the interaction of several constraints on the text generation process. I first propose a classification of local constraints and study the interaction between these constraints. Second, I present a computational mechanism to represent these interactions. I use a unification-based formalism to describe each local constraint, and show how an extended form of unification can be used to model constraint interaction.

1.3 Argumentation and Generation

One of the classes of local constraints I will focus on is *argumentation*. The idea behind a theory of argumentation is that language is used to present facts as supporting or opposing certain conclusions. Argumentation is one way for a speaker to express his attitude concerning the utterances he produces. There are actually two levels to the study of argumentation: conceptual and linguistic. The conceptual study of argumentation identifies how facts can be selected by an agent to serve as arguments for a conclusion and how one actually modifies his beliefs after hearing arguments. The linguistic study of argumentation identifies how the language used to express facts is modified by the argumentative function of the utterance.

An analysis of example 5 illustrates the use of argumentative constraints:

Example 5

(10) A1. I have a description on the back of my instruction set with the numbers 2 and 3 in circles.
A2. Do you have the same numbers?
B3. Yes, the numbers are there
B4. but there are pictures of the crane on mine.

Segment (10) is part of a transcript collected in the following situation [Elhadad 87]: A and B are working together on a common task, and are trying to divide the work. One way to decide how to divide the task is to assign a subtask to the person who has more documentation available on it. In A1, A tries to determine whether this technique of task allocation can be applied. The question in A2 is focused on a detail of the descriptions, but the situation and the preparation in A1 make it clear that A is trying to decide whether B has a different set of instructions. B's answer must address this underlying argumentative orientation. Note here the similarity between argument recognition and plan inference as described in [Pollack 86, Sidner and Israel 81, McKeown 88, Litman 86, Allen & Perrault 80].

Now, given the pool of facts that are known to B, she can present the same facts as arguments for or against the conclusion that the instruction sets are similar. B can observe her instruction sets, but has no information on A's information set, aside from what A has told her. She now has to form a judgment on how similar the two sets are, based on such partial information. This is basically the problem of argumentation at a conceptual level: how can one reach a judgment (on a gradual scale) based on partial premises. It is in other words a problem of *partial inference* [Jayez 88].

The problem of conceptual argumentation is hard. In this work, I will only address one aspect of it. I introduce the notion of *argumentative strategy* to study the relation between argumentation and content selection. The details of the mechanism will be described in section 6.7. The effect of having different argumentative strategies is that, given A's question in (10), and for the same set of facts available to B, different contents and argumentative orientations can be selected to answer the question. I will contrast the *agreeing strategy* (AS) with the *disagreeing strategy* (DS). With the AS, B will first try to select facts that can be presented as arguments supporting the argumentative orientation of A's turn. B's answer in this case, would be only B3. With the DS, in contrast, B will try to select facts that can be presented as arguments against the conclusion supported by A. In this case, B's answer will also include B4. I will not investigate what factors can influence the choice of argumentative strategies, but I will show how this choice can influence both content selection and surface realization in a computational system.

I focus more in this work on the linguistic aspect of argumentation, and its relation to the other types of constraints used. In section 4, I list the types of linguistic decisions that are made possible by a linguistic study of argumentation: choice of connective, sophisticated treatment of the determiner group of noun phrase and choice of adjectives.

Argumentation has not been used in generation work before. There are, however, several existing systems doing argument understanding. Most of these focus mainly on the conceptual study of argumentation. In section 3, I will discuss Robin Cohen's work on the structure of argumentative discourse, and Birnbaum's work on argument networks. In linguistics, a comprehensive theory of argumentation is currently being developed by Ducrot and Anscombre, and many more detailed works are relevant. I will discuss some of these results as well in section 3.

1.4 Statement of the Problem

In summary, I propose to develop a computational system of text generation, able to generate a turn in an ongoing conversation while taking into account several types of interacting constraints imposed by the previous discourse. I aim at developing a system that takes the following items as input:

- A description of the previous conversation.
- A set of relevant content specifications.

and it must:

- filter the relevant propositions and select only a subset,
- express this filtered content as a new move in the conversation which satisfies the local constraints imposed by the previous conversation.

This abstract task will be applied in the domain of a question-answering expert system. The proposed work will extend an existing system called ADVISOR, developed at Columbia [Mckeown88]. An overview of this application is given in section 2.

1.5 Contributions

The contributions of this work to the field of text generation are the following:

- *Systematic use of the notion of local constraint* to explain the coherence of conversation sequences and its effects on the form of the utterances.
- *Multi-dimensional description of coherence* and formulation of the problem of text generation as a problem of constraint interaction.

- *Extension and use of a unification-based formalism* for new applications, beyond syntactic processing.
- *Generation as part of conversation* (as opposed to text generation) and evaluation of the differences between text and conversation generation.
- *Use of argumentative constraints* and study of their interaction with other types of constraints.

All these points contribute to the development of a more sophisticated surface language generator. Its new capabilities include:

- A procedure for the selection of connectives;
- A sophisticated treatment of determiner sequences in noun phrases;
- A procedure for the selection of gradable adjectives;

The generator benefits from the work in that it can now distinguish between surface forms that are close in syntax and semantics, but differ in their discursive function. It can therefore generate a larger variety of text, that is also more appropriate within an extended conversation.

In the rest of this paper, I first present an overview of the system I aim at developing, giving a motivation for the development of the model proposed in this work. I then review previous work relevant to the problem of text generation in context. Then, I delimit the scope of the work: I will concentrate on only a limited set of text generation decisions and show how these decisions are made possible by the approach I propose. In section 5, I present the set of local constraints I will use in the system. Section 6 focuses on the argumentative local constraints and describes the type of interactions that can occur between argumentative and other types of local constraints. In section 7, I describe a computational formalism based on functional unification that is particularly well suited to the task. Finally, in section 8, I present details on the design and implementation of the explanation module for the ADVISOR system and the more general tools I have developed.

2 System Overview

I present in this section a brief overview of the practical system I aim at building. This system illustrates the types of problems I address and motivates the model for text generation presented in this work. The system will be an extension of the ADVISOR system. ADVISOR is an expert system capable of advising university students about courses to take in a semester. It includes a natural language interface. Most significant is its ability to keep track of the user's goals over a conversation, and to tailor the explanation it can produce to the particular goal pursued by the user. The example in Figure 1, from [McKeown 88], illustrates the current capabilities of the system.

The proposed work aims at providing more ‘‘natural’’ explanations. The same techniques will be used to produce the content of the explanation but the new model described in this proposal will be used to organize the explanation and phrase it differently. The main addition is therefore to rewrite the surface explanation generation module. The goal of the extension is to produce explanations of the type shown in Figure 2.

These examples illustrate several aspects of how the model presented in this work can make ADVISOR's generated answers and explanations of better quality. In particular, the following points are important in the context of the ADVISOR system:

- It is possible to present information with different salience: the model provides tools to modulate when information is expressed implicitly (as in II and III, the negative answer is left implicit), when presented as given or assumed (as in I, the ‘‘since’’ proposition is given information, or in III, the ‘‘even though’’ proposition), or when presented as new information (as in II, ‘‘calc_1 is a freshman course’’). This

```
Q: Should I take data structures this semester?

1. Goal = fulfill requirements
   You should take data structures.  Data structures is a requirement.
   All major courses require it.

2. Goal = choose courses for current state
   You should take data structures.  I assume that you want to take
   courses in the normal sequence.  Data structures is a first term
   sophomore course.  You have taken the preceding courses.

3. Goal = choose courses for current state; negative response
   You should not take data structures.  I assume that you want to
   take courses in the normal sequence.  Data structures is a first
   term sophomore course.  You have not taken intro and calc_1.
   They are first term freshman courses.

4. Goal = learn more about AI
   You should take data structures.  I assume that you want to
   concentrate on AI.  Artificial Intelligence is an AI course
   and requires data structures.
   All other AI courses require artificial intelligence.
```

Figure 1: Interaction with the current version of ADVISOR, from [McKeown 88]

```
I Q: Should I take Data Structures?
A: Goal=concentrate on AI
   Since you want to concentrate on AI, you should indeed take
   Data Structures.  It is a prerequisite for intro to AI.


II Q: Should I take data structures this semester?
A. Goal = choose courses for current state; negative response
   You could take it, but you have not yet taken calc_1.
Q. Why calc_1?
A. Calc_1 is a freshman course. Therefore you should take it before
   Data Structures which is a sophomore course.


III Q: Can I take computability instead of fundamental algorithms?
A. Goal = avoid programming
   Yes, you can take it.  But fundamental algorithms would be more
   helpful even though it requires more programming
   because it is a prerequisite for all software courses.
Q. But I plan on concentrating on theory.
A. You still need to take the required software courses.
```

Figure 2: More natural interaction

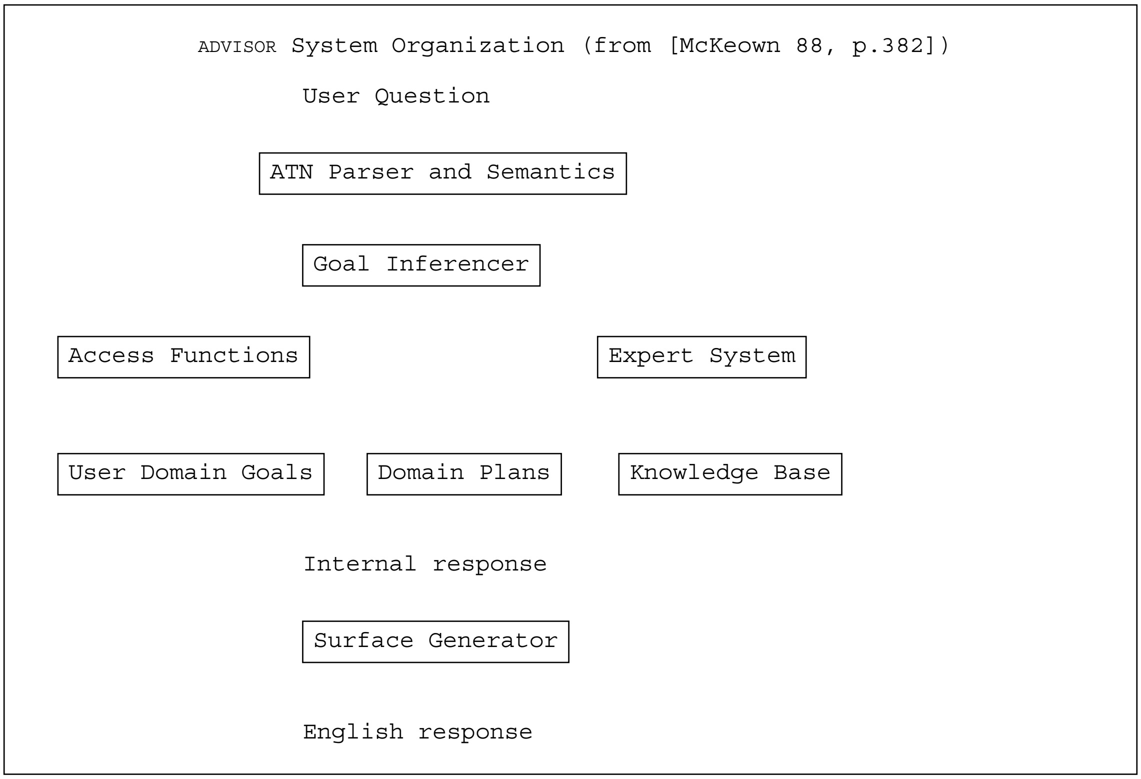
modulation makes the answers much more fluent.

- Because the explanation system relies on argumentation, it is possible to provide ‘‘positive answers with a negative orientation’’ as in II. What this means is that the answer does not have to be totally positive or negative. It can be more scalar, without having to resolve possible contradictions.
- Because the model can describe contradiction, the system goals and the user goals can be different. It is possible, as in III, to recognize a user goal and to present a conflicting goal in the answer. It is therefore possible to reason about goals like ‘‘minimize amount of work’’ or ‘‘take easy courses’’ which are not addressed in the current system.
- Because the model is sensitive to the structure of the conversation, it is possible to devise complex criteria for mixed initiative. For example in III, the followup reaction ‘‘But I plan ...’’ is syntactically a

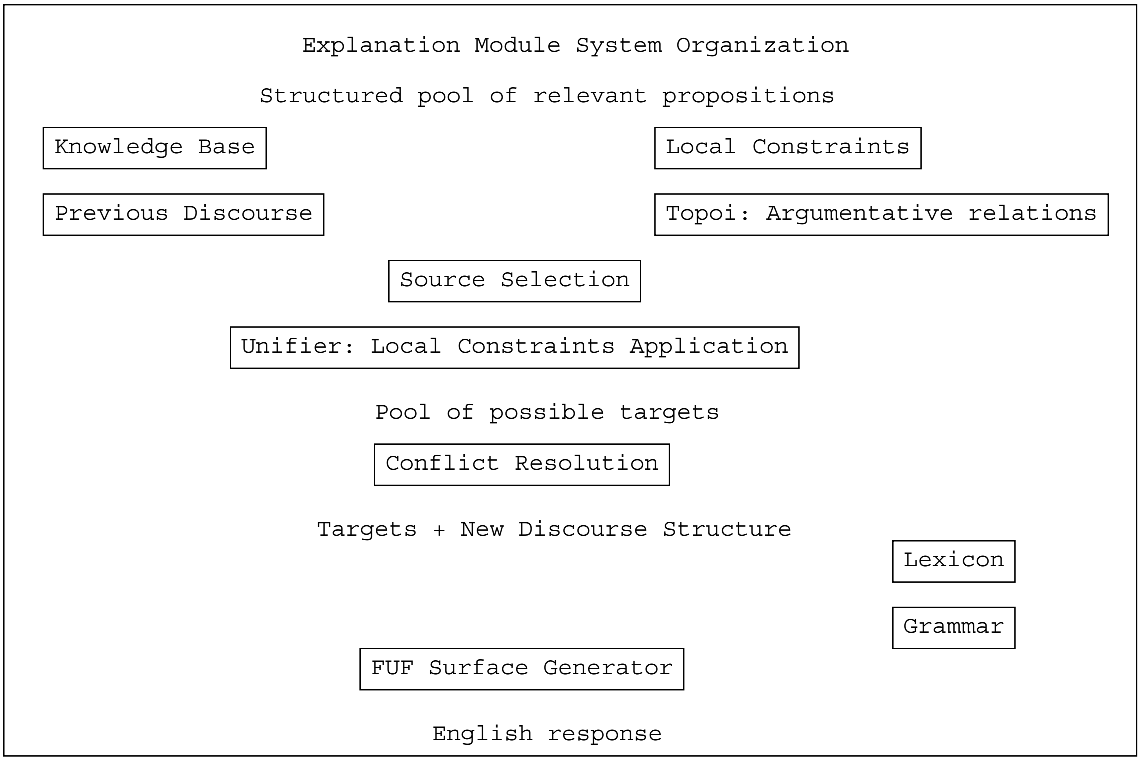
> statement, but because of its argumentative orientation, it can be used to repose the original question, and request a more adequate answer in an indirect fashion.

The criteria needed to fully address these problems are obviously not within the control of the explanation system. I only intend to show how the proposed model of generation can bring an explanation system closer to these goals. To motivate the development of the model described in this paper, I now describe the architecture of the ADVISOR explanation system and define the function of each module.

ADVISOR currently has the following architecture:


ADVISOR System Organization (from [McKeown 88, p.382])

It will be modified so that the expert system now generates a pool of relevant propositions, which is derived from the traced of fired rules. The surface generator will be replaced by the new surface explanation module, which will have the following architecture:



The input of the system is a structured pool of relevant propositions. It has been computed by the expert system, and consists of a tree-structured inference trace. Each node in the tree is a proposition represented in a predicate-arguments form. Another component of the input is a representation of the previous discourse. This representation is also a tree of discourse segments.

The general flow of control in the explanation module is:

1. Find a discourse segment in the previous discourse to which the system is going to react. This segment is called the source. The segment to be produced is called the target.
2. Filter the pool of relevant propositions, to retain the propositions that are connected to the source. Propositions can be connected, for example, when they refer to the same discourse entities or when they support the same conclusion.
3. Map from these propositions to partially lexicalized functional descriptions specifying the discursive function of the potential targets, their illocutionary function and their position in the discourse structure.
4. Determine the target from the set of potential targets (by either choosing the best one or by combining several candidates).
5. Realize the target as an English response, reflecting its discursive function.
6. Update the model of the previous discourse.

Filtering the pool of relevant propositions and mapping the propositions to functional descriptions is controlled by local constraints. Local constraints define what it means for two propositions to be connected and how this connection determines the features of the target. For example, a source and a target can be connected by an argumentative relation. In this case, the mapping from input content to a set of partially lexicalized functional descriptions involves the following steps:

1. Find all scales related to the scale of the argumentative orientation of the source. Call this set the ‘‘relevant scales.’’ Section 6 describes how these argumentative relations can be determined and

defines the term ‘‘scale.’’

2. Map from the relevant scales to a set of relevant concepts which are related to the relevant scales. For example, given the scale `temperature`, the concept `c-temperature` and its specializations in the knowledge base are added to the list of relevant concepts.
3. Filter the pool of relevant propositions to keep only those propositions containing one of the relevant concepts as one of their participants.
4. For each of the relevant propositions, map from the propositional content to an *argument*. This involves mapping from a conceptual scale to a linguistic scale, and making certain lexical decisions (from the concept `temperature-50F`, decide to use the adjective ‘‘cold’’ or ‘‘hot’’ for example). This also determines orientation for the argument. The resulting set of partially lexicalized propositions is the set of relevant arguments.
5. Filter the set of relevant arguments according to the argumentative strategy in use. This means, keep only the arguments that either support or attack the source. This produces the possible argumentative orientations of the target.
6. For each possible argumentative orientation, find a proposition that can express it under the constraints of directness dictated by the interactional local constraints.

The result of this procedure is a set of functional descriptions with a determined argumentative orientation. The argumentative orientation of the target is an important discursive feature, which determines several of the features of its linguistic realization - for example, connectives, adjectives and determiners.

Decisions made on the argumentative dimension also interact with other decisions the system needs to make in order to connect the target move to the previous discourse: for example, does this move close an ongoing exchange, or does it expand it (a structural decision). Similar processes on the other dimensions also determine certain features of the potential targets. The interaction between different classes of local constraints is handled by the LC Application module. The output of this module is a set of potential targets with discursive features set.

The next step in the process is to construct an actual target from this set of potential targets. This is the task of the conflict resolution module. This module can either select one target out of the set or combine several targets together to produce a single complex target. The output of this module is a single target in a format called ‘‘interpretative format’’ (IF), which includes a set of features describing the discursive function of the target (*e.g.*, its argumentative orientation, functional status, illocutionary function).

This target is then sent to the surface generator module, which translates it into English text. Finally, the target is integrated into the model of the previous discourse.

I aim at implementing the system in such a way that example I of Figure 2 at least can be completely generated.

3 Previous Work

3.1 Connected Discourse

This work is a contribution to the problem of generating connected discourse. I first list in this section the techniques that have been used previously to address this problem: discourse grammars, schemas and planning. Note that connected text and planning coherent conversation are different to a large extent. Most of the work reviewed here (except for [Reichman 85] and early work on dialogue games [Levin and Moore 77]) focuses on text and not on conversation. It is still interesting to determine exactly how different are text and conversation, what techniques could be applied to both, and only to conversation.

3.1.1 Discourse Grammars

The notion of discourse grammar is illustrated by [Reichman 85] and [Miller & Rennels 88]. The idea is to represent the connection between the propositions in a text as grammatical relations similar to the relations existing between the phrases of a sentence.

Figure 3: Paragraph-level ATN (from [Miller & Rennels 88, p.43])

Figure 3 illustrates the simplest application of this technique. As can be seen in this example, the grammar can define the structure of a paragraph, produce the connectives and transition phrases necessary to make this structure understood, and call a sentence level grammar to fill in the propositions. This example also shows that such grammars are easy to write, but produce quite stereotyped text, only applicable in very specific situations.

A more ambitious (but less practical) project is described in [Reichman 85]. She describes a grammar of conversation using the ATN formalism. Her grammar identifies the valid moves at a given state in the conversation and maintains a hierarchical representation of the history of the conversation. It is also assumed that it is possible to ‘‘parse’’ a conversation in its constituents. The goal of this parsing is to (1) track the discourse context and (2) ‘‘specify a set of high-level semantic and/or logical constraints that a surface form has to meet in order to fill a certain role at a given point in the conversation’’ [Reichman 85, p.120].

The type of grammatical categories hypothesized in Reichman’s work correspond to possible ‘‘conversational moves’’ such as `indirect-challenge` or `interruption`. In addition to assigning such labels to each turn in the conversation, the discourse grammar is responsible for maintaining a global discourse structure. This discourse structure is made up of ‘‘context space (CS) constituents.’’ Each CS constituent includes information about the type of the constituent (there are 10 possible types, like `evaluative-issue` or `deontic-issue`), the function of the constituent in the conversation (e.g., `support` or `clarification`), the speaker, the status of the constituent (whether it is `active`, `controlling`, `open` or `closed`), and the focus level of the objects and individuals mentioned in the constituent. The structure of the conversation is a stack of CS constituents, and the basic moves altering the structure can be seen as ‘‘push’’ and ‘‘pop’’ on this stack. Using this mechanism, Reichman proposes an explanation for cases of pronominalization and choice of reference.

One of the main problems of this work is that there is no justification given for the different lists of labels used in the structural analysis: the list of possible conversational moves and the list of possible constituent types are arbitrary. More annoying, there is no criteria provided to check for membership: how can one decide whether a move is an `indirect-challenge` or a `emotional-flat-rejection`? It is also unclear why several structural moves which all roughly correspond to either a ‘‘push’’ or a ‘‘pop’’ on the discourse structure are finely distinguished into ten different types of moves: what is gained by this added precision is not specified, and the definition of these finer types is not provided.

At a more abstract level, one may ask whether the goal of developing a grammar of conversation is appropriate. I discuss below some of the problems that make conversation hard to formalize so strictly.

Reichman’s work is interesting in that it proposes a technique to maintain a structured representation of the discourse context. It also indicates ways in which this structured representation can be used to constrain linguistic decisions.

3.1.2 Schemas

[McKeown 85] introduced the use of rhetorical schemas in text generation. The mechanism has also been used and extended by Paris [Paris 87]. Schemas also represent the organization of a paragraph and are implemented with ATNs. Each schema fills a given communicative function (*e.g.*, identification, attribution, definition).

The building units of the schemas are called *rhetorical predicates*. A rhetorical predicate corresponds to a basic communicative function (such as ‘‘illustration’’ or ‘‘comparison’’), and is realized by a single linguistic unit (clause or sentence). Schemas indicate how rhetorical predicates can be composed together to form a paragraph.

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

Figure 4: The Attributive Schema from [McKeown 85][4]

Figure 4 shows the attributive schema. Each of the terms in this schema is realized by a proposition. Each predicate can serve as a knowledge base access function and can identify a proposition that can fill its function. Because schemas rely on the notion of rhetorical predicate, they are more expressive and flexible than the ATN presented in Figure 3. The notion of rhetorical predicate however is not very clear: for one thing, it seems that a notion like ‘‘example’’ is more a relation between two propositions than the type of a single proposition; second, the definition of these predicates is very informal and unprecise. The schema approach suffers therefore of the same unprecision as Reichman’s model. Finally, although they are more general than the fragment of ATN shown from [Miller & Rennels 88], schemas tend to produce stereotyped text. Paris’s mechanism of schema combination helped alleviate this limitation and made the use of schemas more flexible.

---

[4] The notation is: / for alternatives {} for optional items and * for repetition.

3.1.3 Coherence Relations

Coherence relations are very similar to rhetorical predicates. Different models have been described in [Hobbs 85, Hobbs & Agar 85, Mann 84]. Examples of such relations are ‘‘*elaboration*, *example*, *solutionhood*, *enablement*, *motivation* or *justification*’’ - the main difference from the rhetorical predicates is that these are relations between propositions and do not characterize a single proposition.

The model developed in [Mann 84] is called *Rhetorical Structure Theory* (RST). RST also uses schemas to describe the overall organization of a text, but in contrast with the schemas described above, RST supports a more dynamic view: schemas can be constructed by a text planner to satisfy certain communicative goals [Hovy 88].

Each schema is formed of *spans*. A schema consists of *nuclear* and *satellite* spans. *Satellites* are linked to their *nucleus* by one rhetorical relation.

The *Thesis/Antithesis* schema:

1. But I don’t think endorsing a specific nuclear freeze proposal is appropriate for CCC (California Common Cause)
2. We should limit our involvement in defense and weaponry to matters of process, such as exposing the weapons industry’s influence on the political process.

```
Thesis/Antithesis          The Inform schema:
     |
     |<--thesis/anti--+                        Inform
     |                |                          |
1:---+---        2:---+---              +--------+----------+
                                        |        |          |
                                    Background   |   Elaboration
                                        |        |          |
                                     ---+---  ---+---  3:---+---
                                              Conditional
                                                 |
                                                 +-Condition-+
                                                 |           |
                                             1:---+---   2:---+---
```

Figure 5: Mann’s Rhetorical Schemas

Figure 5 illustrates the use of the RST formalism. Each span corresponds to a proposition in a text (referred to by numbers). Spans are related by semantic relations to the nucleus of the schemas. Because schemas can be recursively composed (a satellite can be a complete schema), the representation can be used for large segments of text. The list of such labels (about 20 in the current version of RST) is claimed to cover all possible relations in discourse. Once again, as for Reichman’s work, the informal nature of these relations makes the whole theory rather weak. A formal definition of the RST relations is proposed in [Hovy 88], but its usefulness is questionable (cf. below).

Some additional problems with the RST approach are discussed in section 5.1 (mostly that the tree structure of the schemas is often too inflexible). But the main problem is that RST is a mono-dimensional approach: it tries to explain the connection between an utterance and the rest of the discourse by giving it a unique label. I claim, in contrast, that the connection must be described along several dimensions corresponding to independent theories of human action.

3.1.4 Speech-Act Networks

Speech-act theory has developed the notion that utterances are actions that have a purpose and are about a proposition. Researchers have identified the conditions of use for a wide variety of speech-acts (*e.g.,* [Searle & Vanderveken 85]). Only recently, the issue of sequencing speech-acts and the relation between speech-acts and discourse have become active research problems. The problem is to identify what makes a string of speech-acts coherent (cf. for example [Taylor and Cameron 87, pp.58-63] on the problems of speech-act sequencing).

One solution, proposed in [Winograd & Flores 86], is to consider a complex network of possible speech-act moves, presenting at each transition the subset of the possible moves a participant in conversation can perform. Each move correspond to a speech-act.

Figure 6: A Speech-act network (from [Winograd & Flores 86])

Figure 6 illustrates the approach. There are several problems with this global approach to speech-act sequencing, and [Bowers and Churcher 88, pp.126-129] present a convincing criticism of this work.

3.1.5 Planning

Another approach to the problem of speech-act sequencing based on the technique of planning has been developed within the AI community. [Allen & Perrault 80] have introduced the idea that speech-acts, being actions, can be represented as planning operators - with preconditions and effects - and a sequence of such operators can be built by a planner to satisfy a given goal. In text generation, such goals can be communicative goals like ‘‘refer to an object,’’ ‘‘inform’’ or ‘‘convince.’’

[Appelt 85] has first shown that the approach of planning can be used at all levels of text generation. His system, KAMP, is capable of generating an utterance given a goal to satisfy. KAMP’s reasoning system is based on possible-world semantics which allows it to reason about the beliefs and intentions of the speaker and the hearer. KAMP includes a planner that can extend an input goal into a sequence of communicative acts. Appelt distinguished between acts of different levels: illocutionary acts, surface speech acts, concept activation acts and utterance acts. The planner includes an interesting technique of critics: critics examine a proposed plan for global interactions between substeps and can optimize the plan. In particular, the *action subsumption* critic recognizes situations where one action A_1 produces all the effects of an action A_2 - and therefore, if A_1 is performed, A_2 does not need to be realized anymore. When planning the realization of a referring expression for example, this technique allows KAMP to opportunistically add modifiers to a noun phrase to satisfy additional goals. For example, instead of producing the sequence ‘‘take the box. The box is blue,’’ KAMP would produce the single utterance ‘‘take the blue box.’’ Another interesting contribution of KAMP was the integration of a unification based formalism for the realization of surface linguistic goals with the planning system in the TELEGRAM grammar. KAMP was the first system to integrate the notion of intention in text generation and established planning as a technique to take goals and beliefs into account in the process of text generation.

Later work extends the planning formalism to connected text. [Hovy 88] has developed a text structurer that takes as input a set of propositions and a communicative goal and produces an RST tree that can satisfy the communicative goal. Each RST relation is described in this work as a planning operator, with preconditions and effects expressed in a language describing beliefs of the speaker and hearer. A similar approach is used in [Moore & Paris 89].

```
Purpose
Nucleus Constraints:
1. (BMB S H (ACTION ?act-1))
2. (BMB S H (ACTOR ?act-1 ?agt-1))
Satellite Constraints:
1. (BMB S H (STATE ?state-1))
2. (BMB S H (GOAL  ?agt-1 ?state-1))
3. (BMB S H (RESULT ?act-1 ?act-2))
4. (BMB S H (OBJ ?act-2 ?state-1))
Intended Effects:
1. (BMB S H (BEL ?agt-1 (RESULT ?act-1 ?state-1)))
2. (BMB S H (PURPOSE ?act-1 ?state-1))[5]
```

Figure 7: Planning definition of the ‘Purpose’ RST relation (from [Hovy 88])

Figure 7 shows the definition of the `Purpose` RST relation. If two propositions can be found that satisfy the nucleus and satellite constraints, then they can be related by a `purpose` relation in the output paragraph. The relations in the RST tree produced are used to generate appropriate connectives between the propositions. Note that this definition relies on some domain predicates (like `RESULT` or `PURPOSE`) which makes it a little less general than one would wish: if the ‘‘meaning’’ of the RST relation purpose is to express that there exists a purpose between an act and a result, the definition does not provide much information. As aptly stated by Hovy himself, ‘‘who can hope to represent something like motivation or justification complete with all ramifications?’’ [Hovy 88, p168]. Nevertheless, the proposed mechanism seems promising. Note also that, as mentioned in [Moore & Paris 89], separating a text structurer from the module in charge of selecting the propositions can be a problem: if the pool of propositions is not well formed, it may be possible that none of the defined RST relations can be matched by existing relations. In this case, some arbitrary order must be chosen to express the propositions. [Moore & Paris 89] actually do combine the content selection module with the text structurer module.

[Woolf 84] presents a planning system that can satisfy communicative goals in a tutoring system. In this context, the communicative goals are called *tutoring goals*. Note that this planner handles conversation as opposed to a single paragraph, and that it makes use of the discourse history in its decisions. The planner has three levels: a ‘‘pedagogic component’’ determines a global ‘‘pedagogy’’ specifying how often the tutor can interrupt the student, ask questions or provide answers; next, this pedagogy is translated into more specific schemas indicating for example that the goal is to question the student or to provide examples; finally, specific types of questions or of

---

[5] (BMB x y p) means that p follows from x’s beliefs about x and y mutual beliefs; (GOAL x p) that p follows from x’s goals; (BEL x p) that p follows from x’s beliefs.

examples are chosen, depending on the discourse history and on the student model. No surface generator has been used in this system. These three levels are implemented with an ATN formalism that allows for contextual jumps - triggered by ‘‘meta-rules.’’ The primitives of the system (the units out of which the structure is built) are a rather ad-hoc list of utterance types - similar in spirit to rhetorical predicates. They include for example ‘‘exploratory question,’’ ‘‘question reverse dependency’’ or ‘‘implicit incorrect acknowledgment.’’ Here again, as in Reichman’s case, the weak point of the system is in the fuzziness of this list. In addition, since the output of the generator is an internal form made up of these labels, it is hard to judge the quality of the intended generated text.

[Dale 88] presents a system that can produce the text for recipes. Although the focus of the work is on producing referring expressions, it also includes a planning component. This component has roughly two levels: in the first stage, a domain plan is built - with each preparation step that the user may need (a user model describes what operation are known to the user); in a second stage, this domain plan is ‘‘massaged’’ by discourse procedures to make it more fluent (for example, include a coordination between consecutive propositions using the same predicate and agent and avoid the repetition of the verb). Note here that the goal provided to the system is always to describe the instructions for some meal and that the planner is only a domain planner - it does not reason about the communicative process. This planner produces the set of propositions to express and follows the structure of the task to structure the paragraph. Even with these simple assumptions, the fact that the ‘‘massaging’’ stage is necessary is very important: it shows that there exist some local constraints between consecutive propositions that must be satisfied to make their concatenation linguistically acceptable.

3.2 Local Management of Coherence

The focus of this work is on local constraints on discourse production. I discuss in this section the distinction between a local and a global approach to coherence in discourse and present an example of the use of local constraints in a generation system.

3.2.1 Local vs. Global

In general, it is easier to describe the structure of text globally. The schemas described above are an example of this trend. Designing a schema for a given domain and a given communicative function is rather easy, when using an appropriate formalism, and can quickly produce coherent texts. Unfortunately, this technique quickly reaches practical limits: the texts produced are quite stereotyped, and most importantly, the schema does not provide enough guidance to control the low-level decisions that must be made by the surface generator (for example, pronominalization or the choice of definite vs. indefinite reference). Schemas provide a good technique to describe the high-level progression of a text, but practice shows that there exists a gap between the indications a schema can produce and the specifications a surface generator requires.

Besides their practical appeal, schema-like approaches and global descriptions of coherence also help solve certain theoretical problems. Long distance references, even if they are rare (probably less than 10%), must be explained: which entities are accessible in the previous discourse and can be referred to using a pronoun, even several utterances after they have been mentioned? The notion of global focus [Grosz 77] contributes to such an explanation. That there exists an agenda for certain types of interaction (formal meetings) can also be explained by a global model only. Finally, the notion of insertion sequence [Schegloff 72] also supports the idea that the structure of the conversation must be monitored to contribute to conversation. The following text is an example of insertion sequence:

[Merritt 76,p.333] and [Levinson 83, p.304]:

A:	May I have a bottle of Mich?	(Q1)
B:	Are you twenty one?	(Q2)
A:	No	(A2)
B:	No	(A1)

Insertion sequences are quite frequent in conversations, and correspond to a structural organization of the type [Q1 [Q2 A2] A1].[6] Since such insertion sequences can presumably be recursively embedded to some depth, it is necessary to represent this embedding properly to contribute to conversation.

Global approaches are therefore appealing because they are easy to implement and help explain phenomena such as long distance reference, ritual or agenda-based interactions and insertion sequences.

There are also, however, theoretical reasons to demand a more local description of coherence than schemas can provide, at least for conversation. Work in conversation analysis [Levinson 83, Atkinson & Heritage 84, Sacks, Schegloff & Jefferson 78] has emphasized the local nature of many conversational phenomena. The first and classic example of this approach is the analysis of turn taking in conversation. Analysis of transcripts of conversations indicates that some mechanism allows people to take the floor without visible problems: ‘‘less than 5% of the speech stream is delivered in overlap (two speakers speaking simultaneously), yet gaps between one person speaking and another starting are frequently measurable in just a few micro-seconds’’ [Levinson 83, p297]. Such efficiency is rather surprising since by many other aspects, conversation is not a precise and robust mechanics. In [Sacks, Schegloff & Jefferson 78], it was proposed a set of four rules explaining how turn taking operate. The most important characteristics of this model is that it is *locally managed*: turn allocation is decided one turn at a time, and applies only to the next turn. This approach should be contrasted with another type of model, where allocation of turns is decided in advance for a whole segment of conversation - for example, a model specifying that turns should alternate between two participants (cf [Levinson 83, p.296].

The notions of *uptake* in speech-act theory [Austin 62] and the possibility of *reinterpretation* of previous conversational material [Fox 87, Roulet *et al* 85] also hint at the difficulty of designing a schema-like approach to conversation, and that it is difficult to even only use some of the techniques used for connected text to produce coherent conversations. By uptake, Austin referred to the process by which a hearer makes clear to a speaker that he has recognized the speech-act he intended to perform. Until the uptake, a speech-act cannot be complete. For example, after an utterance ‘‘I bet you $5 it’s going to rain,’’ an answer of ‘‘ok’’ would be the uptake for a betting speech-act (the bet is on); in contrast, an answer like ‘‘you are a joker’’ does not ratify the betting act. No bet can be said to have been performed, even though the speaker had all the right intentions. This notion embodies the cooperative aspect of conversation: performing a speech-act requires the appropriate production of an utterance and an appropriate reaction by a hearer. Because this reaction cannot be reliably controlled *a priori*, it is hard to make more than very local predictions on speech-act sequencing (cf. [Bowers and Churcher 88, pp.126-129] for a discussion of this point).

Instead of the notion of transition network presented above, conversation analysts use the notion of adjacency pair [Levinson 83, p.303]. An adjacency pair (AP) is in some way the smallest structural element one can use. The idea of APs is that issuing the first part of the pair (for example a request for information) makes the second part

---

[6] Whether the structure is [Q1 [Q2 A2] A1] or [[Q1 [Q2 A2]] A1] is not clear. In the second structure, we would consider the complex [Q2 A2] as a way of precising the question Q1, and the whole segment [Q1 [Q2 A2]] would be the question to which A1 is an answer. This analysis would be similar to a case like ‘‘A: I’m new here, is there a way to the top’’ where the first clause serves as a preparation to the question, and the whole segment is a complex question. This example illustrates why in general designing a global structure is difficult: I know that Q2 and A2 are related, I also know that A1 is an answer to Q1. I have no reasonable way to decide what is the connection between the pairs Q1-A1 and Q2-A2.

expectable (for example, providing information). It does not mean that the second part must follow the first part. What it means is that if the subsequent turn is not of the expected type, the violation of this expectation is meaningful.

In [Fox 87] and [Roulet *et al* 85], the notion of ‘‘re-construction’’ or reinterpretation is introduced. It refers to cases where in a subsequent utterance a participant reconstructs the meaning of a previous utterance - as in the following example:

[Roulet *et al* 85,p.227]:

A1:	Do you know the famous pianist Paderewski?
B1:	Paderewski?
A2:	Yes
B2:	Of course
A3:	Well, he plays even better than I.

Here, A3 reconstructs the meaning of the whole exchange A1-B2 as a preparation for an assertion, whereas it was at first built as an exchange of information. This notion illustrates the point that the relation between previous discourse and a new turn can be bidirectional: previous discourse constrains the interpretation of the following discourse, but it is also possible for the following discourse to modify the interpretation of a previous segment. It is therefore very hard to make reliable predictions on how the total structure of the conversation will evolve given the beginning of the conversation.

The notions of uptake and reinterpretation thus favor the conclusion that a global model of conversation is hard to design. They are both specific to conversation. The problem of combining local and global approaches is therefore more accute for conversation. In general, there seems to exist a spectrum of different modes of interaction, from complete ritual (as a religious ceremony), where global predictions are completely reliable, to completely spontaneous (as unplanned chatting), where any global prediction (more than a few utterances) is hard to sustain. When global predictions are not sufficient, local constraints provide a way to describe a type of coherence, sufficient to control the linguistic devices depending on discourse level phenomena (pronominalization or ellipsis for example).

Except for the study of local focus, such local coherence phenomena have not received a lot of attention. In this work, I focus on local constraints and aim at evaluating the range of linguistic phenomena that such a local approach can motivate. Describing how global predictions and local constraints can interact will not be addressed directly in this work.

3.2.2 Example of Local Constraints

Certain classes of local constraints have been used in previous work and are also used in this work. The notion of local or immediate focus, introduced in [Grosz 77], developed in [Sidner 79], under the name of centering in [Grosz, Joshi & Weinstein 83, Joshi & Weinstein 81] and used for generation in [McKeown 85], is a good example. The idea is that some general focus progression rules constrain what can be the focus of each successive proposition. In terms of centering, every utterance is assumed to have a *backward looking center* and a set of *forward looking centers*. A following utterance can either maintain the backward center, shift to a new backward center taken from the forward centers of the preceding utterance, or completely shift to new centers.

In the TEXT generation system, these local focus rules were used to (1) make decisions that were left free by the schema (what proposition from the pool of matching propositions should come next?) and (2) control certain syntactic decisions (what role in the proposition should occur first? use passive or active voice?).

This shows the role of one type of local constraint in a large generation system: it complements a global mechanism and it serves as an intermediary with the surface component to control surface decisions.

3.3 Multi-Dimensional Aspects of Discourse

A major claim of this work is that several classes of local constraints interact to determine the form of a new turn in conversation. In this subsection, I review work in discourse theory that similarly relies on a multi-dimensional description of discourse.

3.3.1 The Three Systemic Meta-Functions

The systemic school of linguistics aims at providing a functional description of language: the goal is to explain how one uses different linguistic devices to realize certain goals - or in other words, to describe the function of each linguistic device.

In [Halliday 85], the distinction between three meta-functions is presented. The NIGEL text generation system uses this theory. The meta-functions can be defined as follows (cf. [Matthiessen 88, p.3]):

- The *interpersonal* meta-function is concerned with establishing and maintaining the interaction between the speaker and listener.
- The *ideational* meta-function is concerned with ‘‘ideation’’ - with the interpretation and representation of our experience of the world around us and the world of our consciousness.
- The *textual* meta-function is concerned with the ongoing presentation of interpersonal and ideational information as contextualized text.

The role of the meta-functions is to partition the set of decisions one must make to generate a sentence into distinct sets depending on the type of knowledge needed to make the decisions. In NIGEL, the environment of the text generator is accordingly separated into three ‘‘bases:’’ a knowledge base, an interaction base and a text base.

In the organization of the theory therefore, the distinction into three meta-functions plays an important organizational role: it makes it possible to sort out the type of influence exercised by the situation on the production of text. Each meta-function corresponds to a simple consistent theory. The main gain is actually that each theory can remain simple: for example, whereas standard speech act theories propose very complex classifications of speech-acts (e.g., [Searle 75]), the interaction theory in the systemic framework proposes an extremely simple classification of speech-act: giving or demanding, goods and services or information [Halliday 85, p.69]. This approach is judged sufficient because this interpersonal function can be supplemented by an ideational function, and the combined specifications can cover the whole range of phenomena that what systemists would call an ‘‘overly ambitious’’ theory, interpersonal only, would try to cover.

3.3.2 Intention, Attention and Discourse Structure

Grosz and Sidner have outlined a theory of discourse structure that also makes use of a three level perspective [Grosz & Sidner 86]. The theory describes discourse as the combination of three structures:

- a *linguistic* structure, which is a tree-like structure of text segments. Segments are connected by the relations of *dominance* (S1 embeds S2) and linear precedence (S1 occurs before S2).
- an *attentional* structure, which is a stack-like structure of the entities referred to in the discourse. The higher an entity is on the stack, the more recently it is supposed to have been under the attention of the hearer.
- an *intentional* structure, which keeps track of the relations between the purposes of each discourse segment (the purposes are related to the plans and intentions of the speaker). Discourse purposes are connected by the relations of *dominance* (DP1 is a sub-goal of DP2) and *satisfaction-precedence* (DP1 is a pre-requisite for DP2).

Here again, the multi-dimensional perspective on discourse permits a simple and homogeneous definition for each dimension. For example, the types of relations between discourse segments described in the theory are only `dominate` (A dom B if B ‘‘depends’’ on A in some fashion; for example if B is an illustration of A) and `precede` (which is just the linear precedence relation). This simple definition can advantageously be contrasted with the dozen of relations proposed in the mono-dimensional model of [Reichman 85] for example.

These two examples illustrate the main advantage of using a multi-dimensional approach to discourse: discourse is the locus of many interacting influences; looking at ‘‘one thing at a time’’ is a wise heuristic to avoid building excessive complexity into its analysis.

Once this methodological approach is accepted, there are two problems that need to be addressed: (1) what classification of dimensions should be used, and how can it be justified (must the dimensions be independent in some way?), and (2) how do the different constraints interact to determine the production of the text. These are the main questions I address in this work.

In this work, I use a different classification of constraints from both Grosz and Sidner’s and Halliday’s. The classification I use is presented in detail in Section 5.2; it has the following five dimensions: structure of the discourse, topic relations, illocutionary constraints, interactional constraints and argumentative constraints. It draws on both approaches presented here: the structure of discourse is included in Halliday’s textual function and corresponds closely to Grosz and Sidner’s discourse structure; topic relations correspond closely to Grosz and Sidner’s attentional structure; in Halliday’s system, topic relations depend on both the ideational and the textual functions; Grosz and Sidner’s intentional structure covers all of the argumentative, interactional and illocutionary constraints. Comparing Grosz and Sidner’s classification and Halliday’s system helps understand why I chose to use a finer classification: G&S’s intentional structure corresponds to aspects of the interpersonal function (when the discourse purposes involve operators like `Know` or `Believe` or `Do`, *e.g.*, `Goal(Speaker, Do(Hearer, Act))`), and to aspects of the ideational function (when the discourse purpose is a domain goal, *e.g.*, `Goal(Speaker, Attached(Pump, Compressor))`. Halliday’s grammatical description of English and extensive work on the notion of politeness for example [Brown & Levinson 87] indicate that the notion of interpersonal constraints is useful for describing many linguistic phenomena. The debate underlying this distinction is whether these interpersonal constraints are primitive or can be derived from more basic principles (*e.g.*, the principle of rationality underlying Grice’s maxims or the planning paradigm). I have chosen to view them as primitive or at least ritualized - and therefore unpredictable from an application of basic principles. This explains why I do not retain the notion of intentional structure in its full generality in my classification. Rather, I propose to distinguish three more specialized forms of ‘‘intentions:’’ the interactional dimension corresponds to intentions involving the social relations between speaker and audience (it corresponds most closely to Halliday’s interpersonal function); the illocutionary dimension captures the conventional ways of acting with language, and the conventional relations between speech acts (it corresponds most closely to G&S’s intentional structure); the argumentative dimension describes the attitude of a speaker regarding the utterances he produces, and a form of domain specific relations between discourse segments.

3.4 Argumentation

Argumentation is viewed in this work as one principle that guides language production. The idea is that often facts and propositions are not uttered for their own sake, but rather are used because of their relation to other facts and propositions: an argument is used to support or oppose a conclusion. The conclusion needs not be explicit. In this section, I review work that has used this assumption to help analyze discourse. As discussed in section 1.3, there are two ways to use argumentation in a discourse theory: conceptual and linguistic. In this section, I first review work done in the conceptual use of argumentation, particularly the system of argument understanding developed by

Flowers, McGuire and Birnbaum and the work developed by Sycara which could be used in a generation system. Then, I discuss the work of Robin Cohen which relates discourse structure to argumentation and investigates what effects argumentation has on clue word selection and interpretation. Finally, I briefly discuss the theory of linguistic argumentation put forward in [Anscombre & Ducrot 83].

3.4.1 Conceptual Argumentation

By conceptual argumentation, I mean the study of argument comprehension as a knowledge-based activity. The goal of the study of conceptual argumentation is to find propositions that can help modify a hearer's beliefs and behavior concerning a given issue.

In [Flowers *et al* 82, Birnbaum 82, McGuire *et al* 81] the notions of argument molecules and networks of arguments have been introduced. A technique similar to argument networks has also been used in many different systems aiming at either supporting the process of argumentation [Conklin and Begeman 88] or at generating arguments (for example, [Sycara 89]). Typically, arguments, issues, positions and conclusions are represented as nodes in a conceptual network. These nodes are linked by relations such as `attack` and `support`.

To build an *argument graph*, one must face the following problems: how do you recognize that a fact is an argument for or against another fact, and how do you find facts to serve as arguments for or against other facts.

[Flowers *et al* 82] propose using an ‘‘inferential memory’’ to explain these phenomena, where the process of ‘‘reminding’’ about an event also uncovers related events. This operation is extremely knowledge intensive. The argument graph is then used to reason about the conversation structure, and about the beliefs of each participant.

[Sycara 89] proposed using the notion of utility from decision theory, and classified types of argumentative moves accordingly. Utility allows her to link agents's beliefs to their decision making mechanisms and their actions. Her work is particularly relevant for generation, as she addresses the problem of generating arguments - not just recognizing argumentative relations between propositions. She presents techniques to generate arguments to change the hearer's perception of the importance of an issue, to change the perception of an issue's value, or to generate threatening arguments. Since her model relies on a model of decision making, she can effectively reason about which beliefs of an agent need to be changed to modify his behavior.

Note that the work on conceptual argumentation does not rely on linguistic information: it does not help in expressing arguments, and does not try to use linguistic clues to recognize argumentative relations.

3.4.2 Structure Markers and Argumentation

Robin Cohen has worked on one important way to relate the form of a text to its argumentative structure [Cohen 83, Cohen 84, Cohen 87]. Assuming that the structure of the argument graph is a tree, where an issue node has positive or negative arguments as children, Cohen proposed that the position of a turn within the argument structure is often marked by the use of special structure markers. She particularly looked at the role of clue words - e.g., ‘‘in addition, first, second, therefore, in sum.’’ The assumption is that each clue word gives an indication of the argumentative role of an utterance in the structure of the discourse. Another interesting finding, is that the structure of certain texts follows the structure of the argumentation they present. I will use these findings in describing the interaction between structural and argumentative constraints.

3.4.3 Argumentation in Language

In [Anscombre & Ducrot 83] a theory of linguistic argumentation is presented that goes further towards integrating linguistic form and argumentation. The point of the theory is that argumentation is built ‘‘into’’ the language we speak: the surface form of an utterance can have an influence on the conclusions it can support, regardless of the information it conveys and of the knowledge of the hearer. Consider the following example:

> I know Jack will pass his exam, because he is smart.
>
> *To illustrate the inadequacies of a particular exam:*
> Take Jack, for example. Since he is smart, he failed the exam.

The same fact ‘‘he is smart’’ is used as an argument for the conclusion ‘‘he will pass his exam’’ or ‘‘he failed the exam.’’ Thus, the fact that a proposition P is an argument for or against a proposition Q is not a property of the information conveyed by P and Q, nor can it be derived from the knowledge of the hearer alone, given this information. Therefore, the theory concludes, the form of an expression constrains how it can be used to function as an argument - independently of the type of information it conveys.

Because it is possible to present the same information as an argument for or against the same conclusion, Ducrot and Anscombre argue that the argumentative intention of a speaker is a primitive of the semantic description of an utterance - in addition to the information conveyed. For text generation, this means that certain argumentative directives must be present in the input to the surface generator to properly choose to use certain linguistic devices (*e.g.*, connectives).

In this work, I focus mainly on the linguistic aspect of argumentation, and use results from the Ducrot school. This school is mainly developing linguistic analyses, but recent work has started a more computational approach [Raccah 87].

4 Scope of the Work

One of the goals of this work is to evaluate the effectiveness of the notion of local constraints to do text generation in context. Given that purpose, I propose to concentrate on a small set of decisions that are part of text generation and show how these decisions can be handled in the proposed architecture. In this section, I list the decisions on which I will focus. I do not explain here how each decision will be made, but just what decisions will be made. The list covers a wide range of phenomena in order to indicate the global relevance of the approach to text generation. Most of these phenomena have not been addressed before as the main focus of text generation systems.

4.1 Content Filtering

The problem of content selection is to decide what information should be conveyed at a given point of the conversation. I will address a small aspect of this full problem.

I consider a system where a set of ‘‘mentionable items’’ (the term is used in the conversation analysis literature, e.g, [Sacks, Schegloff & Jefferson 78]) is provided to the text generator in advance. These items are represented as propositional forms in some knowledge representation system. In the implementation for example, I use the LOOM knowledge representation system [Mac Gregor & Brill 89]. A proposition can conceptually be thought of as a predicate-argument structure `P(x,y...,t)`.[7] The problem in content selection that I will address is the following: at a given point of the conversation, what proposition of the set of mentionable items needs to be conveyed in

---

[7] When no more detail is needed, I will represent most of the knowledge representation structures in the predicate-argument notation. This notation does not correspond to the LOOM notation but can be interpreted easily.

the next turn. It is therefore a problem of content filtering.

I use mainly topical and argumentative constraints to address this problem. Note that I assume here that another mechanism is in charge of constructing the set of mentionable items in advance. This task requires a global perspective on the conversation, and needs a definition of the function of the conversation in a larger framework of activities. I focus in this work only on the local aspect of content selection. This distinction is similar to the local vs. global focus distinction proposed in [Grosz 77]. Note also that the expected length of the contribution is a factor that can determine the appropriateness of this method: if a speaker intends to produce a long turn (*e.g.*, more than four or five clauses), then the turn probably needs to be globally planned. The decision to produce a long turn is itself difficult to make. I will not address this problem further in this work.

4.2 Content Organization

Once a proposition has been selected, the text generator must decide how to ‘‘attach’’ it to the body of the ongoing conversation - where to put the new turn in the structure of the conversation. This decision determines (indirectly) whether clauses will be coordinated or subordinated.

4.3 Implicit Content

Language provides tools to convey certain information implicitly. For example, in the sentence ‘‘John stopped smoking’’ the fact that John used to smoke is conveyed implicitly. In ‘‘Peter drinks even at night,’’ the fact that Peter drinks at other moments of the day is made implicit (cf [Nolke 83, Kay 87] for analyses of ‘‘even’’).

For some propositions that have been selected, the text generator may decide to use an implicit mode of communication. I will look at a subset of the techniques available in language to convey information implicitly based on a linguistic theory of argumentation, such as use of connectives and argumentative operators (*e.g.*, ‘‘even’’).

4.4 Choice of Connectives

Connectives are words like ‘‘and, but, although, since, because’’ which are used to express the relation existing between clauses in the text. In the framework proposed in this work, I propose using local constraints as a procedure for connective selection [Elhadad & McKeown 89, Elhadad & McKeown 90, McKeown&Elhadad 90].

I specifically propose to distinguish between connectives having a very close semantic function - for example between ‘‘but’’ and ‘‘although’’ or between ‘‘because’’ and ‘‘since.’’ The goal is to show that a multi-dimensional description of the relation between clauses is necessary to make such fine distinctions possible.

4.5 Choice of Determiners

The determiner sequence of a noun phrase is a very complex syntactic structure (e.g., ‘‘almost two thirds of the first ten participants’’). The decisions needed to construct the determiner sequence of a noun phrase involve: (1) the choice definite vs. indefinite, (2) the distinction mass vs. countable, (3) the expression of quantity or number (with the problems of collective and partitive expressions), (4) quantification.

I propose to show how topic and argumentative local constraints can be used to determine the selection of the determiner sequence. In particular, I will focus on the problems of gradability in the determiner.

4.6 Gradable Open Class Lexical Choice

Lexical choice refers to the problem of selecting appropriate lexical items to refer to conceptual structures. In contrast to open class words, closed class words are categories of words with a specific syntactic function, such as conjunctions or auxiliaries. The selection of closed class words is generally seen as a syntactic task. When the selection of open class words should be done in the process of text generation is less clear.

For example, if the conceptual model of the underlying system contains an entry used to describe events involving an exchange of property, called `C-exchange`, the problem is to choose a word among ‘‘give, take, sell, buy, steal, ...’’ If the domain contains a finer description of processes, with conceptual entries like `C-give` and `C-buy`, it may still be possible to use words like ‘‘give’’ to express a `C-buy` event in certain conditions (for example, to highlight the feeling that the price paid is excessively low).[8]

Gradable lexical items are words that can express gradual meaning. Many adjectives are gradable [Huebler 83 2.1.2.2, Bolinger 67, Bolinger 72]: expensive and cheap, hot and cold, high, big all refer to a scale or measure. The same gradability phenomenon exists for verbs and nouns: run, walk and rush seem to be different degrees on some scale.

I propose to use local constraints to decide on the selection of gradable lexical items. I will first only work on adjectives.

5 Local Constraints in Generation

This section first defines the notion of local constraint as I will use it. Then, I present the classification of local constraints used in this work. Finally, I describe each type of local constraint in more detail, with the exception of the argumentative constraints, which are presented separately as the focus of the next section.

5.1 Definition of Local Constraints

I define a local constraint as a relation between two discourse segments. A local constraint explains the coherence between these segments on a single dimension. The first part of the pair is called the *source* of the relation, the second the *target*. The relation between source and target is bidirectional: on one hand, the target is produced under the influence of the source; on the other hand, the target contributes to the interpretation of the source, and completes its meaning.

For example, one local constraint specifies that the discourse entities under focus in the source and in the target must be connected. A specialization of this constraint specifies that the focus can remain the same in the source and in the target.

Figure 8 shows one way to represent this constraint. Another local constraint specifies that after an offer speech act in the source segment, the target must realize either an acceptance or a refusal speech-act (cf. Figure 9). Note that these two examples are in a sense ‘‘orthogonal’’ - they specify different aspects of both the source and the target, and put constraints on different features (focus vs. speech-act).

Local constraints are represented uniformly by a pair of partial specifications - one for the source and one for the

---

[8] Note that the issue of lexical choice is very much linked to the knowledge representation decisions made in the underlying system. I take a rather neutral approach and do not question the way the domain is represented in the underlying system, but eventually, lexical choice and domain modeling should be worked out together and in a complementary way.

```
((cat local-constraint)
 (source ((focus F)))
 (target ((focus F))))
```

[9]

Figure 8: Representation of the ‘‘maintain focus’’ local constraint

```
((cat local-constraint)
 (source ((speech-act ((type Offer)))))
 (target ((speech-act ((type {alt (Accept Reject)})))))))
```

[10]

Figure 9: Representation of the Offer/Accept local constraint

target. These specifications both determine whether the local constraint is applicable in a given case and how it must be enforced when it is.

An important claim of this work is that *several* local constraints together explain the coherence relation between the source and the target. Previous analyses of coherence used lists of ‘‘rhetorical relations’’ (presented in Section 3.1.3) and explain coherence as the instantiation of one of these relations. Some of the problems with such an approach are:

- An objective definition is not provided for these relations; so far the labels used to refer to these relations have been arbitrary and one must rely on the judgment of a human analyzer to categorize a relation.
- Because these relations do not rely on some underlying theory of communication, the list of relations is open-ended. There is no criterion to decide of its completeness. Some empirical study can be done, but it is hard to imagine how a given list of rhetorical relations can be validated and declared ‘‘complete’’ for a given domain.
- The existence of a rhetorical relation between two segments does not offer any clue as to how the linguistic form of the segments is affected.
- It is impossible to predict if and how such relations can be ‘‘composed’’ - is it possible that a segment be at the same time a question and an example? (probably not), an example and an explanation? (probably yes, but why?).
- As a consequence of this ‘‘non-compositionality,’’ it is hard to envisage cases where one segment is related to several other segments by several rhetorical relations (either one source is satisfied by several targets (S, T1) (S, T2) or one target fills several sources (S1,T) (S2,T)).

The local constraint approach does not face the same problems. Each local constraint is defined as a relation between features of discourse segments. Of course, this approach puts the burden of the ‘‘definition’’ on the definition of these features. But here, the task is somehow easier, because, ideally, these features must correspond to other aspects of the system (*e.g.,* syntactic properties, semantic or pragmatic features used by the reasoning component of the system). I will have to show that the features I use in the definition of the local constraints can thus be justified.

Moreover, because the definition of the local constraints is rooted in the definition of features of each segment, and the form of the source and of the target is partially specified by these features, the existence of a local constraint is related to the form of the related segments.

Finally, and most importantly, the local constraint approach is compositional, in the sense that compatible constraints can be combined; it is actually the interaction between several local constraints, along several dimensions, which determines the coherence relation between the discourse segments.

To illustrate the importance of this point, consider the example in Figure 10. In this example, the following segments are related by some local constraints:

- (1,2): coordination of questions. 2 is asked with the implicit header ‘‘if yes then.’’
- (3,4): contradiction (expressed by ‘‘but’’).
- (1,3): 3 is an answer to 1.
- (2,4): 4 is an answer to 2.

1. Can I take Data Structures,
2. and will it allow me to graduate in time?

3. You could take it
4. but you should take fundamental algorithms first.

Figure 10: A Complicated Structure

Now let’s consider the form of 3 and hypothesize an explanation for it. Because of the (1,3) relation, it is possible to use the pronoun ‘‘it’’ in 3; because of the relation with 4, the modal ‘‘could’’ is used, instead of ‘‘can.’’ Note here that two distinct sets of constraints are superposed on a single utterance T, in a {(S1,T) (T,S2)} configuration. Such a relation can only be explained because the constraints on T can be combined together. The inclusion of a segment in a larger text does not need to be described as a single link to a tree-like structure (*e.g.*, an RST structure), but can be described as filling constraints from various sources in the text.[11] An even more complex example is discussed in [Levinson 83p.290] and shown in Figure 11. Here, the utterance A functions as both a question and an offer. A compositional approach is necessary to explain how these two functions can be conflated in a single utterance (this is similar to Appelt’s action subsumption critic discussed in Section 3.1.5), and similarly, how constraints with source in A1 and B2 together determine B3’s form.

(From [Levinson 83, p290]):
A1: Would you like another drink?
B2: Yes I would, thank you,
B3: but make it a small one.

Figure 11: An example of utterance serving several function

The list of local constraints is also probably open-ended. But the fact that this list is divided in different categories makes it a little easier to enumerate important classes of local constraints. As mentioned in Section 3.3, the multi-dimensional aspect of the local constraint approach makes it rely on smaller, simpler theories. For example, the argumentative local constraints can be grouped in six classes (cf. Section 6.6), covering all possible argumentative relations between a source and a target. Each one of these classes is likely to contain many specializations of local constraints, but this classification is already a positive result.

---

[11] Disclaimer: this example has an extremely complex structure, and I do not expect my system to handle it...

The main problem of the local constraint approach is to design a practical interaction management technique. The examples above have shown that sometimes the interaction between two local constraints can be ‘‘orthogonal’’ - *e.g.*, the speech-act local constraint and the topic local constraint impose constraints on different independent features. In this case, the formalism used can handle the interaction transparently. The co-presence of the independent features will determine an appropriate surface realization.

There is unfortunately a large number of cases where interaction cannot be transparent. In Section 6, I list the interactions that can occur between the argumentative local constraints and the other classes of local constraints. Most of these interactions cannot be predicted - they must be described explicitly. For example, one empirical result has established a close connection between argumentative agreement and structural completeness (an exchange can be closed when the last two turns are argumentatively co-oriented [Moeschler 85]). Such an interaction must be expressed explicitly. What this means is that in the representation of certain argumentative local constraints, some constraints are added to features describing the structure of the conversation. It also means that the structural and argumentative dimensions are not ‘‘independent.’’ But note that the formalism used does not require expressing the interaction between the local constraints directly - rather it is handled indirectly when two or more constraints impose a constraint on a single feature. This is the most positive point of the unification-based formalism used in this work.

Note finally that local constraints are not used to replace all global guidance from a text generation system. For example, after a discourse segment is complete, it is possible to open a new exchange in a conversation. The topic of the new exchange does not have to be connected to the previous segment. Only a global model can determine what it can be. The idea is rather that local constraints must interact with a global planner that controls the overall development of the conversation. The nature of this interaction remains an open problem that I will not address in this dissertation.

5.2 Classification of Local Constraints

The following classification of local constraints is used in this work:

- *Structural Constraints*: determine whether, in what conditions and how, participants can either extend or close the current exchange.
- *Topic Constraints*: determine how participants can introduce, emphasize or retract discourse entities from the set of discourse entities accessible in the conversation, and relate the entities participating in the target to the entities in the source or to other features of the source.
- *Illocutionary Constraints*: determine what speech-act can be selected by a participant in relation to the speech-act of the source.
- *Interactional Constraints*: determine the level of politeness to use in the interaction, depending on the social relationship existing between speaker and hearer. These constraints determine features like indirection of speech act and or making information implicit.
- *Argumentative Constraints*: determine the type of argumentative relation between the source and the target - whether an argument-conclusion or argument-anti-argument relation etc.

These five classes define the dimensions along which the source-target relation is described. In the rest of this section, I present each class of constraints in more detail.

5.3 Structural Constraints

Structural constraints determine whether and how participants can either extend or close the current segment of conversation. In this subsection, I first define the term ‘‘segment’’ of conversation. Then, using this notion, I define the role of structural local constraints. I next define what it means for a segment to be complete or not: when a segment is complete, it can be closed; when it is not, it must be either expanded or explicitly closed in a special way. Finally, I give examples of how structural constraints interact with other local constraints and show the definition of two structural local constraints.

5.3.1 The notion of discourse segment and the tree model

The assumption behind the definition of the notion of discourse segment is that discourse is built from smaller units. A discourse can be broken into separate segments - which each play a role with respect to the discourse as a whole. This assumption is an extension of sentence-level linguistics, where a sentence can be broken into words or phrases each fulfilling a function with respect to the sentence. In addition, it is assumed that discourse segments can in turn be broken into smaller units - which can be either discourse segments themselves or single utterances. The overall picture is therefore that of a tree-like structure, where larger segments embed smaller ones, and single utterances are leaves. This model is used for example in [Grosz & Sidner 86, Webber 88, Hirschberg & Litman 87, Cohen 87, Reichman 85].

Figure 12 shows a conversation along with a tree describing a possible structure.

```
Q1: Should I take Data Structures?                            | Q1
                                                        |  E1 |    | S21
A2: You could take it,                                  |     | A2 |
    but you have not yet taken calc_1.                  |          | S22
                                                      D |
Q3: Why calc_1?                                         |     | Q3
                                                        |     |
A4: Calc_1 is a freshman course.                        |  E2 |    | S41
    Therefore, you should take it before Data                 | A4 |
    Structures which is a sophomore course.                        | S42
```

Figure 12: A conversation and its structure

The questions that have been addressed when making such an assumption about the nature of discourse structure are:

- What individuates a discourse segment? What makes it complete? Can its boundaries be identified?
- What is the nature of the smallest units (a sentence or part of a sentence)? Can the smallest unit be identified using syntactic criteria? What other criteria?
- What is the nature of the intermediate level units (the segments labeled E1 and E2 for example)? Do we need to characterize them? Is discourse structure recursive or does it have bounded complexity [Polanyi & Scha 83]?
- What effects does the segmentation have on the interpretation of the discourse, and how does it constrain its production?
- Is the tree a complex enough model? Can there be discontinuous segments or structurally more complex units?

I will not attempt to answer these questions here. I simply note that this model of discourse structure is supported by at least the following facts:

- So-called ‘‘clue words’’ (*e.g.*, ‘‘first,’’ ‘‘finally,’’ ‘‘incidentally’’) are best understood as marking boundaries in the discourse structure, and not as modifying the meaning of the sentence to which they are attached [Reichman 85, Cohen 84, Grosz & Sidner 86].
- Referring expressions are sensitive to the discourse segmentation [Grosz 86 p.435]: ‘‘there are different constraints on the use of pronouns and reduced definite noun phrases within a segment than across segment boundaries.’’
- Tense and aspect changes can be related to segment transitions.
- In a conversation, turn boundaries can be related to segment boundaries.

I adopt the tree-model in this work as a working hypothesis to describe the global structure of conversation. Note that the local constraints constitute a different network of relations between discourse segments which do not need to be a tree. Because I deal with conversation as opposed to monologue, I must further refine the classification of segment types. I borrow the classification of [Sinclair & Coulthard 75] of conversational units in:

- Transaction
- Exchange
- Move
- Act[12]

These terms are categories of discourse segments and play the same role in the theory as the notions of Noun and Verb play in the sentence study. A *conversational segment* is any text segment of one of these three ranks. An *act* corresponds to a single speech-act. The syntactic unit corresponding to a single act is difficult to determine. The syntactic unit of semi-act has been proposed in [Auchlin 86], based on the notion of E-node (or ‘‘expression’’) of [Banfield 82]. These definitions try to capture the notion that a speech-act should be associated with each predicative element. In addition, it may be possible that more than one speech-act could be realized by a unit even as small as a semi-act (cf. [Appelt 85] for example). A *move* is a sequence of acts uttered by a single participant. An *exchange* is a sequence of moves uttered by several participants. A transaction is a sequence of exchanges, starting with an opening sequence (the participants meet or start a physically separate interaction) and ending with a closing sequence (the participants terminate the conversation). In terms of local constraints, I will focus on the ranks of exchange, move and act. I will look at ways to open, extend and close moves and exchanges. Since the system will generate short turns and not extended text, it is unlikely that sequences of exchanges will occur.

In addition to these categories, I also use structural relations between discourse segments which are specific to conversation. The standard tree model defines the relation of embedding (or dominance: S1 dominates S2) and linear precedence (S1 occurs before S2). I borrow from [Roulet *et al* 85] the following relations: within the exchange, units can be connected by a relation of *initiation* and/or *reaction*; within the move, sub-units can have a *directive* or *subordinate* status. The directive act is the ‘‘point’’ of the whole move and cannot be removed without changing the function of the whole move. Subordinate moves in contrast are optional and do not affect the function of the move within a larger segment.

For example, Figure 13 shows the tree of the conversation of Figure 12 annotated with the category labels and the additional structural relations.

The notion of initiation/reaction is intuitively easy to understand. It refers basically to the structural aspect of speech acts. I consider in this work that whether a move is initiative and/or reactive in a conversation is directly observable.

---

[12] Note that Sinclair and Coulthard have analyzed conversation in a particular setting (classroom), and have therefore a higher level constituent called ‘‘Lesson.’’ In general, higher level constituents are very much domain dependent.

The notion of directive/subordinate is close to the syntactic notion of subordination - and actually it seems that syntactic subordination implies this notion of structural subordination (cf [Roulet *et al* 85, 2.3.1] for a discussion). But the two notions are not identical. Consider the case of ‘‘but’’: it is syntactically a coordinator (cf [Quirk *et al* 72, 9.37] for a discussion), but I hold that in the segment ‘‘p but q,’’ p has subordinate status and q has directive status. This can be shown using the following test: in the sequence ‘‘p but q therefore c,’’ c must be a conclusion supported by q, not by p. For example, ‘‘it is nice, but it is expensive; therefore I will not buy it’’ is a coherent sequence. In general, the tests to determine what act is directive in a complex move are those that identify the main syntactic clause in a complex clause (cf. [Quirk *et al* 72, 11.8-11.12] for example) and the ‘‘linking test’’ for argumentative orientation. I will work on a more precise characterization of the notion of ‘‘directiveness.’’ Note that the notion seems to be identical to the notion of ‘‘nucleus’’ in RST [Mann & Thompson 87, Matthiessen & Thompson 87] but I haven’t yet compared the arguments for the RST nucleus idea with the notion of directiveness.

```
                              | Q1  [Act/Init]
            |  E1  [Exchange] |                  | S21  [Act/subordinate]
            |                 | A2  [Move/React] |
            |                                    | S22  [Act/Directive]
[Exchange]  |
            |                 | Q3  [Act/React/Init]
            |                 |
            |  E2  [Exchange] |                  | S1  [Act/Subordinate]
                              | A4  [Move/React] |
                                                 | S2  [Act/Directive]
```

Figure 13: Annotated structural description

Note that the phenomenon of ‘‘rank-shift’’ known in sentence syntax, where a higher level constituent serves as sub-constituent to a lower-level constituent can also happen at this level. For example, at the sentence level, a clause can be a sub-constituent of a noun phrase as a relative clause. Similarly, an exchange can be integrated as a sub-constituent of a complex move, as shown in Figure 14.

```
A1: You have taken Data Structures, right?                         | A1  [Act]
                                                    | E1  [Subord] |
B2: Yes.                                            |              | B2  [Act]
                                             [Move] |
A3: So you should definitely take AI next.          | A3  [Directive]
```

Figure 14: Rank shift in conversation structure

5.3.2 The role of structural local constraints

A structural local constraint operates between two conversational segments of any rank. In this work, generally the target is of rank act, because in my system acts are planned one at a time[13].

Local constraints are used to:

- Choose the source of a new turn.
- Determine when a segment is complete.

---

[13] In case of reinterpretation, it may happen that the target is not an act but a move and the local constraint defines a new interpretation of the source.

- Decide how subordinate, directive, reactive and initiative statuses can be combined in a new segment.
- Determine how to extend a move or an exchange by adding a new act.
- Determine the features of a new segment made up of a pair source-target from the features of each of its constituents.

I will define the set of accessible ‘‘sources’’ - that is, the set of conversational segments which are potential sources. While my system will use heuristics to select a source to respond to, it is not at all clear how to choose the best one from this set. In fact, it is not clear that there even exists a best source. People can respond to a wide variety of sources in a conversation. Given the difficulty of defining ‘‘best,’’ this is not a problem I will address. I will define local constraints to combine the features of directive and subordinate with those of initiative and reactive. And finally, I will define a set of local constraints to extend a move and an exchange and build a new segment when necessary.

When producing a new turn, it is important to decide whether it is a new segment, or it expands an existing one. Special linguistic devices must be used to expand a segment that has been previously closed, as shown in example 3 of the introduction. Similarly, the transition from one segment to another must often be marked by devices like ‘‘incidentally’’ or ‘‘by the way’’ when the last segment of the discourse is not explicitly closed. This indicates that in order to generate a move in a conversation, it is important to specify if the move is opening, expanding or closing a segment. It is also important to determine if a segment is ‘‘complete’’ - that is, if it can be closed - or not. The following structural local constraints are therefore the basic structural local constraints:

- Open exchange.
- Open move.
- Expand exchange.
- Expand move.
- Close exchange.
- Close move.

Depending on the nature of the source and of the target, these local constraints can determine the use of linguistic devices and interact with the global model of the conversation.

By interacting with local constraints on other dimensions, it is possible to choose which constraint from this list can be used at a given time. After defining what is a complete segment, I will show examples of this selection process.

5.3.3 What is a complete segment?

At the sentence level, the notion of completeness is clearly defined: an incomplete constituent is non grammatical, *e.g.*, ‘‘looks at’’ has no subject and no complement. At the discourse level, a similar notion of completeness can be defined. In conversation, it is useful to distinguish between move and exchange completeness. Intuitively, a move is complete if it can be interpreted in relation to its source so that it can be classified as initiative and/or reactive (note that this implies that the completeness of a segment depends on its position in a sequence). An exchange needs to be closed to become complete. There are different ways to close an exchange [Schegloff & Sacks 73, Moeschler 85, 3.2.2], *e.g.*, with linguistic devices such as ‘‘ok’’ or ‘‘if you want...’’

The notions of programming and integration introduced in [Roulet *et al* 85] are useful to determine what a complete segment is. A segment ‘‘programs’’ a larger segment if it is necessarily, as marked by its linguistic form, part of a larger segment. For example, an initiative segment always programs an exchange embedding it. Thus a question creates a slot for an exchange embedding the question. Similarly, certain tags, like ‘‘right’’ in the previous example, indicate that the question and therefore the exchange embedding it are part of a larger move (the question is presented as a preparation for some upcoming act). Similarly, a segment can integrate previous lower-level segments into a higher level unit.

Figure 15 illustrates the notion. The two exchanges E1 and E2 are not connected, until A5 integrates them into a complex move, where they serve as subordinate units. Note that the integrative behavior of A5 is marked by the use of ‘‘so,’’ and the subordinate status of E2 is marked by the use of ‘‘and.’’

```
A1: You have taken Data Structures?                         | A1
B2: Yes.                                               | E1 | B2
                                                       |
A3: And you can program in Lisp?                       |    | A3
B4: Approximately.                               [Move]| E2 | B4
                                                       |
A5: So you should take AI this semester.               | A5
```

Figure 15: Integration of discourse segment

If a segment programs a larger segment, it is not complete. It requires an integrator segment to follow.

In conclusion, it is important for a generator to signal the incompleteness of a segment. This signalling allows the hearer to generate appropriate expectations regarding the structure of the conversation. Structural local constraints must therefore determine and indicate if segments are complete or not. Similarly, the fact that a source is complete or not constrains the form of the target. For example, the expansion of an exchange that has been closed must be marked in the surface, as shown in example 3 in the introduction.

5.3.4 Interaction of structural constraints with other dimensions

In order to decide what structural constraint can be used, it is necessary to consider the current global structure of the discourse[14] and other classes of local constraints along different dimensions. I now show an example of such interactions.[15]

The `expand exchange` constraint must be used when the last two moves in the exchange are argumentatively opposed. This means that it is possible to expand an exchange by contradicting the move produced last. Conversely, another way to look at this constraint is that if the last two moves are contradict each other, then the exchange must be marked as incomplete. Note that this means that argumentative coorientation is sufficient but that it is not necessary for the completeness of an exchange. Examples of this effect are shown in Section 6.3. Figure 16

---

[14] Structural local constraints interact more closely with the global model of the conversation than do other local constraints. When opening a new segment, the global model must specify what type of higher level segment must be programmed. In this case, local constraints are not directly applicable, as there is no source available in the previous discourse. Local constraints are still useful, however, when programming occurs: the global model specifies what type of higher level segment must be programmed - for example an exchange. Part of this specification is to ‘‘pre-select’’ some features of the first unit (the initiation of the exchange). But it can also include a partial specification of the reaction. For example, the expected orientation of an answer can be pre-selected before the question is asked - leading for example to a question of the form ‘‘haven’t you already taken DS?’’ In this case, there exists a local constraint between the question and the answer, even though the question has not been produced yet. The form of the question is therefore constrained by the expected form of the answer (cf example 2 on broccoli in the introduction for an example of this phenomenon).

[15] Structural local constraints operate like conventional patterns of conversation. The list of such patterns is probably open-ended or at least very large. Empirical work in conversation analysis has produced many examples of domain dependent patterns (for example, repair sequences [Schegloff, Jefferson & Sacks 77], telephone openings [Schegloff 79], compliment exchange, service encounters [Ventola 87] or commercial transactions). Whether some underlying principle could unify such findings remains to be proven. Even if it could be found, the mechanism of local constraints remains a useful formalism to compile such findings in a regular way.

shows how this constraint is represented. Complex exchanges are represented by bracketing to the right, that is by a structure of the form [[[[1 2] 2] 2] 2]... The local constraint specifies that an exchange of the form [1 2] or a single move 1 can be expanded into a complex exchange [[1 2] 2], which is not complete, if it can be established (by an argumentative local constraint), that the last two moves of the exchange oppose each other.

```
((cat local-constraint)
 (source ((structure
           ((cat exchange)
            (move1 #source
                    ((structure (alt (((cat move)
                                       (complete yes))
                                      ((cat exchange)
                                       (complete no)))))))
            (move2 #target)
            (complete no)))))
 (target #target ((structure
                    ((cat move)))))
 (relater ((cat arg-lc)
           (source #source)
           (target #target)
           (argumentation ((oppose s-t))))
```

Figure 16: Representation of the Expand Exchange local constraint

5.4 Topic Constraints

Topic constraints determine how one can introduce discourse entities into the conversation. They manage the flow of accessible entities, and control when new entities can be introduced. They also constrain the way the surface generator can build referring expressions. The view taken here is that discourse entities are associated with each discourse segment in the global structure, and that their accessibility is conditioned by the accessibility of the discourse segment in the structure.

Topic local constraints are used to:

- Determine what discourse entities are accessible.
- Determine what world entities can be introduced in the conversation (from the set of mentionable contents given in input).
- Determine the focus status of entities in the target.

Topic constraints prescribe that the set of discourse entities in the source and in the target must be related in some way. In general, the constraint is that there must be a semantic connection between the entities. For example, in example 1 given in the introduction, the sequence (1) is not coherent because the entity ‘‘snow’’ is not used in the first sentence, and is not related to the entities of ‘‘Harry’’ and ‘‘fall’’ through some activated semantic relation. In (3), in contrast, the entity ‘‘snow’’ is explicitly related to the entity ‘‘fall’’ and the sequence of the two sentences satisfies the topic local constraint.

Semantic links can be used to establish a connection between entities in the source and entities in the target. One type of semantic link includes links that exist in the knowledge base (for example, `location(e)` where `e` is a concept of type event). Another type of semantic link that can be used is a dynamic inference created by a reasoning program from existing knowledge base objects.

There are, however, other ways to construct a topic link between the source and the target: this is done by using a *thematization procedure*. Thematization procedures re-interpret the source of the move and dynamically create a new discourse entity which is pragmatically related to the source. For example, in the sequence ‘‘A: It’s raining. B: I don’t believe you,’’ the second turn links on the speech act performed by A, not on the semantic content A conveys. I model this situation by using an illocutionary thematization function, which when applied to an utterance makes a discourse entity from the performance of the speech-act - which therefore can be explicitly mentioned in the following conversation.

In general, a thematization procedure is a procedure that creates discourse entities from some pragmatic feature of the utterance event. For example, the accent with which a speaker pronounced a word can become a discourse entity (*e.g.*, ‘‘A: can you pass me ze bread, pliz? B: Are you from France?’’); or the choice of a particular word (*e.g.*, ‘‘A: He is a dumb interlocutor. B: Why do you always use these long words?’’); even the fact that an utterance has been produced at all (*e.g.*, ‘‘A: Hi! B: How dare you address our Presidio Supremo!’’). Formally, a thematization procedure takes as input a representation of the interpretation of a discourse segment (the IFs described in section 7.4) and returns a list of data-structures each representing a new discourse entity and how it is connected to the segment.

The thematization procedures I will use in this work are:

- Role of clause: all entities that are the participant of a clause can become discourse entities. They are marked by the name of the role they fill (*e.g.*, `agent` or `benef`).
- Predicate of clause.
- Clause as a whole.
- Speech-act: the speech-act of the segment.
- Static-inference: an entity related to either one of the roles of the clause or to the predicate by an existing active semantic link in the knowledge base[16]. The entities are marked by the original entity, its role, and the nature of the link.
- Argumentative link: one of the participants of a conclusion that can be related to the original proposition[17].

I will not directly address the problem of controlling when thematization procedures should be activated. The most likely mechanism would be based on the technique of demons which would be activated by certain patterns in the input (*e.g.*, a ‘‘strange’’ accent triggers the ‘‘accent-thematize’’ procedure). In the implementation, I will use the simple strategy of systematically activating all the procedures listed above (the static inference is used with a subsumption link only). Note that I do not choose between the several candidates: they are all added to the list of potential discourse entities.

Thematization procedures produce a list of potential discourse entities. A focusing algorithm is needed to maintain a structured representation of this list, and to distinguish among different levels of salience and familiarity.

To work on this aspect of the focus algorithm, I will use a theory of focus derived from [Grosz 77], [Sidner 79] and [McCoy & Cheng 88]. I will also use the classification of given/new information presented in [Prince 81] and used in [Dale 88]. These theories are not immediately usable in the framework of local constraints, and will require some work to be adapted. The goal is to determine which entities in the target are focused, and what level of familiarity they have. Note that Kamp’s Discourse Representation Theory (DRT) [Kamp 84] addresses these problems also,

---

[16] How active links can be triggered is discussed below

[17] The notion of argumentative relation is described in section 5.

and would be a good candidate. This domain is not the focus of my work, however, and I intend to only use existing theories. But I need to describe this theory enough to integrate it properly with the other local constraints.

The particular topic local constraints on which I intend to work are:

- Maintain focus: *e.g.*, ‘‘You should take Data Structures. It is a prerequisite for AI.’’
- Link on derived focus: *e.g.*, ‘‘A: You have not yet taken calc_1. B: Why calc_1?’’
- Link on previous focus: *e.g.*, ‘‘A: should I take Data Structures? B: You can take it, but you have not yet taken calc_1. Data Structures is a sophomore course.’’

For each of these, I will detail how the choice of local constraint influences the data-structures representing the focus state and the surface realization of the clause (possible pronominalization, focus assignment in the clause, clause ordering and choice of connective are all affected).

5.5 Illocutionary Constraints

Illocutionary constraints determine the type of speech-acts that can be used in the target, and how this affects the type of speech-act realized by the source-target pair as a whole. Remember that considering the notion of uptake properly means that the speech-act of the source is defined to some extent by the speech-act realized in the target. To avoid a ‘‘circular’’ approach where the speech-act of the source is defined only by the reaction it induces in the target, I use the notion of ‘‘surface speech act’’ which is a characteristic of the source’s linguistic form: the surface speech act is either *request* or *commit*[18] and is related to the surface speech act performed at the surface. The exact speech-act realized by the source is left up to negotiation until a target gives it a ‘‘recognized’’ status.

Illocutionary local constraints determine what pairs of ‘‘complete’’ speech-acts are compatible.

The following table of pairs (from [Moeschler 86, p.95]) is an example of how the pairs can be arranged:

Initiation	Positive Reaction	Negative Reaction
offer	acceptation	refusal
request	acceptation	refusal
information request	answer	negative answer
confirmation request	confirmation	infirmation
assertion	positive evaluation	refutation

Note that for each type of act in the source (left column), two types of reactions can be expected: positive and reactive. The positive reaction is always ‘‘preferred’’ [Levinson 83, p.332]. The negative reaction is always ‘‘cohesive’’ with the initiation, but generally requires a further reaction by the participant who initiated the exchange. Note that these relations between speech-acts hold among constituents of an exchange. There is not much that can be constrained in terms of speech-act within a single move, and this is a known limitation of speech-act based analysis of coherence (cf. [Taylor and Cameron 87, p.58 and p.72] for example). Illocutionary constraints thus mainly operate within the exchange, and not much within the move.

The notion of illocutionary force in speech-act theory will be related in this work to a notion of politeness: choice

---

[18] An assertion is a commitment to support the truth of the proposition.

between alternative forces of the same illocutionary function are to be decided by the interactional constraints.

Note finally that I will assume that the speech-act function of a complex unit (such as a multi-act move) is the speech-act of its directive act. This makes it possible to apply this pair model to constituents of higher level.

5.6 Interactional Constraints

Interactional constraints determine the level of ‘‘politeness’’ that should be used in the target. I will not implement any of these constraints, but I will use the theory of politeness presented in [Brown & Levinson 87]. This theory is based on three factors, called P, D and R and maps this three parameters to the choice of an *interaction strategy*. The type of interaction strategies listed in [Brown & Levinson 87] include `be indirect`, `hedge` or `seek agreement`. In this work, I assume that a mechanism such as the one used in [Brown & Levinson 87] is available to map from a given triple (P, D, R) to one of the strategies listed in pages 102 and 131. Therefore, these strategies will be directly given in the input to my system, and will control and interact with the other local constraints.

To be precise, the parameters P, D and R stand for [Brown & Levinson 87, p.74]:

- P: Power. The relative ‘power’ of Speaker and Hearer (an asymmetric relation).
- D: Distance. The ‘social distance’ of S and H (a symmetric relation).
- R: the absolute ranking of imposition in the particular culture.

The role of the interaction strategies defined as the output of the model is to manage a trade-off between the conflicting requirements of two fundamental pulsions in social interaction [Brown & Levinson 87, p.62]:

- *Negative Face:* the want of every ‘competent adult member’ that his actions be unimpeded by others.
- *Positive Face:* the want of every member that his wants be desirable to at least some others.

The mapping described in the model is based on a theory of rational interaction, where social agents seek to maximize some utility from their interaction, in terms of face benefits. It uses a formalism which is very close in spirit to a planning formalism familiar to AI systems.

In summary, I assume here that some ‘‘politeness’’ module is available, that would estimate the P, D, R of a given interaction, map from this estimation to a choice of interaction strategies. These interaction strategies are part of the input to my system, and interact with other local constraints. For example, the `seek agreement` strategy, is used by the argumentative local constraints to select content. Note again that this is not a domain to which I intend to contribute, rather, I use an existing theory and study how it interacts with the other dimensions I examine.

6 Argumentation and Generation

In this section, I present in more details the argumentative local constraints and their interaction with the other types of constraints.

Argumentation is one tool to express the speaker’s attitude to the information he conveys. The speaker can express his support for an opinion or his rejection or disapproval of an opinion. The information or opinions we express do not always originate in ourselves. We often echo or report propositions from diverse sources. Argumentation allows a speaker to express his attitude to these external sources.

6.1 Argumentation, Content and Surface Realization

The theory of argumentation used in this work is mainly based on [Anscombre & Ducrot 83] and [Moeschler 85]. The notion of argumentation covers two distinct problems: how to choose content to ‘‘persuade’’ a hearer of some conclusion; that is, how to modify the beliefs of the hearer (this is conceptual argumentation) and how to present this content as an argument for or against a conclusion (this is linguistic argumentation). The idea defended in [Anscombre & Ducrot 83] is that linguistic argumentation is to a large extent independent of semantic content: language offers a set of tools to present any fact as an argument for any conclusion. This implies that the argumentative intent of an utterance must be part of the input to a surface generator.

The linguistic theory of argumentation assumes that an utterance cannot be produced alone, but that it must be prepared in relation to some background conclusions, with which it maintains an argumentative relation. In the theory, these conclusions form the ‘‘argumentative orientation’’ of an utterance (the basic objects of the theory - arguments, scales, conclusions and topoi - are described in more detail in the next subsection). For example, in (10), repeated here ‘‘Yes the number are here, but there are pictures of the crane on mine,’’ the move is related to the implicit conclusion that A and B do not share the same instruction sets. The particular conclusion is not marked in the linguistic surface of the move, but the relation between this move and the question preceding it (‘‘Do you have the same numbers?’’) is marked by the use of the connective ‘‘but.’’ We must conclude, because ‘‘but’’ is used, that B takes the orientation of the question to be some conclusion C (which we will assume is ‘‘A and B have the same instruction sets’’), the orientation of the first part (‘‘Yes’’) to be the same conclusion C, the orientation of the second part of the move to be the opposite of C, and the orientation of the whole move (p but q) to be not-C. The choice of ‘‘but’’ is justified by these relations between the conclusions, and not by the semantic relations between the content of each act. It is the role of a linguistic theory of argumentation to describe these types of relations, and how they are expressed by devices like ‘‘but’’ or ‘‘only.’’ A conceptual theory of argumentation in contrast, would determine if the fact that there are pictures of the crane on B’s instruction set is a good argument against C, and why.

The notion of argumentative orientation (*i.e.,* what is the set of conclusions that an utterance may support) is very close in spirit to the notion of ‘‘discourse purpose’’ used in [Grosz & Sidner 86]. It is, however, different for three reasons: the notion of argumentation is *gradual*, it includes the notion of *contradiction*, and is related to a ‘*linguistic*’ theory.

Gradual means that the relation ‘‘A supports C’’ can be assigned a strength: A1 can be presented in an utterance as a better argument for C than A2. This configuration would explain the use of a word like ‘‘even’’: ‘‘John drinks even in the morning’’ is a better argument for some conclusion C than ‘‘John drinks at night’’ and this ranking is expressed by the use of ‘‘even.’’ Many aspects of language are gradual, and this aspect is not easily captured by a planning theory.

Supported opinions are best put forward when competing opinions are discredited. This explains why argumentation grants an important place to the notion of *contradiction*. A conclusion is supported by some background justification, but it is made convincing by discarding competing conclusions. The notion of contradiction is not easily captured by a planning theory.

That there is a linguistic theory behind the notion of argumentation is an advantage because it establishes a direct link between argumentative relations and choice of linguistic devices. The relations between discourse purposes used in [Grosz & Sidner 86], `dominance` and `satisfaction-precedence`, have not yet been related to surface decisions in an easy way. The notion of discourse purpose, on the other hand, is easier to use in the context of existing AI planning systems, that make use of the same notion of goal.

Finally, I want to stress the importance of argumentation by relating it to the notion of control of inference [Joshi &

Weinstein 81]. One problem in controlling inference is deciding what inferences should be pursued and which ones should be ignored. The notion of argumentation helps in doing just that, because it indicates, in the linguistic surface, what types of inference rules can be used, and what types of conclusions should be found.

In the rest of this section, I first define the basic notions of an argumentative theory (argument, scale, conclusion and topoi) and describe how they are represented computationally. Then, I describe the interactions existing between argumentative constraints and other constraints. Finally, I list the local constraints used in this work, and introduce the argumentative strategies which I use to control the use of these constraints.

6.2 Defining and Representing Arguments, Scales, Conclusions and Topoi

6.2.1 Arguments

Arguments are classes of utterances that are ranked with respect to a given conclusion. For example, the following set of arguments support the conclusion ‘‘John drinks a lot’’: {John drinks 1 glass, John drinks 2 glasses, ...}. A class of arguments is represented as a propositional content, called the scope of the argument; a focus, which is one of the roles of the scope; a scale, which is an ordered set of values that the focus can hold; and an orientation, which indicates whether moving up the scale makes the argument stronger or weaker. For example, Figure 17 shows a representation of an argument for the conclusion ‘‘John drinks a lot’’ as it would be used in the utterance ‘‘John drank even seven glasses.’’ Figure 18 illustrates the use of scope in arguments. The notation used is that of functional descriptions, and will be explained in section 7. Note that this representation denotes a set of arguments, all supporting a conclusion describing John drinking some quantity of alcoholic beverage.

The relation ‘‘p is an argument for q’’ can be read as: ‘‘the more I have reasons to believe that p, the more I have reasons to also believe q.’’ In other words, it means that p is a partial evidence for q.

```
((cat arg)
 (scope ((process-type action)
         (agent John)
         (process drink)
         (medium ((concept alcoholic-beverage)
                  (quantity Q)))))
 (focus Q)
 (scale "quantity")
 (orientation +))))
```

Figure 17: Argument for ‘‘John drinks a lot’’

```
((cat arg)
 (scope ((process-type action)
         (agent A)
         (process drink)
         (medium ((concept alcoholic-beverage)
                  (quantity Q)))))
 (focus A)
 (scale "number")
 (orientation +))
```

Figure 18: Argument for ‘‘Many people drink’’ in ‘‘Even John drinks’’

6.2.2 Scales

I distinguish in this work between two types of scales: linguistic and conceptual. Conceptual scales are related to some order in the domain and represented in the knowledge base of the system. For example, if the scale ‘‘temperature’’ is relevant in the domain, it will correspond to a domain-specific predicate which can compare two concepts on this scale, and identify which one is warmer. There are also linguistic scales which do not correspond directly to conceptual scales. Linguistic scales correspond to a conventional structuring of the lexicon. Linguistic scales are defined using the test proposed in [Horn 72]: two lexical items x and y are on the same scale if it is possible to use the sentence ‘‘x and even y.’’ For example, ‘‘hot’’ and ‘‘warm’’ are on the same linguistic scale, because one can say ‘‘it is warm, and even hot.’’ But, while ‘‘cold’’ and ‘‘hot’’ are related to the same conceptual scale, they are not members of the same linguistic scale, because one cannot say ‘‘it is cold and even hot.’’ (Other similar tests exist to check membership on a linguistic scale, cf [Hirshberg 85] for a discussion of these tests.)

Conceptual scales have been studied in depth by Hirshberg [Hirshberg 85] who proposed to use partially ordered sets as conceptual scales (she does not distinguish between conceptual and linguistic scales, but her description corresponds to the use of conceptual scales). She also provides a list of conceptual scales of general use.

6.2.3 Linking Content to Arguments

The connection between linguistic and conceptual scales is a difficult problem; it is a well known part of lexical choice (how do you decide to use ‘‘big’’ in ‘‘he is big for a Japanese’’ or ‘‘cold’’ in ‘‘cold fusion’’?). The problem is to map from a value on a conceptual scale to a lexeme on some related linguistic scale. There is actually a larger problem underlying this issue: how does a system relate a fact to an argument? The problem is that facts should be considered as objective representations of reality, and arguments include a subjective bias towards a conclusion. The speaker is free to interpret facts and force them as arguments for a conclusion or against it. This is particularly clear in political debates (for example, on the issue of abortion, the fact that ‘‘pro-lifers’’ are often in favor of capital punishment and pro-choice are not, is an argument used by each cause to show the inconsistency of the other side). In general, placing a fact on a scale is a complex act of interpretation that requires a great deal of world knowledge. An interesting treatment of the problem is proposed in [Dispaux 84].

In this work, I assume that facts are already linked to conceptual scales. This is done by requesting the knowledge base of the system to indicate for each concept it represents what scales it can be a member of. I will then only look at the problem of mapping from a conceptual scale to a linguistic scale - that is, for example, mapping from a temperature of an object to a word like ‘‘hot’’ or ‘‘cold.’’

6.2.4 Linking Arguments to Conclusions: Topoi

The ranking of arguments on a scale allows the comparison of two arguments on a single scale. There is, however, a need to compare arguments across scales. In the sentence ‘‘this is nice, but it is expensive,’’ one needs to express the fact that ‘‘expensive’’ is a better argument for not buying an object than ‘‘nice’’ is for buying the object. To allow this kind of comparison, I use the notion of topoi. Topoi relate two scales and express a correlation between the variations on each scale (cf. [Raccah 87, Anscombre & Ducrot 83, List 84] for a discussion of the notion).

Intuitively, topoi are inference rules. They have a left-hand side which is a premise, and a right-hand side which is a conclusion. They differ, however, from standard rules, in that both the premise and the conclusion are gradual predications - they position an argument on a scale.

Figure 19 represents the topos ‘‘the harder a course Y, the less a student X wants to take Y.’’ Topoi have a left and a right hand-side. Each side is represented as arguments are (cf above). Topoi also include a lower and an upper-bound on each side, which limit the applicability of the inference rule. Such a topos declares that there is a correlation between the variation in ‘‘nice’’-ness of an object and the variation in ‘‘desire’’ a person may feel for it.

Topoi are used to link arguments to conclusions: the topos in Figure 19 allows one to link an argument on the scale ‘‘nice’’ to a conclusion on the scale ‘‘desire.’’

```
((left ((scope ((process-type attributive)
                (carrier   Y:((cat course)))
                (attribute Z)))
        (focus Z)
        (scale "hard")
        (orientation +)))
 (right ((scope ((process-type mental)
                 (process P:want)
                 (processor X:((cat student)))
                 (phenomenon ((process-type action)
                              (process take)
                              (agent X)
                              (medium Y)))))
         (focus P)
         (scale "desire")
         (orientation -))))
```

Figure 19: An example of topos

Obviously, topoi can be linked together. A topos `[+X, +Y]` and a topos `[+Y, +Z]` can be linked to form a topos `[+X, +Z]`. Such combination rules are not always easy to control: the fact that topoi are gradual makes them different from regular inference rules. It is not clear whether topoi should be symmetric (is `[+X, +Y]` equivalent to `[+Y, +X]`?), the rules on sign inversion are also not clear (is `[+X, +Y]` equivalent to `[-X, -Y]`?). In this work, I will not deal with the problems of controlling inferences based on topoi. If a mechanism (based on focus for example) is developed to filter the set of possible inferences, and some limit can be put on the length of acceptable inference chains, then it would be possible to control the inferencing. I will only deal here with chains of length one, and assume topoi are symmetric and that symmetric permutations of signs are also valid.

6.3 Interaction between Argumentation and Structure

Argumentation determines one aspect of the completeness of discourse segments:

- *Within the exchange*: the last two turns of an exchange must be co-oriented for the exchange to reach completion. This constraint is related to the result of conversation analysis stating that agreement is a preferred second in the preference organization [Pomerantz 75, Pomerantz 84]. This means that choosing an argumentatively anti-oriented turn in an exchange implies the choice of continuing the exchange (not closing it). Thus, this interaction can also go the other way: one way to keep an exchange open, is to produce an anti-oriented move. Note also that exchanges can be non-satisfactory, and closed explicitly, without reaching the co-orientation conclusion, and this closing method is one of the possible structural local constraints (cf. [Schegloff & Sacks 73] for a definition of closing an exchange and [Moeschler 85, Chapter 5] for a discussion of closing strategies). The following conversation illustrates how contradiction and exchange expansion are related:

```
A1: can I take computability instead of fundamental algorithms?
B2: yes, you can take it. But fundamental algorithms would be more
    helpful because it is a prerequisite for all software courses.
A3: But I plan on concentrating on theory.
B4: You still need to take the required software courses.
A5: But I would much rather take computability.
B6: It would be a mistake.
A7: Ok, ok.
```

Note that each turn in this exchange is argumentatively opposed to the previous one. The exchange does not reach its conclusion until a sort of agreement is reached (even if this agreement is not satisfactory to A). The following example, taken from a transcript of advising sessions, illustrates how a sequence can be explicitly closed when such an agreement cannot be reached:

```
A1: I could take both of them <1103 and 3310> and depending on
    which I need I can either drop one or it would be possible
    for me to take both.
B2: But they’re offered at the same time.
A3: Yeah, well, I’m saying I don’t think that that would be too
    great a limitation because I may not need to take both. What
    I could do is study in text book for 1103 and go to every third
    class and show up for the tests and stuff, and I have done that
    sort of thing with other courses...
B4: See, I don’t think between the two of them... OK, I’ll tell you
    what you have basically between them.  I’m not sure exactly how
    this is going to run, 1103.  His office is right next door so...
A5: Maybe I can talk to him.
B6: Yeah, if you want you can go and talk to him.
A7: Ok, Well. I have another question (...)
```

- *Within the move*: Co-orientation can be used to expand the move, by adding arguments for the directive act, either before, as a preparation, or after, as a support. As for anti-orientation, a move must have a coherent argumentative orientation to become complete. Therefore, if two anti-oriented acts are used, the contradiction must be resolved. There are different structural ways to resolve the contradiction: (1) by marking one of the acts as directive, the other as subordinate; (2) by further expanding the move with more arguments, of different polarities, and concluding with a resolver act (generally introduced with clue words of the form ‘‘all in all, finally’’); (3) cancelling the contradiction and starting over a new turn (by explicitly opening a new move - thus abandoning the contradictory turn; this is often marked by clue words of the form ‘‘well, ok’’). The following examples, also taken from transcripts of advising sessions, illustrate move expansion with resolution and with cancellation:

```
I do know the programming load of PLT is very heavy.
AI is probably reasonable
although you do programming in that,
and then the Calculus would be no programming.
But that would mean next semester.
Then you would do Computability which is no programming.
And the PLT2 compilers has a very big project toward the end
of the course.
and Natural Language involves... does involve programming,
but I don’t think it’s outrageous.
So now, definitely AI, definitely calculus.

So you could, you could do that in the fall.
Your last semester here is in the fall.
You could do it then,
and you do have to take this Computability and your Calculus,
but you could do the computability next semester.
That would be heavy again.
You could do that...
See, another thing you could do is (...)
```

These two interactions establish the structural conditions of completion for moves and exchanges. They show how the interaction between structural and argumentative local constraints can help control what option is chosen within each class of local constraints.

6.4 Interaction between Argumentation and Illocution

In section 5.5, the adjacency pairs used in this work were listed. I distinguished there between positive and negative reactions. Argumentation allows a better definition of this distinction: a *positive* second part to an adjacency pair is one that is co-oriented with the first part. This definition depends of course on what argumentative orientation is assigned to each type of speech act. Here again a notion similar to uptake is important: the orientation of a segment is part of its interpretation by a hearer, and becomes a real orientation only when it is recognized at such, and validated by a target. Similar to the notion of ‘‘surface speech-act,’’ I use a notion of ‘‘presented argumentation’’ which is only the constraint on the argumentative orientation marked into the surface of a segment by linguistic devices such as connectives or gradual modifiers. This presented argumentation is validated and becomes a full argumentative orientation only when the following discourse validates it, and instantiates the missing components. For example, the question ‘‘Do you have the same numbers’’ has no constraint on the orientation (either for or against), and has standard constraints on the content of the conclusion it can support (the conclusion must be related in some way to the concept of having the same number). In (10) (example 5 in 1.3), it is the answer in B3 and B4 which project an orientation on this question.

In general, the interaction between argumentative and illocutionary local constraints determines what argumentative orientation is compatible with a turn of the form S(PC) - where S is a speech-act, and PC a propositional content. So, it is through the interaction between local constraints that Arg[S(PC)] is constrained.

The problems of negation and refutation is a very important issue which would be handled by the same interaction (assuming refutation is a speech-act); similarly, the relation between the force of an argument and the force of the speech act of the segment realizing the argument should be studied. These are two aspects that I will not address in this work.

6.5 Interaction between Argumentation and Topic

In section 5.4, the notion of thematization procedure was introduced. I assume in this work that there exists an argumentative thematization procedure. The role of this procedure is to determine what conclusions are reachable from an argument, and to include the entities of such an accessible conclusion in the list of accessible discourse entities of the source. For example, in the following sequence ‘‘p: this car is beautiful, q: but I have no money,’’ the connection between the money and the car must be indirect: from p, one can reach the conclusion ‘‘c: this car is worth buying’’ (which is also triggerred by the use of ‘‘but’’), then from c, the entity ‘‘money’’ is semantically introduced (as a static derivation from the concept ‘‘buy’’) - and it is therefore accessible in q. The Argumentative Thematization Procedure uses the notion of *topoi* presented above to relate an argument to a conclusion.

Argumentation can therefore constrain the selection of accessible entities, it can also constrain the focus level assignment of entities in the target. Arguments are represented as a simple propositional content, with a focused element being moved along a scale of comparable elements. For example, in ‘‘John drinks even in the morning,’’ the argument is represented as a complex structure indicating that `Drinks(John, At-time X)` and the focus of the argument is the `at-time` complement. In contrast, in ‘‘Even John drinks in the morning,’’ the focus of the argument is the `agent`. This notion of focus is related to the notion of focus used in the topic relation. The interaction between these two notions is also handled through the interaction of argumentative and topic constraints (cf [Hirshberg 85, section 6.3] on this problem).

6.6 Argumentative Local Constraints

I now can list the argumentative local constraints used in this work. All possible combinations of argumentative relations are listed below. They collectively express the constraint that there must exist an argumentative relation between the source and the target:

- Co-orientation: S (the source) and T (the target) are co-oriented.
    - S and T are both arguments for the same Conclusion
    - S is an argument for T
    - S is a conclusion of T (T is an argument for S)
- Anti-orientation: S and T are anti-oriented.

- S is an argument for C1, T for C2, C1 and C2 are opposed
- S is an argument for C, T and C are opposed
- T is an argument for C, S and C are opposed

When trying to select an argumentative constraint, the system needs to select a proposition from the pool of mentionable items, convert it to an argument, and check whether this argument is compatible with the other local constraints. Depending on what propositions are in the pool, only certain argumentative relations can be instantiated.

Each argumentative LC also imposes constraints on the other dimensions. Through the interaction with other dimensions, only certain argumentative relations can be retained. Consider the example I from the introduction:

Q: *Goal: concentrate on AI* Should I take Data-structures?

Let’s assume the pool contains the following propositions after the expert system has run:

```
1. Goal(S, Concentrate(S, AI))
2. Should(S, Take(S, DS))[19]
3. Prerequisite(DS, Intro_AI)
```

And let’s assume the following argumentative relations can be established given the current topoi base:[20]

```
A. 2 is an argument for 1.
B. 3 is an argument for 1.
```

The question is given a representation including the following:[21]

```
((Speech-act Query-information)
 (Propositional-Content 2)
 (Argumentative-orientation ((scope 1) (orientation +)))
 (Topic [DS])
 (Structure ((Initiative yes) (Complete yes))))
```

The following argumentative LCs can each be matched:

I. S is an argument for T:

```
((Source ((Propositional-content 2)
          (Argumentative-orientation ((scope 1) (orientation +)))))
 (Target ((Propositional-content 1))))
```

Answer: You want to concentrate on AI.

II. S and T are arguments for C:

```
((Source ((Propositional-content 2)
          (Argumentative-orientation ((scope 1) (orientation +)))))
 (Target ((Propositional-content 2)
          (Argumentative-orientation ((scope 1) (orientation +))))))
```

Answer: Yes, you should take DS.

III. S and T are arguments for C:

```
((Source ((Propositional-content 2)
          (Argumentative-orientation ((scope 1) (orientation +)))))
 (Target ((Propositional-content 3)
          (Argumentative-orientation ((scope 1) (orientation +))))))
```

Answer: DS is a prerequisite for Intro to AI.

Interaction with other dimensions:

Case I will not be used because it does not satisfy the topic LCs (there is no common discourse entity between S and T).

Case II can be used. It is marked ‘‘direct’’ by the interactional LCs (same propositional content in S and T).

Case III can also be used. Is is marked ‘‘indirect’’ by the interactional LCs (the argumentative relation is indirect), and will be used only when an ‘‘indirect’’ answer is necessary.

Note that the answers II and III can be combined to produce the answer shown in example I of the introduction.

---

[19] Note that the modal operators can and should are not real predicate. They are represented under the modality feature. Their exact behavior in this context is to be defined by the expert system.

[20] The mapping from the facts 1,2,3 to these relations is described below. It makes use of the argumentative strategies.

[21] This is a schematic notation. The complete representation is described in section 7.4.

Section 7.7 discusses how this can be done.

This example illustrates how only the proper argumentative relations can be selected when considering other dimensions of the answer.

6.7 Argumentative Strategies

Argumentative strategies control how facts from the pool of mentionable items given in input to the system can be converted to arguments. Argumentative strategies eventually are controlled by interactional constraints, setting parameters like ‘‘seek-agreement.’’ I will work on two types of strategies:

- The Agreeing Strategy (AS)
- The Disagreeing Strategy (DS)

The module of the system in charge of argumentative constraints needs to find an argumentative relation between the source and a fact that will become part of the argumentative orientation of the target. To do that, the module scans the pool of mentionable items, and tries to map facts on a scale which is related to the scale of the source in some way. This search procedure must be controlled. The task of argumentative strategies is to explicitly control this search. The first step of the search is to find ‘‘relevant scales’’ from the scale of the source. This is done by searching the set of topoi containing the source scale either on the left or on the right. At this stage of the search, AS and DS differ. AS will order the search by looking at only the topoi which include the source scale with the same orientation as it has in the source; DS will first look at opposite orientations. The net effect is that, with AS, the speaker will first try to find reasons to support what is said in the source, while with DS, the speaker will first try to disagree.

Consider the following example:

Q: *Goal: Avoid programming*
Can I take computability instead of fundamental algorithms?

Let’s assume the following facts are in the pool after the expert system has run:

`1. Can( Take(S, Comp))` *# Student can take Computability.*
`2. Can( Take(S, FA))` *# Student can take Fundamental Algorithms.*
`3. Topic( FA, Software)` *# The topic of Fund. Alg. is Software.*
`4. Topic( Comp, Theory)` *# The topic of Computability is Theory.*

And the following topoi are defined:

 # Computability does not require a lot of programming.
`A. {+Take(S, Comp), -Program(S)}`
 # Fundamental Algorithms require a lot of programming.
`B. {+Take(S, FA), +Program(S)}`
 # Soft. classes require a lot of programming.
`C. {+Topic(X, Software) & Take(S, X), +Program(S)}`
 # Theory classes do not require a lot of programming.
`D. {+Topic(X, Theory) & Take(S, X), -Program(S)}`

Given the underlying goal of the student, the argumentative orientation of Q is: -Program(S)[22]

The problem is to map some of the facts 1-4 to arguments related to the argument -Program(S):

I. With AS activated
The following arguments are constructed:

`1′. +Take(S, Comp)` *# You should take computability*
`2′. -Take(S, FA)` *# You should not take Fund. Alg*
`3′. -Topic(FA, Software)` *# Fund Alg is not that much on software...*
`4′. +Topic(Comp, Theory)` *# Computability is a Theory course.*

II. With DS activated
The following arguments are constructed:

`1′′. -Take(S, Comp)` *# You should not take computability*
`2′′. +Take(S, FA)` *# You should take Fund. Alg*
`3′′. +Topic(FA, Software)` *# Fund Alg is a software course*
`4′′. -Topic(Comp, Theory)` *# Computability is not really a Theory course.*

Other more sophisticated heuristics can be used to control the search, but, in this dissertation, the system will be limited to this simple distinction between AS and DS. The reason is that this distinction can be controlled by the interactional constraints. Similar heuristics could be implemented if they can be controlled by some decision made by other local constraints.

This examples illustrates several problems concerning the connection between the explanation module of an expert system and the expert system itself: (1) what is the relation between the topoi of the explanation module and the rules of the expert system? (2) should the orientation of the answer to a question be decided by the expert system or by the explanation module?

There is a clear relation between topoi and expert system rules. Both are inference rules. Note, however, that the topoi presented in the example could hardly be used as is in an expert system. Consider for example the following rule, taken from the current version of the ADVISOR system:

---

[22] A simple test to verify that Q is compatible with this orientation is to consider if the connected sentence ‘‘Can I take computability instead of fundamental algorithms, because I do not want to program too much?’’

```
(RULE-08  CONCENTRATE-ON-AREA-TOPICS
           (IF (superc ?feature-course ?area)
               (topics ?feature-course ?t1)
               (topics ?course ?t2)
               (~ (equal ?feature-course ?course))
               (equal ?t1 ?t2))
           (THEN (advances-plan c-user ?course)))
```

Topoi do not represent the same knowledge as this inference rule. There would be difficult problems to surmount to base an inference system on topoi. Topoi are used to capture two notions: a notion of relevance (when P is mentioned, Q is relevant) (cf. [Sperber & Wilson 86] for a discussion and a more general treatment of relevance) and a notion of correlation of judgement (I judge P the same way as Q). These notions are important in a theory of communication but do not coincide with the notion of inference needed in an expert system. There is, however, very likely, a procedure to derive topoi from inference rules.

As for (2), who of the expert system or of the explanation module should determine the orientation of an explanation? The example shows that depending on which argumentative strategy is activated, explanations of opposite orientation can be generated. Which explanation is most convincing would have to be determined by a more elaborate study of conceptual argumentation. Shouldn't the orientation be decided by the expert system? In general, the distinction between an ‘‘underlying system’’ and a ‘‘generation module’’ is quite arbitrary. The view taken in this work is that the underlying system provides a set of propositions to the explanation module. Another view could be that the expert system provides an ‘‘answer’’ to a problem and the explanation module needs to phrase this answer. Which view is most appropriate depends on engineering factors (how complex is each component, how modular need they be, what type of communication can be established between them) and the domain of application (what type of information is available to the user and to the system, is the precision of an answer important, what is the function of the information provided by the system). In the case of the ADVISOR domain, it seems reasonable to let the explanation module choose the orientation of the text it generates.

7 A Computational Treatment of Decision Making as Interacting Constraint Satisfaction

The picture drawn so far of the model is: several classes of local constraints interpret an input (a description of the structure of the previous conversation and a pool of mentionable items) and interact to construct a representation of a target move. I present in this section a computational treatment of this problem of interaction. The approach is based on the notion of functional unification, which is widely used in computational linguistics, but has generally been restricted to grammatical applications (cf [Dale 88] for an exception). I argue in this section that the properties of functional unification make it a good candidate to solve this interaction problem. In this section, I present the mechanism of functional unification; I then describe how I have extended the formalism to make it a more general tool. I then show how local constraints are represented in the formalism. I finally describe the algorithms controlling the whole system of local constraint application. I only point in this paper to my work on extending functional unification, but it constitutes as a whole a significant contribution of this work.

7.1 Functional Unification

7.1.1 Functional Descriptions

Unification determines if two descriptions of objects d1 and d2 are compatible, and when they are, computes a description for objects that match both d1 and d2. Unification generally refers to the mechanism of ‘‘structural unification.’’ Kay [Kay 79] has introduced the notion of ‘‘functional unification’’ (in the following, FU) to formalize the linguistic notion of ‘‘feature.’’ Structural unification works on first-order terms of the form `P(x,y)`. FU works on *feature structures*. Features structures differ from first-order terms in four ways (cf [Knight 89, p.105]):

- ‘‘Substructures are labeled symbolically, not inferred by argument position.
- Fixed arity is not required.
- The distinction between function and argument is removed.
- Variables and coreference are treated separately.’’

Feature structures or *functional descriptions* (FD) can be viewed as lists of attribute-value pairs, where each value can be an atomic value or recursively, an FD. An attribute-value pair is called a *feature*. The following is an example of FD (in Lisp notation):

```
        ((process build)
         (agent Steve)
         (medium ((concept crane)
                  (weight 2)
                  (height 4))))
```

This FD can be used to represent the fact that ‘‘Steve builds a crane that is 2 lbs and 4 feet high.’’ A corresponding first-order term can be:

```
        build(Steve, Crane(C1, 2, 4))
```
[23]

Contrasting these two notations for the same example illustrates the differences:

- *Symbolic labels for substructures*: the arguments, that is the `agent` and the `medium`, are clearly labeled in the feature structure notation.
- *Fixed arity*: features can be added at will to an FD. FDs are used to represent *partial information*. This is not the case for first-order terms.
- *Function and argument*: first-order terms have a head (the function) which plays a central role in the unification process. This is not the case in FDs. All information plays the same role. In most cases, one feature plays a special role: for example, the `cat` attribute can specify the category or type of a description. But this is not built into the syntax.
- *Variables and coreference*: in standard unification, a variable is used to mean two distinct things: that the value of the role is unknown (there are no constraints on it), and that the value of the role is the same as all other objects referred to with the same variable. So for example, in a term such as `like(X, X)`, the use of the variable `X` means that we don’t know who likes whom, and that we know that the agent and the medium of `like` must be the same objects. The distinction between these two functions of variables is best explained in [Ait-Kaci 84]. In FDs, coreference and unspecification are represented by two different syntactic devices: variables are features which are unspecified (they simply do not appear in the FD, or appear with an empty value), coreference is handled with reference markers in certain implementations, and with *paths* in other. For example, to express the constraint that the medium of `like` must refer to the same object as its agent, the following FD can be used:

```
Path notation:              Reference Marker notation:
   ((process like)          ((process like)
    (agent <medium>))        (agent #1)
                             (medium #1))
```

`<Medium>` is a path: it is best understood as pointer in the FD to the value of the attribute `medium`. So the feature `(agent <medium>)` means that the attribute `agent` must point to the same object as the attribute `medium`. The FD representation of the topos ‘‘the more an object Y is beautiful, the more a person X wants to buy Y’’ given in section 6.2.4 is also an example of the reference marker notation,

---

[23] Other representations are of course possible using first-order notation. Some of them have some of the advantages of features structures. For example: `build(B1), agent(B1, Steve), medium(B1, C1), crane(C1), weight(C1, 2), height(C1, 4).` In fact, any FD can always be translated in a one-to-one mapping to a class of restricted first-order terms.

where X and Y are markers.

The FD notation has turned out to be extremely popular among linguists, and several linguistic formalisms are now based on a notion of unification between feature structures (cf [Shieber 86] for an excellent survey of this field).

7.1.2 The Unification Operation

Unification between FDs can be seen as the operation of enriching one FD with features of the other.

```
FD1 =    ((process build)        FD2 =   ((process build)
          (agent Steve))                  (medium ((concept crane))))

Unify(FD1, FD2) =  ((process build)
                    (agent Steve)
                    (medium ((concept crane))))
```

Note that the FU formalism includes the notion of disjunction of features - which makes it very expressive. In general, FU has been used to describe grammatical constraints [Kay 79, Shieber 86, Elhadad 88]. Therefore, it is common to use the term of ‘‘FU grammar’’ or FUG to refer to a complex FD, often containing disjunctions. FUGs are generally organized around a class of categories: each `cat` is described by a different branch of the FUG.

7.1.3 Constituents and Recursion

FUGs as described in [Kay 79] have introduced the notion of constituent in FDs. A constituent is a special feature of an FD. Constituents must be recursively unified with the FUG. So the general algorithm for unification of an FD FD1 with a grammar G in FUGs is as follows:

1. Unify-top-level FD1 with grammar G.
2. Identify the constituents {Ci} of the result.
3. For each constituent Ci, recursively, unify Ci with G.

where unify-top-level is the simple unification operation, without constituents. Constituents are identified by a special feature called CSET as in the following example:

```
        ((cat clause)
         (agent ((cat np)
                 (lex ``John'')))
         (process ((cat vp)
                   (lex ``laugh'')))
         (cset (agent process)))
```

In this example, the `process` and the `agent` feature are identified as constituents. They will therefore each be unified with the grammar G (actually, the `agent` will be unified with a branch of the grammar specific to noun phrases, and the `process` with the verb phrase branch). Constituents trigger recursion in the unification algorithm and are a powerful mechanism in FUGs.

7.2 Extensions to Functional Unification

In general, the types of constraints between features that are expressed in the standard FU formalism are:

- Equality with a constant is expressed as `(attribute constant)`
- Equality between two features is expressed as `(attribute1 <attribute2>)` (also called conflation).

And these primitive types of constraints can be combined using conjunction, disjunction (for example, to limit the possible values of a feature a disjunction is used `(attribute (alt (val1 ... valn)))`). Although negation is not part of the basic unification formalism, it has been added to many unifiers ([Shieber 86, Karttunen 84]). Constituents and recursion also add to the power of the formalism.

This set of constraint expressions is quite powerful but can prove too limited to handle certain problems. Constraints on sets, for example, are problematic. It is difficult to express anything about sets using the standard FUG formalism. FDs allow values to be either atomic or FDs. Representation of sets is not easy to determine. This is a limitation of the formalism. In addition, there is nothing in the formalism to handle numbers, or operations between them. It is of course possible to write grammars to deal with sets or lists; for example, a grammar to compute the *append* of two lists, test for membership, or compute the intersection of two lists is presented in [Elhadad 88]. Such grammars are, however, inefficient and not very readable. It is more productive to acknowledge the limitation of the formalism, and to add facilities to express more complex constraints.

In [Elhadad 89, Elhadad 90a, Elhadad 90b], I have presented a set of extensions to the basic FU formalism which allows the expression of more complex constraints in a more efficient way than FU alone would allow. These extensions include a mechanism to deal with type hierarchies and inheritance, to use special purpose unification procedures for certain types of features and to declaratively restrict the set of features usable in certain constituents. This also includes a mechanism similar to Guarded Horn Clauses, called `TEST`. `TEST` allows testing of an arbitrary LISP expression that must be satisfied before unifying, but has no effect on the FD being unified.

In summary, the extended formalism is powerful enough to express a large variety of constraints, but keeps enough of the spirit of FU to maintain the overall advantages of the formalism. The extensions to the FU formalism I have designed are described in [Elhadad 90b] and [Elhadad 90a]. They constitute a significant contribution of this work.

7.3 Expression of Local Constraints

Local constraints are relations between a source and a target discourse segment. I represent these relations as complex FDs containing unification constraints. When the description of a source is unified with such an FD, the description of the target gets partially instantiated. As more local constraints are unified with the source, the FD of the target is increasingly refined.

FDs are best viewed as constraints on the object they describe, and unification as the process of merging two sets of constraints. I use this natural interpretation of FU to represent local constraints and manipulate them. Unification is thus the process of adding more constraints (in the form of features to be satisfied) to a description. The choice of FU for expressing local constraints has several advantages: (1) it allows use of the same mechanism all across the implementation; in particular, since FU is the formalism of choice for implementing the surface generator, using FU for handling local constraints allows the surface generator to interact closely with the local constraints on content generation; (2) each local constraint can be expressed separately and combined easily through the use of unification; (3) the unification operation captures the bidirectional nature of the relation source/target, and thus makes it easy to use the notions of uptake and reinterpretation.

Figure 20 illustrates how complex constraints can be expressed in the FU formalism.

This FD is a typical local constraint. It contains constituents for the source and the target, plus one constituent `relater` which is used only for the expression of the constraint. It can be read as follows: a source and a target satisfy this argumentative local constraint if there exists a topos which can relate them, such that the argumentative orientation of the source is #X, the orientation of the target is #Y, and the topos is of the form ‘‘the more #X, the more #Y.’’

Expression of the argumentative constraint [S is an argument for T]:

```
((cat argumentative-lc)
 (source ((cat interpretative-format)
          (argumentative-orientation #X)))
 (target ((cat interpretative-format)
          (argumentative-orientation #Y)))
 (relater ((cat topos)
           (left  #X)
           (left  ((orientation +)))
           (right #Y)
           (right ((orientation +)))))
 (argumentation ((support s-t)))
 (cset (target relater)))
```

Figure 20: A complex constraint in FD notation

Given the way constituents are used in FUGs, this local constraint is unified with a source as follows:

1. The top level local constraint is unified with the source S - and the features `(cat interpretative-format)` and `(argumentative-orientation #X)` is added to the source.
2. The unification now recurses on each constituent: namely, the `target` and the `relater`.
3. The `target` is unified with the branch of the grammar describing well-formed interpretative formats (IFs). The result is an empty legal IF with the feature `(argumentative-orientation #Y)` defined.
4. The `relater` is now unified with the branch of the grammar describing well-formed topos. This FUG works as a sort of lookup table with some inferential power. It contains a disjunction of the known topoi plus some rules of legal transformations on topoi (for example, {+X, +Y} is equivalent to {-X, -Y}). When the `relater` is unified, the feature `left` is already unified with the `argumentative-orientation` of the source through the `#X` marker. The unification will therefore find a topos whose left-hand side can be unified with `#X` and with the signs +/+. The right-hand side of this topos will also become the `argumentative-orientation` of the `target` through the marker `#Y`.

Thus, the unification of this local constraint can be summarized by: fill in the `argumentative-orientation` of the target with a conclusion reachable from the source. It shows how such complex constraints can be represented in the FU formalism by composing low level constraints.

7.4 Definition of Features: Interpretative Formats

In the FU formalism, interaction between local constraints is indirectly managed through the operation of unifying the same features. For example, if both the argumentative and the structural constraints unify a feature called `exchange-complete`, then this feature is the locus of the interaction between these constraints. The choice of features describing discourse segments is therefore critical. I present in this section a set of features called *interpretative formats* (IF) which I use in this work. IFs define a common terminology for all types of constraints.

IFs are used to represent the pragmatic features of a discourse segment. In a complete generation system, IFs are an intermediate representation between the deep and the surface components: all decisions of the deep component are reflected by the value of some feature in the IF, and all decisions of the surface component are conditioned by the value of the features in the IF.

IFs contain the following features (cf [Elhadad & McKeown 88] for a discussion of most of these features):

- *Propositional-Content*: a case-based description of the semantic information conveyed by the segment.

This section contains pointers to an underlying knowledge representation system.

- *Argumentative-Orientation*: a description of the conclusions that the segment supports.
- *Argumentative-Relater*: a description of the type of connection between the propositional content and the argumentative relation.
- *Topic*: a description of the discourse entities used in the segment, structured in terms of focus and familiarity.
- *Thematization-Procedure*: a description of the way certain discourse entities are derived from the propositional content or from other pragmatic feature of the segment.
- *Functional-Status*: in a move, whether the segment is directive or subordinate, in an exchange, whether it is initiative, reactive, or both.
- *Structural-category*: whether the segment is a move, and exchange or an act.
- *Complete*: whether the segment is a complete move or exchange, or whether it needs to be completed.
- *Speech-act*: the speech-act realized in the segment.

Note that this list is not intended to be exhaustive. Many aspects of the semantics of the segment are ignored here - for example, time, aspect, mood and modality. But for the constraints studied in this work, IFs provide most of the information needed.

Figure 21 illustrates how IFs encode a possible interpretation of example III.

```
((Speech-Act
  ((Category Info-Request)
   (Speaker #Student)
   (Hearer  Advisor)))
 (Propositional-Content
  ((process #Take)
   (modal   #Can)
   (agent  ((concept  #Student)))
   (medium ((concept  #Computabiliy)
            (modifier  ((process replace)
                        (agent #Computability)
                        (medium #Fund-alg))))))
 (Argumentative-Orientation
  ((scope ((process #desire)
           (processor #Student)
           (phenomenon ((process take)
                        (agent #Student)
                        (medium ((concept #Computability))))))
   (focus #desire)
   (scale "desire")
   (orientation +)))
 (Topic
  ((entities [#Student #Advisor #Take #Can #Computability #Fund-alg #Desire])
   (focus-state [#Computability [#Can #Take #Student] [#Fund-alg]
                                [#Desire #Advisor]])))
 (Structure
  ((Functional-Status initiative)
   (Category move)
   (Complete yes))))
```

Figure 21: IF for ‘‘Can I take computability instead of fundamental algorithm?’’ in III

This IF can be read as follows: a student asks for some information from the advisor. The information requested is whether the student can take the course *computability* to replace the course *fund-alg*. The conclusion supported by the student is that he wants to take the computability course. This question is about the entities listed under the `topic-entities` list, and the salience of each of these entity is represented by the bracketing in the `topic-focus-state` feature. Note that the values of the features `topic-entities` and `topic-focus-state` are not standard FDs, but special types of data, defined with their own method of unification. Finally, this question is a move with initiative status in a programmed exchange and is a complete segment in itself (it does not need to be continued by the student).

7.5 Algorithm for Constructing a Turn: Control of Unification

So far, all the tools to construct a new turn given a previous conversation have been described piece by piece. In this section, the algorithm that puts everything together is described.

Input to the system consists of: a set of IFs describing the previous conversation (the Prev-Disc); a set of propositions described as FDs and describing the possible content of a new turn. These FDs are similar to the FD appearing under the `Propositional-Content` feature of an IF. This set is called Poss-Cont. The system also has available as a resource a set of local constraints (LCs) and a grammar for the realization of the surface string (FUF). The output of the system is an English clause. The system also updates internally the Prev-Disc to record the production of the new turn.

The mechanism used to produce the new turn is basically to search through the set of all possible unifications of the Pre-Dis with the local constraints until one possible IF can be constructed. When this is achieved, the IF is unified with FUF to produce the English text. Unfortunately, this approach is extremely time-consuming, as the search space is extremely large. Unification of the local constraints therefore needs to be controlled.

First, I assume that the source of the new turn is already determined. The search for the source would be controlled by the discourse structure, which determines what discourse segments are accessible. The source is, in this dissertation, an input to the system.

What remains to be done is to unify this source with all local constraints. To control the unification, an order on the local constraints is defined. This limits the search space because, when more features in the IFs are instantiated, unification becomes ‘‘more’’ deterministic. The ordering does not need to be complete: it is possible to class several local constraints as having the same priority; it is then possible to backtrack among all local constraints of the same priority. (Note that this ordering of local constraints corresponds to what systemists call the degree of delicacy of features.)

The second tool used to limit the search space is to limit the time to spend on each local constraint *a priori*. The functional unifier, as described in section 7.3, recurses on each sub-constituents of an FD. If the unifier is written to traverse the constituents of an FD in breadth-first order, then each recursion of the unifier can be seen as successive refinements of the unification - adding features at the top-level first, and adding ‘‘details’’ at lower levels of each recursion. Therefore, if unification stops after a set limit of backtracking points have been used, the result can be an inconsistent FD, but it can also be a good first approximation of what a more complete unification would have produced. If the limit on the number of backtracking points the unifier can use on each local constraint is set appropriately, one can assume the result of a partial unification can be usable.

It can also happen, unfortunately, that even if the unifier spends a lot of time on one local constraint, the partial result after the limit is reached is completely unusable. There are however quick tests which can determine rapidly if a local constraint has been successful ‘‘a little bit’’ or completely wrong. These tests can be applied when the limit is reached, and based on their result, the partial result can be kept or rejected as a failure.

This technique has already been tested on the surface generator (FUF). The quick tests check whether the output of the unification satisfies certain important syntactic constraints (*e.g.*, all obligatory syntactic roles have been processed, agreement subject-verb is correct). Designing the quick tests is, however, quite hard. Quick tests are implemented as stripped down versions of the grammar (simplified FUGs) which are unified with the output. The results are encouraging for the surface part. Experimentation will be necessary to assess the feasibility of this technique for the local constraints.

In summary, the heuristics to control the unification procedure are expressed by assigning to each local constraint a priority level, an upper-bound on the number of backtracking points it can use, and a ‘‘quick test’’ pattern which determines if a partial result makes sense on the most important constraints or not.

The order which is currently used is the following:

1. Choose the source.
2. Determine possible discourse entities using static inference topic local constraints only.
3. Filter set of propositions using topic local constraints and the set of possible entities.
4. Find all relevant arguments by mapping relevant contents to arguments related to the scale of the source using argumentative local constraints and argumentative strategies.
5. Choose argumentative orientation using argumentative local constraints. May need to use thematization procedures using topic local constraints.
6. Choose speech-act using illocutionary local constraints. May need to use thematization procedures using topic local constraints.
7. Choose functional status using structural local constraints.

The idea behind this ordering is to try to generate ‘‘normal unmarked’’ turns as directly as possible. This ordering of decisions tends to follow the topics introduced in the source and introduces new topics only when already mentioned topics cannot be used. It is therefore expected to make an answering system behave very ‘‘literally’’ and reactively, with not much initiative. The eventual ordering I will use is not yet determined and will be refined after experimenting with an implementation. Different orderings would very likely correspond to different interpersonal strategies.

7.6 Content Planning

Decisions made by the system about the content of the turn are primarily a side-effect of applying the local constraints.

Content filtering from the pool of information is triggered by the topic local constraints and, indirectly, the argumentative local constraints. Topic local constraints determine a set of accessible discourse entities, and only the propositions containing some of these entities as participants can be selected. Topic local constraints interact with the argumentative local constraints through the use of the argumentative thematization procedure. The role of this procedure is to force relevant facts to act as arguments related to the argumentative orientation of the source. This procedure has already been presented in section 2. It is repeated here with more details:

1. Relevant scales: Find all scales related to the scale of the argumentative orientation of the source through one topos. Call this set the ‘‘relevant scales.’’ For example, from the scale `nice` in the source, the scales `desirable, expensive` are also selected (this depends on the set of topoi defined in the system).
2. Relevant concepts: Map from the relevant scales to a set of relevant concepts which are related to the scales. For example, given the scale `temperature`, the concept `c-temperature` and its specializations in the knowledge base are added to the list of relevant concepts.

3. Relevant propositions : Filter the pool of potential content to keep only those propositions containing one of the relevant concepts as one of their participants - this forms the set of relevant propositions.
4. Relevant arguments: For each of the relevant propositions, map from the propositional content to an argument: this involves mapping from a conceptual scale to a linguistic scale, and making certain lexical decisions (from the concept `temperature-50F`, decide to use the adjective ‘‘cold’’ or ‘‘hot’’ for example). This also determines orientation for the argument. The resulting set of partially lexicalized propositions is the set of relevant arguments.
5. Potential Arguments: Filter the set of relevant arguments according to the argumentative strategy in use. This means, keep only the arguments that either support or attack the source. This produces the possible argumentative orientations of the target.
6. Potential Propositions: For each possible argumentative orientation, find a proposition that can express it under the constraints of directness dictated by the interactional local constraints: if the constraint is to `be direct`, use the scope of the argumentative orientation as a content for the target; otherwise, find an argument related to the conclusion that can be easily expressed (the conclusion is implicit). For example, if the `be direct` strategy is in use, the sequence ‘‘Yes I have the same numbers but our instruction sets are different’’ would be generated instead of (10); that is, the conclusion is explicitly stated. Otherwise, a sequence like ‘‘Yes, I have the same numbers, but there are pictures on my instruction sets’’ would be generated; that is, the conclusion is made implicit, and only an argument for it is stated explicitly.

The decision to attach the new proposition to the source as a directive or subordinate turn is made by the structural local constraints.

It is important to relate this procedure to existing work in content planning in text generation. As discussed in Section 3.1, the major techniques used are: schemas, planning and discourse grammars. Of these approaches, the technique discussed here is closest to planning. It relies on topoi (argumentative relations) in a manner similar to planning operators to establish a semantic connection between propositions. However, the main difference is that this procedure does not address the same need: it does not aim at selecting and structuring the content of an entire paragraph or discourse; rather, it is used on top of an existing content generator (*i.e.*, a schema, planner or discourse grammar) that can produce a pool of relevant propositions. The goal is only to filter and organize this pool. It is therefore conceivable that this technique be combined with one of the three other techniques. The advantage is that it would relieve the ‘‘content generator’’ from the task of organizing its output, and, more importantly, make fine decisions about what to verbalize depending on the local context.

7.7 Conflict Resolution

Conflict resolution is a complicated problem. Unification is a non-deterministic process, and it may happen that many conflicting local constraints can be applied to the current discourse state. In this case, a conflict resolution mechanism should decide what local constraint to use, and in what order the local constraints should be applied.

For example, consider again example 5 of the introduction:

Example 5

(10)	A1.	I have a description on the back of my instruction set with the numbers 2 and 3 in circles.
	A2.	Do you have the same numbers?
	B3.	*Yes, the numbers are there*
	B4.	*but there are pictures of the crane on mine.*

The answers B3 and B4 satisfy different sets of constraints: B3 satisfies the illocutionary constraints (provide-info after a query-info) and the topic constraints at the semantic level (keep focus on numbers). But it also conveys an argumentative orientation opposite to the one in B4. B4 would not satisfy the illocutionary constraint if it had to come just after A2. There is here a conflict between the illocutionary local constraint and the argumentative local constraint. In (10), this conflict is solved by building a complex turn of the form ‘‘p but q,’’ where p satisfies the illocutionary constraint, and its argumentative orientation is ‘‘revised’’ by q. Note also that p is assigned a subordinate status in the complex ‘‘p but q’’ and can therefore be viewed only as a ‘‘filler’’ used to satisfy some transition constraints.

Possible solutions include constraint prioritization and conflict set merging. If each local constraint is given a priority, it is possible to order the candidate targets in the conflict set. The rank of each target is obtained by combining the priority of each local constraint it matched plus the quality of the match (that is how much information was in the input and how much information has been added by the local constraint). This ranking can be used to choose among the candidate targets.

The technique of conflict set merging is even more promising. As shown in the example, the different targets in the conflict set can be merged together, using new local constraints to create a more complex move. In this case, one of the candidates becomes the source, and the process tries to adjoin the other candidates as targets related to this source.

Obviously, the two techniques need to be combined: the highest-ranking candidate should probably become the source for conflict set merging.

Consider the example outlined in Section 6.6. Two possible targets were determined for a given question:

```
Q: Goal: concentrate on AI   Should I take DS?

T1: Yes, you should take DS.
T2: DS is a prerequisite for Intro to AI.

T3: Yes, you should take DS because it is a prerequisite for Intro to AI.
```

One way to resolve the conflict between T1 and T2 is to create a complex move containing both T1 and T2. Thus, considering that T1 is a source, we try to find if T2 can be related to T1 in an appropriate manner. This creates the answer T3.

There is another reason to use a technique similar to conflict set merging: if no content can be selected that satisfy all the classes of local constraints, it may be possible to create a complex move that as a whole can satisfy all local constraints. An example of this case is given in example (10) above. The ‘‘p but q’’ move as a whole satisfies all classes of local constraints, but neither p nor q alone can fit.

A final reason to use one of the candidate targets as a source for expansion is that it may be good to provide more information when possible. If the pool of propositions contains 20 propositions and only one is used in the answer, there is a sense of ‘‘waste’’. One may want to provide more information when possible, even if it is not directly related to the question.

7.8 Surface Realization

Once the IF for the target has been constructed, it needs to be realized as English text. The surface component of the system in charge of this task is called FUF. It is a FUG with a good coverage of English syntax that follows the syntactic descriptions provided in [Halliday 85, Quirk *et al* 72] and [Winograd 83, Appendix B]. The main originality of FUF is that it takes the features of the IF into account to make syntactic decisions. FUF is described in detail in [McKeown&Elhadad 90].

The grammar is separated in different sections handling different types of features in the input IF: content, argumentative, structural, illocutionary and interactional. The interactional section is not developed at this point.

The grammar is also separated in different categories - each category being separated in five sections. The categories currently covered are:

- Clause complex: a conjunction or subordination of a main clause and another subordinate or conjunct clause. A clause complex generally contains a connective.
- Clause: a single clause with one verb and its inherent case roles plus many circumstantial participants (*e.g.*, at-time, purpose, at-loc).
- Verb group: the main verb, with auxiliaries, marks of negations or intensification, modals and mood modifiers.
- Noun group: can include a relative clause, a complex determiner group, and modifiers.
- Prepositional group: a preposition followed by a noun group.
- Adjectival group: used as modifiers of noun groups.
- Adverbial group: used as modifier of the clause.
- Determiner group

At this point, FUF has a good description of the clause complex, with a description of the connectives ‘‘because,’’ ‘‘since,’’ ‘‘but,’’ and ‘‘although.’’ Both conjunction and subordination are covered. At the clause level, the content region includes the transitivity system, that maps semantic roles (e.g., agent or medium) to syntactic roles (e.g., subject or object). It also includes the focus system, which can determine the voice of the clause (active or passive), the order of the participants (*e.g.*, put the purpose role in front to present it as new information, as in ‘‘In order to clear the display, push the CLR button’’ as opposed to ‘‘push the CLR button to clear the display’’), and when cleft is to be used (‘‘It is John who came’’ as opposed to ‘‘John came’’).

At the nominal group level, FUF determines whether to express modifiers as describers (adjectives coming before the head noun) or as qualifiers (relative clauses, appositions or prepositional groups coming after the nouns. When the noun group includes a classifier (as in ‘‘a holding battery’’ or ‘‘a science book’’), FUF decides whether to include the classifier based on the focus features. At this point FUF does not do pronominalization. I will work on some rules of pronominalization based on Dale’s techniques [Dale 88] which would all reside in the focus region. FUF also handles the ordering of several describers (do you say ‘‘a big red box’’ or ‘‘a red big box’’).

The determiner group is rather complex and relies on the argumentative local constraints for the expression of quantity. Quantifiers and modifiers like ‘‘all,’’ ‘‘most’’ or ‘‘almost’’ have a gradual meaning which makes them good candidates for an argumentative analysis. It is not yet done.

The verb group is the main tool to express mood and modality. It is influenced mostly by illocutionary local constraints, for the decisions on mood, and by the interactional local constraints for the decisions on hedging and uses of modals as an indirection device. Currently the verb group is minimal. It supports modals and adverbs only minimally. Negation is also not supported. Negation is a vast problem, and I will address it only minimally. FUF supports phrasal and prepositional verbs (multi-word verbs like ‘‘back up’’ or ‘‘rely on’’).

8 Implementation

In this section I describe the implementation effort that will support this work. It is separated into two sections: I first describe general purpose tools which have to be developed to implement the algorithm defined above; in particular a general purpose functional unifier. Then I describe the application of local constraints to the ADVISOR question answering system.

8.1 Tools

I have developed two general purpose tools to support the implementation of this work: first a general purpose functional unifier, called FUF; second a domain independent grammar for surface generation using FUF.

FUF is a functional unifier written in Common Lisp. It includes several extensions to the basic formalism of FUGs and debugging facilities. The current version of FUF (3.0) has been extensively tested and debugged over a period of two years by a large number of users. FUF is used in many different projects at Columbia, and is also used at other research cites doing natural language generation. At this point, it is a rather robust system.

FUF is quite fast, making the use of the FUG formalism practical. With the current version of the program and of the grammar, FUF can generate a complex sentence in less than one second on a SUN 3 machine. The run time of FUF is roughly proportional to the number of backtracking points that need to be tried during the unification. On a SUN 3 machine, the CPU time required per backtracking point is 9 ms; the average number of backtracking points for a simple clause is around 300; for a complex clause it can go up to 3,000. This timing includes all the tracing code. I am working on an improved version which should reduce these measures by a factor of 40%. This means that FUF can work at a speed of up to 200 bp/s (backtracking points) on a SUN 3/60. Although this does not compare favorably with the speed of a Prolog compiler running at up to 10Klips on the same hardware, it must be recalled that functional unification is much harder than the structural unification used in Prolog which relies heavily on the fixed arity of terms for optimization. It should also be noted that compiling FUGs is possible [Hirsh 88].

FUF includes a facility to limit the number of backtracking points allowed in a single unification. This is used as a control device in the algorithm described above. This means FUF can do ‘‘resource limited’’ processing. This turned out to be a major help in debugging grammars. FUF also includes a ‘‘loop spy’’ which tries to detect if the grammar is caught in an infinite loop. FUF supports various tracing methods which read in tracing indications put in the grammar.

FUF comes with a library of functions to manipulate FDs. This includes functions to access values at any level, modify or delete features, filter features based on complex tests. It also includes facilities to express lists as FDs. FUF works as a generator of all possible unifications but can also be used to generate only a given number of unifications. This feature is useful to work on conflict resolution.

FUF implementation is based on the union-find algorithm. The idea is to maintain equivalence classes of all paths in the grammar which must refer to the same FD. Union-Find allows to maintain these classes in an efficient fashion. The other main problem of functional unification implementations is to limit the amount of copying done. FUF uses the technique of undoable physical modifications to limit copying.

FUF includes several extensions to the FU formalism. In particular, it can deal with hierarchical classification of symbols. The extensions are described in detail in [Elhadad 89, Elhadad 90a]. FUF also allows the `TEST` special feature to call arbitrary Lisp functions during the unification procedure. In this way, FUF can be used as a general purpose tool that can be integrated with a complex Lisp environment.

The second general purpose tool is the surface grammar developed to be part of the system. The current grammar can be used as a stand-alone language generation system in association with FUF. The coverage of the grammar has been described above.

8.2 Application

The ADVISOR system has been described in section 2. In this section, I evaluate what implementation needs to be done to make the system run.

The knowledge base will be rewritten using a more reliable environment than the current one (either LOOM or a FUF-based system). The expert system inference engine can be used as it currently stands. The rule base will be rewritten to include students' goals (such as `avoid programming` or `limit amount of work`). The domain plans and recognized user domain goals will be accordingly modified. The goal inferencer will also need to be changed to recognize these new goals. The ATN parser and the semantics interpretation module are working and will not need much modification. The expert system will be modified to produce the pool of relevant propositions (derived from the trace of fired rules) in an appropriate formalism.

Within the surface explanation module, which is the focus of this work, the following modules need to be implemented: source selection, local constraint application, conflict resolution and surface generator:

1. Source selection will be implemented by first determining the right border of the global discourse model (cf [Webber 88] for a discussion). This determines a set of source candidates. To select one source out of this set I will implement simple heuristics.
2. Local constraint application will be implemented on top of FUF. Each class of local constraints will be implemented as a separate FUG. There will be different procedures for controlling the overall flow of control within each class. A separate procedure will handle interaction among the different classes of constraints. It will also be built on top of FUF.
3. Conflict resolution: I will implement constraint prioritization and constraint merging.
4. Surface generation: I have already implemented a grammar. I need to extend the section on determiners and adjectives in the noun phrase section. I will also implement simple rules of pronominalization.

9 Limitations

I list in this section the main limitations of this work.

1. Connection between the global structure of discourse and local constraints: Local constraints form a complex network of relations between discourse segments. There also exists a global discourse structure. I will not determine the connection between these two aspects of discourse structure. This has many implications:
   - The global discourse structure is related to the semantic structure of the domain (what type of semantic relations are described in the domain, *e.g.*, cause, logical implication, temporal order). The relation between these relations and the local constraints is also not clear.
   - Long turns and monologues tend to follow the domain structure. I will limit my work to the generation of short turns and will not determine under what conditions long turns should be used.
   - The work is limited to reactive moves. Initiative moves are generated to reach a goal and follow a plan. Since I do not study the relation between this goal or plan and local constraints, the current system will not be able to generate initiative moves.
   - I will only use simple heuristics to determine the source of a new turn. A better understanding of the relation between local and global structure could lead to a better method of selection. I assume a global discourse model can produce a list of accessible discourse segments. But it is not sure a ''best'' source can be chosen out of this list.
2. Connection between argumentative relations and domain relations: I use a list of topoi to represent and compute possible argumentative relations between propositions. These topoi are similar to rules of the underlying domain, and could be derived from domain knowledge. I will not work on this aspect and assume that topoi are already defined. This affects the way the mapping from propositions

to argument will be done.

3. No implementation of the interactional local constraints: I assume some interactional strategies are set, controlling directness, for example. I do not justify the use of these strategies except that they are used in [Brown & Levinson 87]. I do not maintain a model of the social relations between participants.

4. No implementation of a complete interpretation component: I assume some text understanding system will be able to produce the features needed in an interpretative format. It is not clear whether this can be done given the current state of the art in text understanding. If the features used in this work prove useful to maintain a model of an ongoing conversation, then this work will set a goal for a text understanding program to fulfill, defining what it means to successfully interpret an utterance.

5. Limitations in syntactic coverage: in the surface realization component, I will limit the selection of adjectives to gradable adjectives and will not work on the problems of ambiguity in scoping when quantification is used in the determiner group.

10 Schedule

- March to June 1990:
    - Gather all code for ADVISOR, organize all parts, make the expert system, parser and goal inferencer work on current knowledge base.
    - Create new knowledge base, define scales for the domain and set of topoi from the rule base. Determine precisely format of pool of propositions.
    - Implement structural, illocutionary and topic LCs.
    - Implement argumentative LCs and content selection.
    - Implement conflict resolution.
- Summer 1990: Write grammar for determiners. Do simple pronominalization. Work on adjective selection in FUF. Debug clause and connective sections of the current grammar.
- Fall 1990 ...:
    - Finish implementation. Run examples. Debug.
    - Write dissertation (Nov 1 - Nov 7).

11 Conclusion

This work introduces a new notion, that of local constraint, and shows its use to address the problem of text generation in the context of a conversation. This is a novel approach to the problem of explaining coherence and generating coherent text. Previous work in text generation has failed to systematically distinguish between global organizational coherence and local phenomena. This work in contrast focuses only on the aspects of conversation that are locally managed. Local constraints are classified into five categories and the relation between discourse segments is explained by the interaction among these categories. This leads to an easier description of discursive constraints. In particular, I focus on argumentative constraints, and show the connection between argumentation and low level linguistic decisions.

Practically, this theoretical study of coherence leads to a promising solution to many of the problems of managing coherent interactions with users of knowledge-based software systems. I will apply the model of local constraints to the domain of explanation generation for an advising expert system.

This work also illustrates one of the first uses of functional unification for non-syntactic processing. I have defined extensions to the FU formalism which make it possible to use it as a constraint interaction engine.

The surface text generator defined in this work will include a large and sophisticated grammar and will be able to choose among the different linguistic options offered by this grammar in a motivated way, to construct text that is appropriate in context.

This work is a step towards a better understanding of the nature of relations existing between discourse segments. These relations can be described as semantic, rhetorical or as local constraints. The connection between these different perspectives remains an intriguing problem.

References

[Ait-Kaci 84] Ait-Kaci, H.
A Lattice-theoretic Approach to Computation Based on a Calculus of Partially Ordered Type Structures.
PhD thesis, University of Pennsylvania, 1984.
UMI #8505030.

[Allen & Perrault 80]
Allen, J.F. & Perrault, C.R.
Analysing Intentions in Utterances.
Artificial Intelligence 15(1):pages 143-178, January, 1980.

[Anscombre & Ducrot 83]
Anscombre, J.C. & Ducrot, O.
Philosophie et langage: *L'argumentation dans la langue.*
Pierre Mardaga, Bruxelles, 1983.

[Appelt 85] Appelt, D.E.
Planning English Sentences.
Cambridge University Press, Cambridge, England, 1985.

[Atkinson & Heritage 84]
Atkinson, D. and Heritage.
Structures of Social Action: Studies in Conversation Analysis.
Cambridge University Press and Maison des Sciences de l'Homme, Cambridge and Paris, 1984.

[Auchlin 86] Auchlin, A.
Complementarite des structures thematiques et fonctionnelles pour l'acces aux interpretations dans le discours.
Cahiers de Linguistique Francaise (7):pages 169-188, 1986.

[Austin 62] Austin, J.L.
How to do Things with Words.
Harvard University Press, Cambridge, Mass., 1962.

[Banfield 82] Banfield, A.
Unspeakable Sentences: Narration and Representation in the Language of Fiction.
Routledge and Kegan Paul, Boston, 1982.

[Birnbaum 82] Birnbaum, L.
Argument Molecules: a Functional Representation of Argument Structure.
In *Proceedings of the 2nd AAAI Conference*, pages 63-65. AAAI, 1982.

[Bolinger 67] Bolinger, D.
Adjective Comparisons: A Semantic Scale.
Journal of English Linguistics 1:2-10, 1967.

[Bolinger 72] Bolinger, D.
Degree Words.
Mouton, The Hague, 1972.

[Bowers and Churcher 88]
Bowers, J. and Churcher, J.
Local and Global Structuring of Computer Mediated Communication: Developing Linguistic Perspectives on CSCW in COSMOS.
In *CSCW: Proceedings of the Conference of Computer Supported Cooperative Work*, pages 125-139. ACM SigCHI and SigOIS, Portland, OR, September, 1988.

[Brown & Levinson 87]
Brown, P. & Levinson, S.C.
Studies in Interactional Sociolinguistics 4: *Politeness : Some universals in language usage.*
Cambridge University Press, Cambridge, 1987.

[Cohen 83] Cohen, R.
A Computational Model for the Analysis of Arguments.
PhD thesis, University of Toronto, Department of Computer Science, October, 1983.
Tech Report No. 151.

[Cohen 84] Cohen, R.
A computational theory of the function of clue words in argument understanding.
In *Coling84*, pages 251-258. COLING, Stanford, California, July, 1984.

[Cohen 87] Cohen, R.
Analysing the structure of argumentative discourse.
Computational Linguistics 13(1-2):11-24, 1987.

[Conklin and Begeman 88]
Conklin, J. and Begeman, M.L.
gIBIS: A Hypertext Tool for Exploratory Policy.
In *CSCW: Proceedings of the Conference of Computer Supported Cooperative Work*, pages 140-152. ACM SigCHI and SigOIS, Portland, OR, September, 1988.

[Dale 88] Dale, R.
Generating referring expressions in a domain of objects and processes.
PhD thesis, University of Edinburgh, 1988.

[Dispaux 84] Dispaux, G.
La Logique et le Quotidien: Une Analyse Dialogique des Mechanismes d'Argumentation.
Les Editions de Minuit, Paris, 1984.

[Elhadad 87] Elhadad, M.
A Collection of Four Transcripts of Long Conversations through Computers.
Technical Report CUCS-307-87, Columbia University, New York, 1987.

[Elhadad 88] Elhadad, M.
The FUF Functional Unifier: User's manual.
Technical Report CUCS-408-88, Columbia University, June, 1988.

[Elhadad 89] Elhadad, M.
Extended Functional Unification ProGrammars.
Technical Report CUCS-420-89, Columbia University, 1989.

[Elhadad 90a] Elhadad, M.
Types in Functional Unification Grammars.
In *To appear in proceedings of ACL'90*. Pittsburgh, 1990.

[Elhadad 90b] Elhadad, M.
A Set-theoretic Semantics for Extended FUGs.
Technical Report CUCS-90, Columbia University, 1990.

[Elhadad & McKeown 88]
Elhadad, M. and McKeown, K.R.
What do you need to produce a 'but'.
Technical Report CUCS-334-88, Columbia University, January, 1988.

[Elhadad & McKeown 89]
Elhadad, M. and McKeown, K.R.
A Procedure for the Selection of Connectives in Text Generation: How Deep is the Surface?.
Technical Report CUCS-419-89, Columbia University, 1989.

[Elhadad & McKeown 90]
Elhadad, M. and K.R. McKeown.
Generating Connectives.
Technical Report, Columbia University, 1990.

[Flowers *et al* 82] Flowers, M., McGuire, R., and Birnbaum, L.
Adversary arguments and the logic of personal attacks.
Strategies fornatural language processing.
Lawrence Erlbaum, Hillsdale, NJ, 1982, pages 275-294.

[Fox 87] Fox, B.A.
Interactional Reconstruction in Real-time Language Processing.
Cognitive Science 11:365-387, 1987.

[Grosz 77] Grosz, B. J.
Representation and use of focus in a system for understanding dialogs.
In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence*. Morgan Kaufmann, Los Altos, CA, 1977.

[Grosz 86] Grosz, B.J.
A Theory of Discourse Structure.
In *Proceedings of the 8th Conference of the Cognitive Science Society*, pages 432-441. 1986.

[Grosz & Sidner 86]
Grosz, B. and Sidner, C.
Attentions, intentions, and the structure of discourse.
Computational Linguistics 12(3):175-204, 1986.

[Grosz, Joshi & Weinstein 83]
Grosz, B.; Joshi, A.K. and Weinstein, S.
Providing a Unified Account of Definite Noun Phrases in Discourse.
In *21th Annual Meeting of the Association for Computational Linguistics*, pages pages 44-50. ACL, Cambridge, Mass., June, 1983.

[Halliday 85] Halliday, M.A.K.
An Introduction to Functional Grammar.
Edward Arnold, London, 1985.

[Hirschberg & Litman 87]
Hirschberg, J. and Litman, D.
Now let’s talk about Now: identifying clue phrases intonationally.
In *Proceedings of the 25th Conference of the ACL*, pages 163-171. Association for Computational Linguistics, 1987.

[Hirsh 88] Hirsh, Susan.
P-PATR: A Compiler for Unification-based Grammars.
Natural Language Understanding and Logic Programming, II.
North Holland, Amsterdam, 1988, pages 63-77.

[Hirshberg 85] Hirshberg, J.B.
A Theory of Scalar Implicature.
PhD thesis, University of Pennsylvania, 1985.

[Hobbs 85] Hobbs, J.R.
On the Coherence and Structure of Discourse.
Technical Report CSLI-85-37, Center for the Study of Language and Information, October, 1985.

[Hobbs & Agar 85]
Hobbs, J.R. and Agar, M.H.
The Coherence of Incoherent Discourse.
Technical Report CSLI-85-38, Center for the Study of Language and Information, October, 1985.

[Horn 72] Horn, L.R.
On the Semantic Properties of Logical Operators in English.
PhD thesis, University of California at Los Angeles, 1972.

[Hovy 88] Hovy, E.H.
Planning Coherent Multisentential Text.
In *Proceedings of the 26th ACL Conference*, pages P.163-169. ACL, Buffalo, June, 1988.

[Huebler 83] Huebler, A.
Pragmatics and Beyond. Volume IV:6: *Understatements and Hedges in English.*
John Benjamins Publishing Company, Amsterdam, 1983.

[Jayez 88] Jayez, J.
L'inference en langue naturelle.
Hermes, Paris, 1988.

[Joshi & Weinstein 81]
Joshi, A.K. and Weinstein, S.
Control of Inference: Role of some Aspects of Discourse Structure - Centering.
In *Proceedings of IJCAI*, pages 385-387. Vancouver, Canada, 1981.

[Kamp 84] Kamp, H.
A Theory of Truth and Semantic Representation.
Truth, Interpretation and Information.
Floris, 1984, pages 277-322.

[Karttunen 84] Karttunen, L.
Features and Values.
In *Coling84*, pages 28-33. COLING, Stanford, California, July, 1984.

[Kay 79] Kay, M.
Functional Grammar.
In *Proceedings of the 5th meeting of the Berkeley Linguistics Society*. Berkeley Linguistics Society, 1979.

[Kay 87] Kay, Paul.
Even.
July, 1987.
University of California, Berkeley.

[Knight 89] Knight, K.
Unification: a Multidisciplinary Survey.
Computing Surveys 21(1):93-124, March, 1989.

[Levin and Moore 77]
Levin, J. A. and Moore, J. A.
Dialogue-Games: Meta-Communication Structures for Natural Language Interaction.
Cognitive Science 1(4), 1977.
also: Technical Report No ISI/RR-77-53, Information Science Institute, Marina del Rey, Ca, 1977.

[Levinson 83] Levinson, S.C.
Pragmatics.
Cambridge University Press, Cambridge, England, 1983.

[List 84] List, K.L.
Coherence and Cohesion: contextualization of Oswald Ducrot's general theory of linguistic semantics.
PhD thesis, University of Michigan, 1984.

[Litman 86] Litman, D.J.
Linguistic Coherence : a Plan-Based Alternative.
In *24th Annual Meeting of the Association for Computational Linguistics*, pages pages 215-223. ACL, Columbia University, June, 1986.

[Mac Gregor & Brill 89]
Mac Gregor, Robert and David Brill.
LOOM Reference Manual.
Technical Report, USC-ISI, 4676 Admiralty Way, Marina del Rey, Ca 90292-6695, 1989.

[Mann 84] Mann, W.C.
Discourse Structure for Text Generation.
Technical Report ISI/RR-84-127, Information Sciences Institute, February, 1984.
4676 Admiralty Way, Marina del Rey, California 90292-6695.

[Mann & Thompson 87]
Mann, W.C.
Antithesis: A Study in Clause Combining and Discourse Structure.
Technical Report ISI/RR-87-171, Information Sciences Institute, 4676 Admiralty Way, Marina del Rey, California 90292-6695, April, 1987.

[Matthiessen 88] Matthiessen, C.M.I.M.
What's in NIGEL: Lexicogrammatical Cartography.
Technical Report, USC/ISI, 4676 Admiralty Way, Marina Del Rey, CA 90292, 1988.

[Matthiessen & Thompson 87]
Matthiessen, C.M.I.M. and S.A. Thompson.
The Structure of discourse and subordination.
Technical Report ISI/RS-87-183, Information Sciences Institute, 4676 Admiralty Way, Marina Del Rey, CA 90292-6695, April, 1987.
Also published in *Clause Combining in Grammar and Discourse*, Haiman & Thompson eds, Benjamins, Amsterdam, 1986.

[McCoy & Cheng 88]
McCoy K.F. and J. Cheng.
Focus of Attention: Constraining what can be Said Next.
Presented at the Workshop on Natural Language Generation.
Forthcoming, 1988.

[McGuire *et al* 81]
McGuire, R., Birnbaum, L. and Flowers, M.
Opportunistic Processing in Arguments.
In *Proceedings of the 7th IJCAI*, pages 58-60. IJCAI, 1981.

[McKeown 85] McKeown, K.R.
Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text.
Cambridge University Press, Cambridge, England, 1985.

[McKeown 88] McKeown, K.R.
Generating Goal Oriented Explanations.
International Journal of Expert Systems 1(4):377-395, 1988.

[McKeown&Elhadad 90]
McKeown, K. and M. Elhadad.
A Contrastive Evaluation of Functional Unification Grammar for Surface Language Generators: A Case Study in Choice of Connectives.
Natural Language Generation in Artificial Intelligence and Computational Linguistics.
Kluwer Academic Publishers, 1990.
CUCS-407-88.

[Merritt 76] Merritt, M.
On Questions Following Questions (in Service Encounters).
Language in Society 5(3):315-357, 1976.

[Miller & Rennels 88]
Miller, P.L. and G.D. Rennels.
Prose Generation from Expert Systems.
AI Magazine 9(3):37-44, 1988.

[Moeschler 85] Moeschler, J.
LAL: *Argumentation et Conversation: Elements pour une analyse pragmatique du discours.*
Hatier-Credif, Paris, 1985.

[Moeschler 86] Moeschler, J.
Connecteurs pragmatiques, lois de discours et strategies interpretatives: parce que et la justification enonciative.
Cahiers de Linguistique Francaise (7):pages 149-168, 1986.

[Moore & Paris 89]
Moore, J.D. and C.L. Paris.
Planning Text for Advisory Dialogues.
In *Proceeding 27th ACL*, pages 203-211. ACL, Vancouver, BC, June, 1989.

[Nolke 83] Nolke Henning.
Etudes Romanes de l'Universite de Copenhague: *Les adverbes paradigmatisants: Fonction et analyse.*
Universite de Copenhague, 1983.
Revue Romane, special issue 23.

[Paris 87] Paris, C.L.
The Use of Explicit User models in Text Generation: Tailoring to a User's level of expertise.
PhD thesis, Columbia University, 1987.

[Polanyi & Scha 83]
Polanyi, L. and Scha, R.
On the Recursive Structure of Discourse.
Connectedness in Sentence, Discourse and Text.
Tilburg University, 1983, pages 141-178.

[Pollack 86] Pollack, M.E.
A Model of Plan Inference that Distinguishes between the Beliefs of Actors and Observers.
In *24th Annual Meeting of the Association for Computational Linguistics*, pages pages 207-214. ACL, Columbia University, June, 1986.

[Pomerantz 75] Pomerantz, A.
Second Assessments: a Study of some Features of Agreements/Disagreements.
PhD thesis, University of California, Irvine, 1975.

[Pomerantz 84] Pomerantz, A.
Agreeing and Disagreeing with Assessments: some Features of Preferred/Dispreferred Turn Shapes.
Structures of Social Action: Studies in Conversation Analysis.
Cambridge University Press, Cambridge, 1984, pages 152-163.

[Prince 81] Prince, E.F.
Toward a Taxonomy of Given-New Information.
Radical Pragmatics.
Academic Press, New York, 1981, pages 223-256.

[Quirk *et al* 72] Quirk, R. *et al*.
A Grammar of Contemporary English.
Longman, 1972.

[Raccah 87] Raccah, P.Y.
Modelling argumentation and modelling with argumentation.
Argumentation , 1987.

[Reichman 85] Reichman, R.
Getting computers to talk like you and me: discourse context, focus and semantics (an ATN model).
MIT press, Cambridge, Ma, 1985.

[Roulet *et al* 85] Roulet, E. *et al*.
L'articulation du discours en francais contemporain.
Berne, Lang, 1985.

[Sacks, Schegloff & Jefferson 78]
Sacks, H. Schegloff, E.A. and Jefferson, G.
A Simplest Systematics for the Organization of Turn Taking in Conversation.
Studies in the Organization of Conversational Interaction.
Academic Press, New York, 1978.
also in *Language*, 1974, 50, 696-735.

[Schegloff 72] Schegloff, E.
Sequencing in Conversational Opening.
Directions in SocioLinguistics.
Holt Rinehart & Wilson, New York, 1972, pages 346-380.

[Schegloff 79] Schegloff, E.
Identification and recognition in telephone conversation openings .
Everyday Language: Studies in Ethnomethodology.
Irvington, New York, 1979, pages 23-78.

[Schegloff & Sacks 73]
Schegloff, E. and H. Sacks.
Opening up closings.
Semiotica 7(4):289-327, 1973.

[Schegloff, Jefferson & Sacks 77]
Schegloff, E.A., Jefferson, G. and Sacks, H.
The preference for self-correction in the organization of repair in conversation.
Language 53:361-382, 1977.

[Searle 75] Searle, J. R.
Indirect Speech Acts.
In P. Cole and J. L. Morgan (editor), *Syntax and Semantics*. Academic Press, New York, 1975.

[Searle & Vanderveken 85]
Searle, J. and Vanderveken, D.
Foundations of Illocutionary Logic.
Cambridge University Press, Cambridge, 1985.

[Shieber 86] Shieber, S.
CSLI Lecture Notes. Volume 4: *An introduction to Unification-Based Approaches to Grammar*.
University of Chicago Press, Chicago, Il, 1986.

[Sidner 79] Sidner, C.L.
Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse.
PhD thesis, MIT, 1979.

[Sidner and Israel 81]
Sidner, C. and Israel D.
Recognizing Intended Meaning and Speakers' Plans.
In *Proceedings of the IJCAI*. International Conferences on Artificial Intelligence, August, 1981.

[Sinclair & Coulthard 75]
Sinclair and Coulthard.
Towards an Analysis of Discourse.
Oxford University Press, Oxford, England, 1975.

[Sperber & Wilson 86]
Sperber, D. and D. Wilson.
Relevance - Communication and Cognition.
Basil Blackwell, Oxford, UK, 1986.

[Sycara 89] Sycara, K.P.
Argumentation: Planning Other Agents' Plans.
In *Proceedings of the 11th IJCAI*, pages 517-523. IJCAI, Detroit, MI, 1989.

[Taylor and Cameron 87]
Taylor, T.J. and Cameron, D.
Language & Communication Library. Volume 9: *Analysing Conversation: Rules and Units in the Structure of Talk.*
Pergamon Press, Oxford, 1987.

[Ventola 87] Ventola, E.
Open Linguistics Series: *The Structure of Social Interaction: A Systemic Approach to the Semiotics of Service Encounters.*
Frances Pinter Publishers, London, 1987.

[Webber 88] Webber B.L.
Discourse Deixis: Reference to Discourse Segments.
In *Proceedings of the 26th ACL*, pages 113-122. Association for Computational Linguistics, Buffalo, NY, June, 1988.

[Winograd 83] Winograd, T.
Language as a Cognitive Process.
Addison-Wesley, Reading, Ma., 1983.

[Winograd & Flores 86]
Winograd, T. and Flores, F.
Language and Being: *Understanding Computers and Cognition - A New Foundation for Design.*
Ablex Publishing Corporation, Norwood, New Jersey, 1986.

[Woolf 84] Woolf, B.P.
Context Dependent Planning in a Machine Tutor.
PhD thesis, University of Massachussets, Amherst, 1984.
UMI #8418906.

Table of Contents