A Collaborative Planning Model of Intentional Structure

Karen E. Lochbaum*
U S WEST Advanced Technologies

An agent's ability to understand an utterance depends upon its ability to relate that utterance to the preceding discourse. The agent must determine whether the utterance begins a new segment of the discourse, completes the current segment, or contributes to it. The intentional structure of the discourse, comprised of discourse segment purposes and their interrelationships, plays a central role in this process (Grosz and Sidner 1986). In this paper, we provide a computational model for recognizing intentional structure and utilizing it in discourse processing. The model is based on the collaborative planning framework of SharedPlans (Grosz and Kraus 1996).

1. Introduction

People engage in dialogues for a reason. Their intentions guide their behavior and their conversational partners' recognition of those intentions aids in the latter's understanding of their utterances (Grice 1969; Sidner 1985; Grosz and Sidner 1986). In this paper, we present a computational model for recognizing the intentional structure of a discourse and utilizing it in discourse processing.

The embedded subdialogues in Figures 1 through 3 illustrate a variety of intentions that a person or computer system must recognize to respond effectively to its conversational partner. These dialogues are drawn from the computational linguistics literature and will be used throughout the paper. We have chosen to use these dialogues, rather than constructing or collecting new ones, in order to elucidate the differences between our theory and previous ones.

The dialogue in Figure 1 contains two **subtask subdialogues**; the dialogue in Figure 2 a **correction subdialogue** (Litman 1985; Litman and Allen 1987); and the dialogue in Figure 3 two **knowledge precondition subdialogues**. The names of the subdialogue types are suggestive of a conversational participant's reason for engaging in them. Although these reasons are diverse, the dialogues all exhibit a common structural regularity; the recognition of this structure is crucial for discourse processing.

Intuitive analyses of the sample dialogues serve to illustrate this point. Before presenting these analyses, however, we first introduce some terminology that will be used throughout the paper. A discourse is composed of **discourse segments** much as a sentence is composed of constituent phrases (Grosz and Sidner 1986). The segmental structure of the sample dialogues is indicated by the bold rule grouping utterances into segments. Whereas the term discourse segment applies to all types of discourse, the term **subdialogue** is reserved for segments that occur within dialogues. All of the examples in this paper are subdialogues. For expository purposes, we will take the initiator of a discourse to be female and the other participant to be male, thus affording the use of the pronouns *she* and *he* in analyzing the sample dialogues. We will also

---

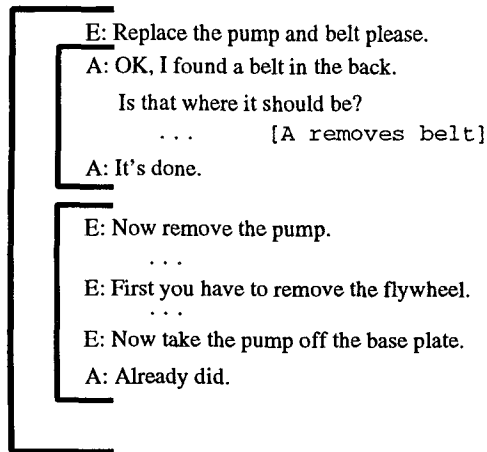* 4001 Discovery Dr., Boulder, CO 80303. E-mail: klochba@uswest.com

E: Replace the pump and belt please.
A: OK, I found a belt in the back.
    Is that where it should be?
            . . .   [A removes belt]
A: It's done.

E: Now remove the pump.
            . . .
E: First you have to remove the flywheel.
            . . .
E: Now take the pump off the base plate.
A: Already did.

Figure 1
Sample subtask subdialogues (Grosz 1974).

(1)   User: Show me the generic concept called "employee".
(2) System: OK.   <system displays network>
(3)   User: I can't fit a new ic below it.
(4)   Can you move it up?
(5) System: Yes.   <system displays network>
(6)   User: OK, now make an individual employee concept
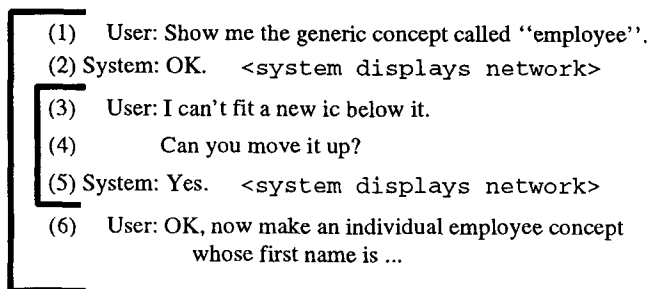   whose first name is ...

Figure 2
A sample correction subdialogue (Sidner 1983; Litman 1985).

use the terms *agent* and *it* in more abstract discussions and will apply them to both people and computer systems, both individually as well as in groups.

   A subtask subdialogue, then, occurs within a task-oriented dialogue and is concerned with a subtask of the overall act underlying the dialogue.[1] An agent initiates a subtask subdialogue to support successful execution of the subtask: communicating about the subtask enables the agent to perform it, as well as to coordinate its actions with its partner's. In the dialogue of Figure 1, the Apprentice (participant "A") initiates the first subdialogue for two reasons: (i) because he believes that removing the belt of the air compressor plays a role in replacing its pump and belt and (ii) because he wants to enlist the Expert's help in removing the belt. Reason (ii) underlies the subdialogue itself, while reason (i) is reflected in the relationship of the subdialogue to the preceding discourse. The Expert must recognize both of these reasons to respond effectively to the Apprentice. For example, suppose that the Expert believes that the Apprentice's belief in (i) is incorrect; that is, she believes that the proposed subtask does not play a role in performing the overall task. The Expert should then communicate that information to the Apprentice (Pollack 1986b). If we were to design a system

---

1 In some sense, all dialogues are task oriented. The task simply varies from the physical (e.g., removing a pump) to the more intellectual (e.g., satisfying a knowledge precondition).

(1) NM: It looks like we need to do some maintenance on node39.
(2) NP: Right.
(3)   How shall we proceed?
(4) NM: Well, first we need to divert the traffic to another node.
(5) NP: Okay.
(6)   Then we can replace node39 with a higher capacity switch.
(7) NM: Right.
(8) NP: Okay good.

(9) NM: Which nodes could we divert the traffic to?
(10) NP: [puts up diagram]
(11)   Node41 looks like it could temporarily handle the extra load.
(12) NM: I agree.
(13)   Why don't you go ahead and divert the traffic to node41
            and then we can do the replacement.
(14) NP: Okay.
(15)   [NP changes network traffic patterns]
(16)   That's done.

    ...

Figure 3
Sample knowledge precondition subdialogues. (Adapted from Lochbaum, Grosz, and Sidner [1990].)

to play the role of the Expert in this scenario and that system were not designed to recognize the relationship of an embedded subdialogue to the previous discourse, then it would not attribute reason (i) to the Apprentice. It thus would not recognize that the Apprentice mistakenly believes that the proposed subtask contributes to the overall task. As a result, the system would fail to recognize that it should inform the Apprentice of his mistaken belief.

Correction subdialogues provide an even more striking example of the importance of recognizing discourse structure. An agent initiates a correction subdialogue when it requires help addressing some problem. For example, the dialogue in Figure 2 is concerned with modifying a KL-ONE network (Brachman and Schmolze 1985). The User produces utterance (3) of the dialogue because she is unable to perform the next act that she intends, namely adding a new concept to the displayed portion of the network. As with the subtask example, the System must recognize this intention to respond appropriately. In particular, it must recognize that the User is not continuing to perform the subtasks involved in modifying the KL-ONE network, but rather is addressing a problem that prevents her from continuing with them. For the System to recognize the User's intention, it must recognize that the User has initiated a new segment of the discourse and also recognize the relationship of that new segment to the preceding discourse.

If the System does not recognize that the User has initiated a new discourse segment with utterance (3), then it will not interpret the User's subsequent utterances in the proper context. For example, it will take the User's utterance in (4) to be a request that the System perform an act in support of further modifying the network, rather than in support of correcting the problem. If the System does not believe that the act of raising up a displayed subnetwork is part of modifying a network, then it will conclude that the User has mistaken beliefs about how to proceed with the mod-

ification. In its response, the System may then choose to correct the User's perceived misconception, rather than to perform the act requested of it.

Even if the System does recognize the initiation of a new discourse segment with utterance (3), i.e., it recognizes that the User is talking about something new, if it does not recognize the relationship of the new segment to the preceding discourse, then it may also respond inappropriately. For example, if the System does not recognize that the act the User cannot perform, i.e., "fitting a new ic below it," is part of modifying the network, then the System may respond without taking that larger context into account. In particular, the System might clear the screen to give the User more room to create the new concept. Such a response would be counterproductive to the User, however, who needs to see the employee concept to create a new instantiation of it.

The last sample dialogue contains a third type of subdialogue, a knowledge precondition subdialogue. Whereas an agent initiates a correction subdialogue when it is physically unable to perform an action, it initiates a knowledge precondition subdialogue when it is "mentally" unable to perform it, i.e., when it lacks the proper knowledge. The dialogue in Figure 3 contains two types of knowledge precondition subdialogues. The first type is concerned with determining a means of performing an act, while the second type is concerned with identifying a parameter of an act. As with the other types of subdialogues discussed above, an agent may respond inappropriately if it does not recognize the relationship of these subdialogues to the preceding discourse. For example, prior to the second subdialogue in Figure 3, agents NM (the Network Manager) and NP (the Network Presenter) have agreed to maintain node39 of a local computer network, in part by diverting traffic from node39 to some other node. To perform the divert traffic act, however, the agents must identify that other node. Agent NM initiates the second subdialogue for this purpose. If agent NP does not recognize the context in which the node is to be identified, i.e., for the purpose of diverting network traffic, then it may respond with a description that will not serve that purpose. For example, it may respond with a description like "the node with the lightest traffic," rather than with a name like "node41."

Thus, although the sample dialogues include a variety of subdialogue types, the type of processing required to participate in the dialogues is the same. In each case, an agent must recognize both the purpose of an embedded subdialogue and the relationship of that purpose to the purposes associated with the preceding discourse. These purposes and their interrelationships form the **intentional structure** of the discourse (Grosz and Sidner 1986). In this paper, we present a computational model for recognizing intentional structure and utilizing it in discourse processing. Our model is based on the collaborative planning framework of SharedPlans (Grosz and Sidner 1990; Lochbaum, Grosz, and Sidner 1990; Grosz and Kraus 1993, 1996).

In the first part of this paper, we describe the SharedPlan model in general. In Section 2, we summarize its concepts and definitions and then in Section 3 discuss the model of discourse processing that it imposes, contrasting it with more traditional approaches to planning and plan recognition. In the second part of the paper, we turn to Grosz and Sidner's theory of discourse structure. We first describe their theory and then show how SharedPlans may be used to model its intentional component. Next, we demonstrate how the process for reasoning with SharedPlans presented in Section 3 may be mapped to the problem of recognizing and reasoning with intentional structure. In this paper, we focus on the theory of using SharedPlans to model intentional structure; however, the theory has also been implemented in a working system (Lochbaum 1994). In the third part of the paper we evaluate our approach against previous ones. We first return to Grosz and Sidner's theory and evaluate our model against the constraints that it imposes. We then compare our approach to discourse

processing with previous plan-based approaches (Litman 1985; Litman and Allen 1987; Lambert and Carberry 1991; Ramshaw 1991) and show that our approach is aimed at recognizing and reasoning with a different type of intention. Whereas our approach is concerned with discourse-level intentions, previous approaches have been concerned with utterance-level intentions.

2. The SharedPlan Definitions

Grosz and Sidner (1990) originally proposed SharedPlans as a more appropriate model of plans for discourse than the single-agent plans based on AI planning formalisms such as STRIPS (Fikes and Nilsson 1971). SharedPlans differ from these other types of plans in providing a model of collaborative, multiagent plans. Collaborative plans better characterize the nature of discourse. As Grosz and Sidner put it (1990, 418):

> Discourses are fundamentally examples of collaborative behavior. The participants in a discourse work together to satisfy various of their individual and joint needs. Thus, to be sufficient to underlie discourse theory, a theory of actions, plans, and plan recognition must deal adequately with collaboration.

Models of single-agent plans are not sufficient for this purpose. As Grosz and Sidner and others (Searle 1990; Bratman 1992; Grosz and Kraus 1996) have shown, collaboration cannot be modeled by simply combining the plans of individual agents.

SharedPlans are also distinguished from other planning formalisms in taking plans to be complex mental attitudes rather than abstract data structures. As Pollack noted (1990, 77):

> There are plans and there are plans. There are the plans that an agent "knows": essentially recipes for performing particular actions or for achieving particular goal states. And there are the plans that an agent adopts and that subsequently guide his action.

Whereas data-structure approaches to planning and plan recognition are focused on the first type of plan, mental phenomenon approaches are focused on the second.[2] To distinguish these two types of "plans," we adopt Pollack's terminology and use the term *recipe* for the first type. Recipes are structures of actions; they represent what agents know when they know a way of doing something. We also follow Pollack in reserving the term *plan* for the collection of mental attitudes that an agent, or set of agents, must hold to act successfully. Thus, while recipes are comprised primarily of actions, plans are comprised of beliefs and intentions that are directed at those actions. We elaborate on this point in Section 3.

For an agent G to have an **individual plan** for an act α, it must satisfy the requirements given below (Grosz and Kraus 1996).[3] We will refer to the act α as the **objective** of the agent's plan.

1. G has a recipe for α

---

2 The terms data-structure view of plans and mental phenomenon view of plans were coined by Pollack (1986b).

3 These requirements, and those to follow for collaborative plans, omit the case present in Grosz and Kraus's (1996) work of one agent contracting an act to another.

2.   For each constituent act β_i of the recipe,

  G intends to perform β_i
  G believes that it can perform β_i
  G has an individual plan for β_i

SharedPlans are more complex than individual plans in several ways. First, the group of agents involved in a SharedPlan must have mutual belief of a recipe for action. Second, they must designate a single agent or subgroup of agents to perform each subact in their recipe. If a single agent is selected, that agent must form an individual plan for the subact; if a subgroup is selected, they must form a SharedPlan. Third, the agents involved in a SharedPlan must have commitments toward their own actions, as well as those of their partners. The requirements for a group of agents GR to have a SharedPlan for α are as follows (Grosz and Kraus 1996):

0. GR is committed to performing α

1. GR has a recipe for α

2.   For each single-agent constituent act β_i of the recipe, there is an agent $G_{\beta_i} \in GR$, such that

  (a) G_{β_i} intends to perform β_i

    G_{β_i} believes that it can perform β_i

    G_{β_i} has an individual plan for β_i

  (b) The group GR mutually believe (2a)

  (c) The group GR is committed to G_{β_i}'s success

3.   For each multiagent constituent act β_i of the recipe, there is a subgroup of agents $GR_{\beta_i} \subseteq GR$ such that

  (a) GR_{β_i} mutually believe that they can perform β_i

    GR_{β_i} has a SharedPlan for β_i

  (b) The group GR mutually believe (3a)

  (c) The group GR is committed to GR_{β_i}'s success

Table 1 summarizes the operators used by Grosz and Kraus (1993, 1996) to formalize the requirements of individual and shared plans. Two of these operators, FIP and PIP, are used to model the plans of individual agents. An agent has a FIP or **full individual plan** when it has established all of the requirements outlined above. When the agent has satisfied only a subset of them, it is said to have a **partial individual plan** or PIP.[4] For multiagent plans, Grosz and Kraus provide two SharedPlan operators: FSP and PSP. A set of agents have a **full SharedPlan** (FSP) when all of the mental attitudes outlined above have been established. Until then, the agents' plan will only be partial (the PSP case). In what follows, we will use the term SharedPlan when the degree of completion of a collaborative plan is not at issue. The definitions of FIP and FSP are given in Figures 4 and 5 respectively.[5]

---

4 This description of a PIP is only a rough, though useful, approximation to Grosz and Kraus's (1993, 1996) formal definition.
5 These definitions are high-level schematics of Grosz and Kraus's (1993, 1996) formal definitions. They serve to highlight those aspects of individual and SharedPlans that are relevant to our work, but omit various formal details.

Table 1
Operators used in Grosz and Kraus's (1993, 1996) definitions.

	Operator	Interpretation
Single-Agent	**FIP**	An agent has a **full individual plan** for an act
	PIP	An agent has a **partial individual plan** for an act
	Int.To	An agent **intends to** perform an act
	Int.Th	An agent **intends that** a proposition hold
	CBA	An agent **can bring about** an act
	BCBA	An agent **believes** that it **can bring about** an act
	BEL	An agent **believes** a proposition
Multi-agent	**FSP**	A set of agents have a **full SharedPlan** for an act
	PSP	A set of agents have a **partial SharedPlan** for an act
	CBAG	A set of agents **can bring about** an act
	MBCBAG	A set of agents **mutually believe** that they **can bring about** an act
	MB	A set of agents **mutually believe** a proposition

$FIP(G, \alpha, T_p, T_\alpha, R_\alpha)$

An agent G has a full individual plan at time T_p to perform act α at time T_α using recipe R_α

1. G has a recipe for α
   $$R_\alpha = \{\beta_i, \rho_j\} \land BEL(G, R_\alpha \in Recipes(\alpha), T_p)$$

2. For each constituent act β_i of the recipe,

   (a) G intends to perform β_i
       $$Int.To(G, \beta_i, T_p, T_{\beta_i})$$
       There is a recipe R_{β_i} for β_i such that

       i. G believes that it can perform β_i according to the recipe
           $$(\exists R_{\beta_i})[BCBA(G, \beta_i, R_{\beta_i}, T_{\beta_i}, T_p, \{\rho_j\}) \land$$
       ii. G has a full individual plan for β_i using the recipe
           $$FIP(G, \beta_i, T_p, T_{\beta_i}, R_{\beta_i})]$$

Figure 4
Full individual plan (FIP) definition.

As indicated in Clause (1) of the definitions in Figures 4 and 5, recipes are modeled in Grosz and Kraus's definitions as sets of constituent acts and constraints. To perform an act α, an agent must perform each constituent act (the β_i in Clause (1)) in α's recipe according to the constraints of that recipe (the ρ_j). Actions themselves may be further decomposed into act-types and parameters. We will represent an action α as a term of the form $\bar{\alpha}(p_1, \ldots, p_n)$ where $\bar{\alpha}$ represents the act-type of the action and the p_i its parameters. Figure 6 provides a graphical representation of a recipe.

The operators Int.To and Int.Th in Grosz and Kraus's definitions are used to represent different types of intentions. Int.To represents an agent's intention *to* perform an action, while Int.Th represents an agent's intention *that* a proposition hold. Int.To's occur in both types of plans (Clause (2a) in Figures 4 and 5), while Int.Th's occur only in SharedPlans (Clauses (0), (2c) and (3c) in Figure 5). Int.Th's are used to represent commitment to the joint activity and also engender the type of helpful behavior required of collaborating agents (Bratman 1992; Grosz and Kraus 1993, 1996).

The operators CBA, BCBA, CBAG, and MBCBAG in Grosz and Kraus's definitions are ability operators; they encode requirements on an agent's ability to perform an

$FSP(GR, \alpha, T_p, T_\alpha, R_\alpha)$

> *A group of agents GR have a full shared plan at time T_p to perform act α at time T_α using recipe R_α*

0. GR is committed to performing α
$$MB(GR, (\forall G_j \in GR)Int.Th(G_j, Do(GR, \alpha, T_\alpha), T_p, T_\alpha), T_p)$$

1. GR has a recipe for α
$$R_\alpha = \{\beta_i, \rho_j\} \wedge MB(GR, R_\alpha \in Recipes(\alpha), T_p)$$

2. For each single-agent constituent act β_i of the recipe, there is an agent $G_{\beta_i} \in GR$, such that

   (a) G_{β_i} intends to perform β_i
$$Int.To(G_{\beta_i}, \beta_i, T_p, T_{\beta_i})$$
   There is a recipe R_{β_i} for β_i such that

       i. G_{β_i} believes that it can perform β_i according to the recipe
$$(\exists R_{\beta_i})[BCBA(G_{\beta_i}, \beta_i, R_{\beta_i}, T_{\beta_i}, T_p, \{\rho_j\}) \wedge$$
       ii. G_{β_i} has a full individual plan for β_i using the recipe
$$FIP(G_{\beta_i}, \beta_i, T_p, T_{\beta_i}, R_{\beta_i})]$$

   (b) The group GR mutually believe (2a)

$$MB(GR, Int.To(G_{\beta_i}, \beta_i, T_p, T_{\beta_i}) \wedge$$
$$(\exists R_{\beta_i})[CBA(G_{\beta_i}, \beta_i, R_{\beta_i}, T_{\beta_i}, \{\rho_j\}) \wedge$$
$$FIP(G_{\beta_i}, \beta_i, T_p, T_{\beta_i}, R_{\beta_i})], T_p)$$

   (c) The group GR is committed to G_{β_i}'s success
$$MB(GR, (\forall G_j \in GR)$$
$$Int.Th(G_j, (\exists R_{\beta_i})CBA(G_{\beta_i}, \beta_i, R_{\beta_i}, T_{\beta_i}, \{\rho_j\}), T_p, T_{\beta_i}), T_p)$$

3. For each multi-agent constituent act β_i of the recipe, there is a subgroup of agents $GR_{\beta_i} \subseteq GR$ such that

   (a) There is a recipe R_{β_i} for β_i such that

       i. GR_{β_i} mutually believe that they can perform β_i according to the recipe
$$(\exists R_{\beta_i})[MBCBAG(GR_{\beta_i}, \beta_i, R_{\beta_i}, T_{\beta_i}, T_p, \{\rho_j\}) \wedge$$
       ii. GR_{β_i} has a full SharedPlan for β_i using the recipe
$$FSP(GR_{\beta_i}, \beta_i, T_p, T_{\beta_i}, R_{\beta_i})]$$

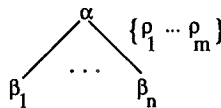   (b) The group GR mutually believe (3a)

$$MB(GR, (\exists R_{\beta_i})[CBAG(GR_{\beta_i}, \beta_i, R_{\beta_i}, T_{\beta_i}, \{\rho_j\}) \wedge$$
$$FSP(GR_{\beta_i}, \beta_i, T_p, T_{\beta_i}, R_{\beta_i})], T_p)$$

   (c) The group GR is committed to GR_{β_i}'s success
$$MB(GR, (\forall G_j \in GR)$$
$$Int.Th(G_j, (\exists R_{\beta_i})CBAG(GR_{\beta_i}, \beta_i, R_{\beta_i}, T_{\beta_i}, \{\rho_j\}), T_p, T_{\beta_i}), T_p)$$

Figure 5
Full SharedPlan (FSP) definition.

A recipe for α is comprised of a set of constituent acts ($\{\beta_1, \ldots, \beta_n\}$) and constraints ($\{\rho_1, \ldots, \rho_m\}$).

Figure 6
A graphical representation of a recipe.

$$has.recipe(G, \alpha, R, T) \Leftrightarrow$$
$$(1) \quad [basic.level(\alpha) \ \wedge$$
$$\qquad BEL(G, basic.level(\alpha), T) \wedge R = R_{Empty}] \ \vee$$
$$(2) \quad [\neg basic.level(\alpha) \ \wedge$$
$$(2a) \quad R = \{\beta_i, \rho_j\} \ \wedge$$
$$(2a1) \qquad \{[\|G\| = 1 \wedge BEL(G, R \in Recipes(\alpha), T)] \ \vee$$
$$(2a2) \qquad [\|G\| > 1 \wedge MB(G, R \in Recipes(\alpha), T)]\}]$$

Figure 7
Definition of *has.recipe*.

action. CBA (read "can bring about") and BCBA ("believes can bring about") are single-agent operators, while CBAG ("can bring about group") and MBCBAG("mutually believe can bring about group") are the corresponding group operators.

An agent's ability to perform an action depends upon its ability to satisfy both the physical and knowledge preconditions of that action (McCarthy and Hayes 1969; Moore 1985; Morgenstern 1987). For example, for an agent to pick up a particular tower of blocks, it must (i) know how to pick up towers in general, (ii) be able to identify the tower in question, and (iii) have satisfied the physical preconditions or constraints associated with picking up towers (e.g., it must have a free hand). In Grosz and Kraus's (1996) definitions, conditions of the form in (iii) depend upon the constraints under which an act β_i is to be performed. These constraints derive from the recipe in which β_i is a constituent and are represented by $\{\rho_j\}$ in the plan definitions in Figures 4 and 5. Conditions of the form in (i) and (ii) above are knowledge preconditions. Knowledge preconditions were not represented in Grosz and Kraus's (1993) original definitions, but were subsequently formalized by the author. We now present definitions of the relations necessary to model these conditions.

2.1 Knowledge Preconditions

2.1.1 Determining Recipes. For an agent G to be able to perform an act α, it must know *how* to perform α; i.e., it must have a recipe for the act. The relation *has.recipe*(G, α, R, T) is used to represent that agent G has a recipe R for an act α at time T. Its formalization is as shown in Figure 7.

Clause (1) of the definition indicates that an agent does not need a recipe to perform a **basic-level** action, i.e., one executable at will (Pollack 1986a).[6] For non-basic-level actions (Clause (2)), the agent of α (either a single agent (2a1) or a group of agents (2a2)) must believe that some set of acts, β_i, and constraints, ρ_j, constitute a recipe for α.

The *has.recipe* relation can be used to represent one of the knowledge precondition requirements of the ability operators, as well as the recipe requirement in Clause (1) of the plan definitions in Figures 4 and 5.

2.1.2 Identifying Parameters. An agent must also be able to identify the parameters of an act α to be able to perform it. For example, if an agent is told, *"Now remove the pump [of the air compressor],"* as in the dialogue of Figure 1, the agent must be able to identify the pump in question. The ability to identify an object is highly context dependent. For example, as Appelt points out (1985, 200), "the description that one

---

6 Basic-level actions are by their nature single-agent actions.

must know to carry out a plan requiring the identification of 'John's residence' may be quite different depending on whether one is going to visit him, or mail him a letter."

The relation $id.params(G, \alpha, T)$ is used to represent that agent G can identify the parameters of act α at time T. If α is of the form $\bar{\alpha}(p_1, \ldots, p_n)$, then $id.params(G, \alpha, T)$ is true if G can identify each of the p_i. To do so, G must have a description of each p_i that is suitable for $\bar{\alpha}$. The relation $id.params$ is thus defined as follows:

$$id.params(G, \bar{\alpha}(p_1, \ldots, p_n), T) \Leftrightarrow (\forall i, 1 \le i \le n) \ has.sat.descr(G, p_i, \mathcal{F}(\bar{\alpha}, p_i), T)$$

The function $\mathcal{F}$ in the above definition is a kind of "oracle" intended to model the context-dependent nature of parameter identification. This function returns a suitable **identification constraint** (Appelt and Kronfeld 1987) for a parameter p_i in the context of an act-type $\bar{\alpha}$. For example, in the case of sending a letter to John's residence, the constraint produced by the oracle function would be that John's residence be described by a postal address.

The relation $has.sat.descr(G, P, C, T)$ holds of an agent G, a parameter description P, an identification constraint C, and a time T, if G has a suitable description, as determined by C, of the object described as P at time T. To formalize this relation, we rely on Kronfeld's (1986, 1990) notion of an **individuating set**. An agent's individuating set for an object is a maximal set of terms such that each term is believed by the agent to denote that object. For example, an agent's individuating set for John's residence might include its postal address as well as an identifying physical description such as "the only yellow house on Cherry Street." To model individuating sets we introduce a function $IS(G, P, T)$; the function returns an agent G's individuating set at time T for the object that G believes can be described as P. This function is based on similar elements of the formal language that Appelt and Kronfeld (1987) introduce as part of their theory of referring. The function returns a set that contains P as well as the other descriptions that G has for the object that it believes P denotes.

The relation $has.sat.descr$ is used to represent that an agent can identify a parameter for some purpose. For that to be the case, the agent must have a description, P', of the parameter such that P' is of the appropriate sort. For example, for an agent to visit John's residence, it is not sufficient for the agent to believe that the description "John's residence" refers to the place where John lives. Rather, the agent needs another description of John's residence, one such as "the only yellow house on Cherry Street," that is appropriate for the purpose of visiting him. To model an agent's ability to identify a parameter for some purpose, we thus require that the agent have an individuating set for the parameter that contains a description P' such that P' satisfies the identification constraint that derives from the purpose. The definition of $has.sat.descr$ is thus as shown in Figure 8.[7] The predicate $suff.for.id(C, P')$ is true if the identification constraint C applies to the parameter description P'. The oracle function $\mathcal{F}(\bar{\alpha}, p_i)$ in $id.params$ is used to produce the appropriate identification constraint on p_i given $\bar{\alpha}$. Identification constraints can derive from syntactic, semantic, discourse, and world knowledge (Appelt and Kronfeld 1987).

In Figures 7 and 8, we have separated the requirements of recipe identification from those of parameter identification. That is, we have defined $has.recipe$ and $id.params$ as independent relations, and do not require an agent to know the parameters of an act to be said to know a recipe for that act. The separation of these two requirements

---

7 A more precise account of what it means to be able to identify an object is beyond the scope of this paper; for further details, see the discussions by Hobbs (1985), Appelt (1985), Kronfeld (1986, 1990), and Morgenstern (1988).

$$has.sat.descr(G, P, C, T) \Leftrightarrow$$
$$\{[|G| = 1 \ \wedge \ (\exists P')BEL(G, [P' \in IS(G, P, T) \ \wedge$$
$$suff.for.id(C, P')], T)] \ \vee$$
$$[|G| > 1 \ \wedge \ (\exists P')MB(G, (\forall G_j \in G)[P' \in IS(G_j, P, T) \ \wedge$$
$$suff.for.id(C, P')], T)]\}$$

Figure 8
Definition of *has.sat.descr*.

derives from the distinction between recipes and plans. Whereas an agent may know many recipes for performing an act, he will have a plan for that act only if he is committed to its performance using a particular recipe. For example, an agent may know that one way to hijack a plane involves smuggling a gun on that plane, without actually intending to hijack a plane at all. Similarly, an agent can know a way of hijacking a plane without actually having a particular plane, or gun, in mind. For this reason, we do not make *id.params* a requirement of *has.recipe*. The separation of these two requirements has particular consequences for our model of discourse processing, as will be discussed in Section 7.

With the addition of the knowledge precondition relations defined above, the definitions of the SharedPlan ability operators (CBA, BCBA, CBAG, and MBCBAG) include three components. The definitions of these operators now state that for an agent to be able to perform an act α, it must (i) have a recipe for α (*has.recipe*), (ii) be able to identify the parameters of α (*id.params*), and (iii) be able to satisfy the constraints of its recipe for α (the $\{\rho_j\}$ in the plan definitions).

3. Reasoning with SharedPlans

In more traditional plan-based approaches to natural language processing (e.g., the work of Cohen and Perrault [1979], Allen and Perrault [1980], Sidner [1985], Carberry [1987], Litman and Allen [1987], Lambert and Carberry [1991]), reasoning about plans is focused on reasoning about actions. In these models, actions are represented using operators derived from STRIPS (Fikes and Nilsson 1971) and NOAH (Sacerdoti 1977). Such operators include: a header, specifying the action and its parameters; a precondition list, specifying the conditions that must be true for the action to be performed;[8] a body, specifying how the action is to be performed; and an effects list, specifying the conditions that will hold after the action is performed. Under these models, reasoning about plans involves reasoning according to rules that derive from the components of the plan operators. For example, Allen (1983) introduces a precondition-action rule stating that if agent G wants to achieve proposition P and P is a precondition of an act ACT, then G may want to perform ACT. This rule is used for plan recognition. The corresponding rule for plan construction states that if agent G wants to execute ACT, then G may want to ensure that precondition P is satisfied. Heuristics, derived from both planning and natural language principles, are used to guide the application of the rules to recognize (or construct) the best possible plan accounting for an agent's observations (or desired effects).

---

8 The preconditions may also be supplemented by a list of applicability conditions, specifying the conditions under which it is reasonable to pursue the action, and a list of constraints specifying restrictions on instantiations of the operator's parameters (Litman and Allen 1987; Carberry 1987).

Problem-solving Plan-P1: {_agent1 and _agent2 build a plan for _agent1 to do _action}

Action:	Build-Plan(_agent1, _agent2, _action)
AppCond:	want(_agent1,_action)
Constr:	plan-for(_plan,_action)
	action-in-plan-for(_Jaction,_action)
Prec:	selected(_agent1,_action,_plan)
	know(_agent2,want(_agent1,_action))
	knowref(_agent1,_prop,prec-of(_prop,_plan))
	knowref(_agent2,_prop,prec-of(_prop,_plan))
	knowref(_agent1,_Jaction,need-do(_agent1,_Jaction,_action))
	knowref(_agent2,_Jaction,need-do(_agent1,_Jaction,_action))
Body:	1. for all actions _Jaction in _plan
	Instantiate-Vars(_agent1,_agent2,_Jaction)
	2. for all actions _Jaction in _plan
	Build-Plan(_agent1,_agent2,_Jaction)
Effects:	have-plan(_agent1,_plan,_action)
Goal:	have-plan(_agent1,_plan,_action)

Figure 9
Lambert and Carberry's (1991, 49) Build-Plan operator.

In these more traditional approaches, the inference rules are typically expressed in terms of the beliefs and goals of the speaker and hearer. Allen's (1983, 126) precondition-action rule, for example, is represented as:

$$\text{SBAW}(P) \supset_i \text{SBAW}(ACT) - \text{if } P \text{ is a precondition of action ACT}$$

where SBAW(P) represents that the inferring agent S believes that agent A wants P. As Pollack (1986b, 1990) has noted, however, these mental attitudes are typically transparent to the reasoning process. The system reasons about the action operators themselves—in this case whether P is a precondition of ACT—and essentially ignores the mental attitudes in the rules; they are simply carried forward from antecedent to consequent. Pollack has dubbed these approaches data-structure approaches because of their focus on the action operators themselves, rather than on the mental attitudes that are required for planning and plan recognition.

In more recent work, the action operators have come to incorporate many more requirements on the agents' mental states. For example, Lambert and Carberry's operator for the action of building a plan is given in Figure 9. The Build-Plan operator is used to represent the process by which two agents build a plan for one of them to perform an action. The preconditions of the operator specify requirements on the agents' mental states, e.g., that the agents know the referents of the subactions that one of them needs to perform to accomplish the overall action. The main difference between this type of approach and the SharedPlan approach discussed in this paper is in the focus of the representation. The representation in Figure 9 specifies requirements on performing an action, some of which are requirements on mental states. The SharedPlan definition in Figure 5 specifies requirements on mental states, some of which refer to actions and their decompositions. One might thus think of the representation in Figure 9 as being "inside-out" from that in Figure 5. Because the focus of the representation in Figure 9 remains on the action and its decomposition, we continue to refer to these types of approaches as data-structure approaches. We reserve the term mental phenomenon approach for those approaches, such as SharedPlans

and Pollack's individual plans (Pollack 1986b, 1990; Grosz and Kraus 1996), that take mental states to be primary. We will return to the implications of representations such as those in Figure 9 in Section 8.2.

The process of reasoning with SharedPlans differs significantly from the process of reasoning with plan operators. Under the SharedPlan approach, agents engaged in discourse are taken to be collaborating on performing some action or on achieving some state of affairs. Each agent brings to their collaboration different beliefs about ways in which to achieve their goal and the actions necessary for doing so. Each agent may have incomplete or incorrect beliefs. In addition, their beliefs about each other's beliefs and capabilities to act may be incorrect. The participants use the discourse to communicate their individual beliefs and to establish mutual ones. Under the SharedPlan approach, the utterances of a discourse are thus understood as contributing information toward establishing the beliefs and intentions that are required for successful collaboration. These beliefs and intentions are summarized by the full SharedPlan definition in Figure 5 and form the basis for the discourse participants' utterances.

Until the agents have established all of the requirements of a full SharedPlan, they will have a partial SharedPlan. The agents' partial SharedPlan evolves over the course of the agents' discourse as they communicate about the actions they will perform, the effects of those actions as they perform them, and the need to revise their plans when things do not proceed as expected. The agents' partial SharedPlan is thus always in a state of flux. At any given point in the agents' discourse, however, it represents the current state of the agents' collaboration. It thus indicates those beliefs and intentions that have been established at that point in the discourse, as well as those that remain to be established over the course of the remaining discourse. The agents' partial SharedPlan thus serves to delineate the information that the agents must consider in interpreting each other's utterances and in determining what they themselves should do or say next. For the agents' utterances to be coherent, they must advance the agents' partial SharedPlan towards completion by helping to establish the "missing" beliefs and intentions.

The concept of **plan augmentation** thus provides the basis for our model of discourse processing. Under this approach, discourse participants' utterances are understood as augmenting the partial SharedPlan that represents the state of their collaboration. Figure 10 provides a high-level specification of this process.[9] It is based on the assumption that agents G_1 and G_2 are collaborating on an act α and models G_1's reasoning in that regard. It thus stipulates how G_1's beliefs about the agents' partial SharedPlan are augmented over the course of the agents' discourse.[10]

It is important to emphasize here that SharedPlans are complex structures that are distributed in nature. The full SharedPlan for a group activity does not, typically, reside in any single agent's mind, nor is there any notion of a group mind in which the SharedPlan resides. Rather, the beliefs and intentions that form a SharedPlan are distributed among the individual minds of the participating agents. Each agent has individual beliefs about its capabilities to act, as well as individual intentions to do so. In addition, agents have commitments towards other agents' abilities to act (represented by intentions that, Int.Th) and mutual beliefs about others' capabilities and

---

9 The details of this process differ significantly from that described in a previous paper (Lochbaum, Grosz, and Sidner 1990).

10 For expository purposes, we will take G_1 to be male and G_2 to be female. We have omitted the time and recipe arguments from the PSP specification for simplicity of exposition and will continue to do so subsequently when they are not at issue.

Assume:

$PSP(\{G_1, G_2\}, \alpha)$,

G_1 is the agent being modeled.

Let *Prop* be the proposition communicated by G_2's utterance $\mathcal{U}$.

1. As a result of the communication, G_1 assumes
   $MB(\{G_1, G_2\}, BEL(G_2, Prop))$.

2. G_1 must then determine the relationship of *Prop* to the current SharedPlan context:

   (a) If G_1 believes that $\mathcal{U}$ or *Prop* indicates the initiation of a subsidiary SharedPlan for an act β, then G_1 will

      i. ascribe $Int.Th(G_2, FSP(\{G_1, G_2\}, \beta))$,
      ii. determine if he is also willing to adopt such an intention.

   (b) If G_1 believes that $\mathcal{U}$ or *Prop* indicates the completion of the current SharedPlan, then G_1 will

      i. ascribe $BEL(G_2, FSP(\{G_1, G_2\}, \alpha))$,
      ii. determine if he also believes the agents' current SharedPlan to be complete.

   (c) Otherwise, G_1 will

      i. ascribe to G_2 a belief that *Prop* is relevant to the agents' current SharedPlan,
      ii. determine if he also believes that to be the case.

3. (a) If Step (2) is successful, then G_1 will signal his agreement (possibly implicitly) and assume mutual belief of the inferred relationship in (2a), (2b), or (2c) as appropriate, updating his view of the agents' PSPs in the process.
   (b) Otherwise, G_1 will query G_2 or communicate his dissent.

Figure 10
The SharedPlan augmentation process.

commitments. The combination of mutual belief and intention is sufficient to model collaboration. No notion of irreducible joint intention (as in Searle's [1990] work), or any other attitude that would refer to a group mind is necessary (Grosz and Kraus 1996).

The processing outlined in Figure 10 assumes that agent G_2 has just communicated an utterance $\mathcal{U}$ with propositional content *Prop*.[11] To make sense of this utterance, G_1 must determine how *Prop* contributes to the agents' PSP for α. In some cases, the linguistic signal $\mathcal{U}$ may aid in this process. As indicated in Figure 10, *Prop* may be interpreted in one of three basic ways. It may indicate the initiation of a subsidiary SharedPlan (Case (a) of Step (2)), signal the completion of the current SharedPlan (Case (b)), or contribute to it (Case (c)). In each of these cases, G_1 first ascribes a particular mental attitude to G_2 on the basis of her utterance (Step (i) in each case) and then reasons about the relevance of that mental attitude to the agents' PSP (Step (ii)). If G_1 is able to make sense of the utterance in this way, he then updates his beliefs about the agents' PSP to reflect their mutual belief of the inferred contribution of *Prop* (Step (3a)). Otherwise, if G_1 does not understand the relevance of G_2's utterance, or

---

11 The recognition of propositional content from surface form has been studied by other researchers (e.g., Allen and Perrault [1980], Litman and Allen [1987], Lambert and Carberry [1991]) and is not discussed in this paper.

disagrees with it, he may simply communicate his dissent to G_2 or query her further (Step (3b)).

In Case (a) of Step (2), *Prop* indicates G_2's intention that the agents collaborate on an act β. G_1 first ascribes this intention to G_2 and then tries to explain it in the context of the agents' PSP for α. If G_1 believes that the performance of β will contribute to the agents' performance of α, and is willing to collaborate with G_2 in this regard, then G_1 will adopt an intention similar to that of G_2's and agree to the collaboration. This process is modeled by Step (2aii). On the basis of his reasoning, G_1 will also update his view of the agents' PSP to reflect that β is an act in the agents' recipe for α for which the agents will form a SharedPlan. This behavior is modeled by Step (3a) of the augmentation process. In this step, agent G_1 updates his view of the agents' partial plan to reflect their mutual belief of the communicated information.

In Case (b) of Step (2), *Prop* indicates G_2's belief that the SharedPlan on which the agents are currently focused is complete. This SharedPlan may represent the agents' primary collaboration or a subsidiary one. In either case, G_1 must determine if he also believes the agents to have established all of the beliefs and intentions required for them to have a full SharedPlan for α. If he does, then he will agree with G_2 and update his view of the agents' PSP for α to reflect that it is complete.
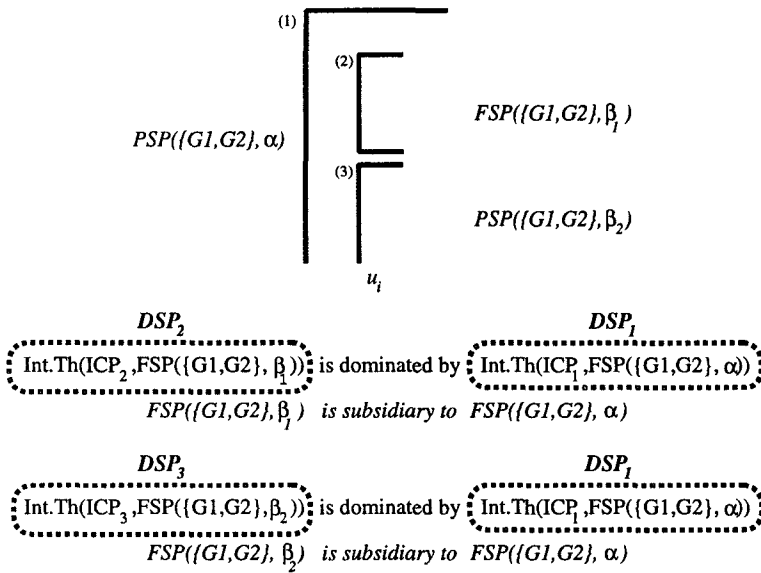
Case (c) of Step (2) is the default case. If G_1 does not believe that *Prop* indicates the initiation or completion of a SharedPlan, then he will take it to contribute to the agents' current SharedPlan in some way. G_1 will first ascribe this belief to G_2 and then reason about the specific way in which *Prop* contributes to the agents' PSP for α. If he is successful in this regard, he will indicate his agreement with G_2 and then update his view of the agents' PSP to reflect this more specific relationship.

Figure 10 provides a high-level specification of the use of SharedPlans in interpretation. In Section 6 we will provide algorithms for further modeling two of the steps in this process, while in Section 10, we will discuss the use of SharedPlans in generation. The main focus of this paper, however, is on modeling the intentional structure of discourse. In the next section, we thus provide a model of that structure. We then show how the model of utterance interpretation presented in Figure 10 can be mapped to the problem of recognizing intentional structure and utilizing it in discourse processing.

4. Grosz and Sidner's Theory of Discourse Structure

According to Grosz and Sidner's (1986) theory, discourse structure is comprised of three interrelated components: a linguistic structure, an intentional structure, and an attentional state. The linguistic structure is a structure that is imposed on the utterances themselves; it consists of discourse segments and embedding relationships among them. The linguistic structure of the sample dialogues in Section 1 is indicated by the bold rule grouping utterances into segments.

The intentional structure of discourse consists of the purposes of the discourse segments and their interrelationships. Discourse segment purposes or DSPs are intentions that lead to the initiation of a discourse segment. DSPs are distinguished from other intentions by the fact that they, like certain utterance-level intentions described by Grice (1969), are intended to be recognized. There are two types of relationships that can hold between DSPs, **dominance** and **satisfaction-precedence**. One DSP dominates another if the second provides part of the satisfaction of the first. That is, the establishment of the state of affairs represented by the second DSP contributes to the establishment of the state of affairs represented by the first. This relationship is reflected by a corresponding embedding relationship in the linguistic structure. One DSP satisfaction-precedes another if the first must be satisfied before the second. This

(1)

(2)

$FSP(\{G1,G2\},\beta_1)$

$PSP(\{G1,G2\},\alpha)$

(3)

$PSP(\{G1,G2\},\beta_2)$

u_i

DSP_2 DSP_1

Int.Th(ICP$_2$,FSP($\{G1,G2\}$, β_1)) is dominated by   Int.Th(ICP$_1$,FSP($\{G1,G2\}$, α))

$FSP(\{G1,G2\},\beta_1)$ is subsidiary to $FSP(\{G1,G2\},\alpha)$

DSP_3 DSP_1

Int.Th(ICP$_3$,FSP($\{G1,G2\}$,β_2)) is dominated by   Int.Th(ICP$_1$,FSP($\{G1,G2\}$,α))

$FSP(\{G1,G2\},\beta_2)$ is subsidiary to $FSP(\{G1,G2\},\alpha)$

Figure 11
Modeling intentional structure.

relationship is reflected by a corresponding sibling relationship in the linguistic structure.

The attentional state component of discourse structure serves as a record of those entities that are salient at any point in the discourse; it is modeled by a stack of focus spaces. With each new discourse segment, a new focus space is pushed onto the stack (possibly after other focus spaces are first popped off), and the objects, properties, and relations that become salient during the segment are entered into it, as is the segment's DSP. One of the primary roles of the focus space stack is to constrain the range of DSPs to which a new DSP can be related; a new DSP can only be dominated by a DSP in some space on the stack. Once a segment's DSP is satisfied, the segment's focus space is popped from the stack.

5. A SharedPlan Model of Intentional Structure

Figure 11 illustrates the role of SharedPlans in modeling intentional structure. As indicated in the figure, we take each segment of a discourse to have an associated SharedPlan. The purpose of the segment is taken to be an intention that (Int.Th) the discourse participants form that plan. This intention is held by the agent who initiates the segment. Following Grosz and Sidner (1986), we will refer to that agent as the ICP for initiating conversational participant; the other participant is the OCP. DSPs are thus represented as intentions of the form $Int.Th(ICP, FSP(\{ICP, OCP\}, \beta))$ in our model.

Relationships between DSPs derive from relationships between the corresponding SharedPlans. For example, a satisfaction-precedence relationship between DSPs corresponds to a temporal dependency between SharedPlans.[12] When one DSP satisfaction-

---

12 Thanks to Christine Nakatani for initially suggesting this correspondence.

precedes another, the SharedPlan used to model the first must be completed before the SharedPlan used to model the second. Dominance relationships between DSPs depend upon subsidiary relationships between the corresponding SharedPlans. In Section 3, we used the term subsidiary SharedPlan to indicate a subordinate relationship between SharedPlans. More generally, one plan is subsidiary to another if the completion of the first plan establishes one of the beliefs or intentions required for the agents to have the second plan. One plan is thus subsidiary to another if the completion of the first plan contributes to the completion of the second.

The utterances of a discourse are understood in terms of their contribution to the SharedPlans associated with the segments of the discourse. Those segments that have been completed at the time of processing an utterance have a full SharedPlan associated with them (e.g., segment (2) in Figure 11), while those that have not have a partial SharedPlan (e.g., segments (1) and (3) in Figure 11).

5.1 Dialogue Analyses

We now return to the dialogues in Section 1 to illustrate the use of SharedPlans in modeling intentional structure. In this section of the paper, we simply describe the intentional structure representations for these examples. In Section 6, we describe the process by which these structures may be recognized and reasoned with.

5.1.1 Example 1: Subtask Subdialogues. The overall purpose of the dialogue in Figure 1 may be represented as:[13]

$DSP_1 = Int.Th(e, FSP(\{e, a\}, replace(pump(ac1)\&belt(ac1), \{a\})))$
  "E intends that the agents collaborate to replace the pump and belt of the air compressor, ac1."

The circumstances surrounding this dialogue are such that only the Apprentice is physically capable of performing actions; the Expert is in another room and can only instruct the Apprentice as to which actions to perform. Both of the agents participate in the act of replacing the pump and belt of the air compressor, though each agent brings different skills to the task. The Expert provides the expertise, while the Apprentice provides the manual dexterity. Thus, the agent specification of the FSP in DSP_1 includes both the Expert and the Apprentice, while only the Apprentice is the agent of the *replace* act itself.

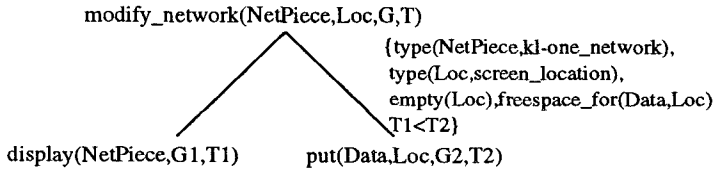The purpose of the first subdialogue in Figure 1 may be represented as:

$DSP_2 = Int.Th(a, FSP(\{e, a\}, remove(belt(ac1), \{a\})))$
  "A intends that the agents collaborate to remove the belt of the air compressor."

while the purpose of the second subdialogue may be represented as

$DSP_3 = Int.Th(e, FSP(\{e, a\}, remove(pump(ac1), \{a\})))$
  "E intends that the agents collaborate to remove the pump of the air compressor."

The SharedPlans used to model DSP_2 and DSP_3 are subsidiary to that used to model DSP_1 by virtue of the subsidiary plan requirement of the SharedPlan definition. As shown in Clauses (2aii) and (3aii) of the definition in Figure 5, an FSP for an act α includes as components full plans for each subact in α's recipe. A plan for one of the subacts β_i thus contributes to the FSP for α, and is therefore subsidiary to it. Because the tasks of removing the air compressor's belt and pump are subtasks of the act of replacing the belt and pump, the SharedPlans to perform those subtasks are subsidiary to the SharedPlan of the main task. DSP_1 thus dominates both DSP_2 and DSP_3.

---

13 We follow the Prolog convention of specifying variables using initial uppercase letters and constants using initial lowercase letters.

modify_network(NetPiece,Loc,G,T)

{type(NetPiece,kl-one_network),
type(Loc,screen_location),
empty(Loc),freespace_for(Data,Loc)
T1<T2}

display(NetPiece,G1,T1)   put(Data,Loc,G2,T2)

Figure 12
A recipe for modifying a network.

5.1.2 Example 2: Correction Subdialogues. The overall purpose of the dialogue in Figure 2 may be represented as:

$DSP_4 = Int.Th(u, FSP(\{u, s\}, modify\_network(NetPiece, Loc, \{u, s\})))$
"U intends that the agents collaborate to modify the piece of network displayed at some screen location."

Figure 12 contains one possible recipe for the act $modify\_network(NetPiece, Loc, G, T)$.[14] The recipe requires that an agent display a piece of a network and then put some new data at some screen location. The constraints of the recipe require that the screen location be empty and that there be enough free space for the data at that location.

The purpose of the subdialogue in Figure 2 may be represented as:[15]

$DSP_5 = Int.Th(u, FSP(\{u, s\}, Achieve(freespace\_for(Data, below(ge1)), \{u, s\})))$
"U intends that the agents collaborate to free up some space below the employee concept."

The SharedPlan used to model DSP_5 is subsidiary to that used to model DSP_4 by virtue of the ability operator $BCBA$. As discussed in Section 2, an agent G's ability to perform an act β depends in part on its ability to satisfy the constraints of its recipe for β. A plan to satisfy one of the constraints thus contributes to the plan for β and is therefore subsidiary to it. Because the condition $freespace\_for(Data, Loc)$ is a constraint in the recipe for $modify\_network(NetPiece, Loc, G, T)$, the SharedPlan in DSP_5 to free up space on the screen is subsidiary to the SharedPlan in DSP_4 to modify the network. DSP_4 thus dominates DSP_5.

5.1.3 Example 3: Knowledge Precondition Subdialogues. The overall purpose of the dialogue in Figure 3 may be represented as:

$DSP_6 = Int.Th(nm, FSP(\{nm, np\}, maintain(node39, \{nm, np\})))$
"NM intends that the agents collaborate to maintain node39 of the local computer network."

The purpose of the first subdialogue in Figure 3 may be represented as:

$DSP_7 = Int.Th(np,$
$\qquad FSP(\{nm, np\},$
$\qquad\qquad Achieve(has.recipe(\{nm, np\}, maintain(node39, \{nm, np\}), R), \{nm, np\})))$
"NP intends that the agents collaborate to obtain a recipe for maintaining node39."

The SharedPlan used to model DSP_7 is subsidiary to the SharedPlan used to model DSP_6 by virtue of the recipe requirement of the SharedPlan definition. As shown in Clause (1) of the definition in Figure 5, for a group of agents to have an FSP for an act α, they must have mutual belief of a recipe for α. The SharedPlan in DSP_7 to obtain

---

14 This recipe derives from the operators used in Sidner's (1985) and Litman's (1985) representations of the acts and constraints underlying the exchange in Figure 2.
15 The function *Achieve* takes propositions to actions (Pollack 1986a).

a recipe for maintaining node39 thus contributes to the SharedPlan in DSP_6 to do the maintenance and is therefore subsidiary to it. As a result, DSP_6 dominates DSP_7.

The second subdialogue in Figure 3 is concerned with identifying a parameter of an act. The purpose of this subdialogue may be represented as:

$DSP_8 = Int.Th(nm,$
$\qquad FSP(\{nm, np\},$
$\qquad\qquad Achieve(has.sat.descr(\{nm, np\}, ToNode, \mathcal{F}(divert\_traffic, ToNode)),$
$\qquad\qquad\qquad \{nm, np\})))$
$\qquad$ *"NM intends that the agents collaborate to obtain a suitable description of the* ToNode
$\qquad$ *parameter of the* divert_traffic *act."*

The SharedPlan used to model DSP_8 is subsidiary to that used to model DSP_6 by virtue of the ability operator *BCBA*. As discussed in Section 2, an agent G's ability to perform an act β depends in part on its ability to identify the parameters of β. At this point in the agents' discourse, NM and NP have agreed that the acts $divert\_traffic(node39, ToNode, G1)$ and $replace\_switch(node39, SwitchType, G2)$ will be part of their recipe for maintaining node39. Because the agents' recipe for maintaining node39 includes the act of diverting network traffic, the SharedPlan in DSP_8 to identify the *ToNode* parameter of the *divert_traffic* act contributes to the SharedPlan in DSP_6 to maintain node39. DSP_6 thus dominates DSP_8.

The SharedPlan in DSP_8 is not subsidiary to the SharedPlan in DSP_7, because, *id.params* is not a requirement of *has.recipe*. As we argued in Section 2.1, knowing a recipe for an act should not require identifying the parameters of the act or the acts in its recipe. However, because an agent must have a recipe in mind before it can be concerned with identifying the parameters of the acts in that recipe, the SharedPlan in DSP_7 must be completed before the SharedPlan in DSP_8.[16] DSP_7 thus satisfaction-precedes DSP_8.

6. Reasoning with Intentional Structure

Intentional structure plays a central role in discourse processing. For each utterance of a discourse, an agent must determine whether the utterance begins a new segment of the discourse, completes the current segment, or contributes to it (Grosz and Sidner 1986). If the utterance begins a new segment of the discourse, the agent must recognize the DSP of that segment, as well as its relationship to the other DSPs underlying the discourse and currently in focus. If the utterance completes the current segment, the agent must come to believe that the DSP of that segment has been satisfied. If the utterance contributes to the current segment, the agent must determine the effect of the utterance on the segment's DSP.

We now show how the SharedPlan reasoning presented in Section 3 may be mapped to the problem of recognizing and reasoning with intentional structure. Step (2) of the augmentation process in Figure 10 is divided into three cases based upon the way in which an utterance affects the SharedPlans underlying a discourse. An utterance may indicate the initiation of a subsidiary SharedPlan (Case (2a)), the completion

---

16 There are several means by which an agent can determine a recipe for an act α. If an agent chooses a recipe for α from some type of manual (e.g., a cookbook), then the agent will have a complete recipe for α before identifying the parameters of α's constituent acts. On the other hand, when being told a recipe for α by another agent, the ignorant agent may interrupt and ask about a parameter of a constituent act before knowing all of the constituent acts. In this case, the agent may have only a partial recipe for α before identifying the parameters of the acts in that partial recipe. Thus, if β_i is an act in α's recipe, a discourse segment concerned with identifying a parameter of β_i could be linguistically embedded within a segment concerned with obtaining a recipe for α. This case poses interesting questions for future research regarding the relationship between the two segments' DSPs.

Assume:
  $PSP(\{G_1, G_2\}, \alpha)$,
    The purpose of the current discourse segment, DS_c, is thus
      $DSP_c = Int.Th(ICP, FSP(\{G_1, G_2\}, \alpha))$
  G_1 is the agent being modeled,
  S is a stack of SharedPlans used to represent G_1's beliefs as to which portion of the intentional structure is currently in focus.

Let *Prop* be the proposition communicated by G_2's utterance $\mathcal{U}$.

  2. G_1 must then determine the relationship of *Prop* to S:

    (a) *Does $\mathcal{U}$ or Prop indicate the initiation of a new discourse segment?*

      If G_1 believes that $\mathcal{U}$ or *Prop* indicates the initiation of a subsidiary SharedPlan for an act β, then

        i. G_1 believes that the DSP of the new segment is
            $Int.Th(G_2, FSP(\{G_1, G_2\}, \beta))$.
        ii. G_1 explains the new segment by determining the relationship of the SharedPlan in (i) to the SharedPlans maintained in S.

    (b) *Does $\mathcal{U}$ or Prop indicate the completion of the current discourse segment?*

      If G_1 believes that $\mathcal{U}$ or *Prop* indicates the satisfaction of DSP_c, then

        i. G_1 believes that G_2 believes DS_c is complete.
        ii. If G_1 believes that the agents' PSP for α is complete, then G_1 will also believe that DSP_c has been satisfied and thus DS_c is complete. DSP_c is thus popped from S.

    (c) *Does Prop contribute to the current discourse segment?*

      Otherwise, G_1 will

        i. ascribe to G_2 a belief that *Prop* contributes to the agents' PSP for α
        ii. determine if he also believes that to be the case.

Figure 13
Step (2) of the augmentation process.

of the current SharedPlan (Case (2b)), or its continuation (Case (2c)). These three cases may be mapped to the problem of determining whether an utterance begins a new segment of the discourse, completes the current segment, or contributes to it. In Figure 13, we have recast Step (2) of the augmentation process to reflect this use.

  The augmentation process in Figure 13 specifies the process by which agent G_1 makes sense of agent G_2's utterances given the current discourse context. We use a stack of SharedPlans S to model this context. The stack corresponds to that portion of the intentional structure that is currently in focus. It thus mirrors the attentional state component of discourse structure and contains PSPs corresponding to discourse segments that have not yet been completed. Because the augmentation process depends most heavily upon the SharedPlans that are used to represent DSPs, it simply makes use of the SharedPlans themselves, rather than the full intentions. The full intentions are easily recoverable from the stack representation.

  Case (2a) in Figure 13 models the recognition of new discourse segments and their purposes. If G_1 believes that G_2's utterance indicates the initiation of a new SharedPlan, then G_1 will take G_2 to be initiating a new discourse segment with her utterance.[17] G_1 first ascribes this intention to G_2 (Step (2ai)) and then tries to explain it given the

---

17 As discussed in Section 7.3, the DSP of the new segment may be only abstractly specified at this point.

current discourse context (Step (2aii)). Whereas at the utterance level, a hearer must explain why a speaker said what he did (Sidner and Israel 1981), at the discourse level, an OCP must explain why an ICP engages in a new discourse segment at a particular juncture in the discourse. The latter explanation depends upon the relationship of the new segment's DSP to the other DSPs underlying the discourse. In Step (2aii) of the augmentation process, G_1 must thus determine whether the new SharedPlan would contribute to the agents' SharedPlan for α or to some other plan on the stack **S**. If the new SharedPlan does not contribute to any of the plans on the stack, then it is taken as an interruption. If it does not contribute to the agents' SharedPlan for α, but to another plan on the stack, one for γ say, then G_1 must also determine whether the plans that are above γ on the stack have been completed.

Case (2b) in Figure 13 models the recognition of a segment's completion. If G_1 believes that G_2's utterance signals the completion of the current segment, then G_1 must reason whether he too believes the segment to be complete. For that to be the case, G_1 must believe that all of the beliefs and intentions required of an FSP have been established over the course of the segment. The completion of a segment may be signaled in either the linguistic structure or the intentional structure. For example, in the linguistic structure, cue phrases such as "but anyway" may indicate the satisfaction of a DSP (as well as a pop of the focus space stack). In the intentional structure, the completion of a segment may be signaled by the initiation of a new SharedPlan, as described above.

Case (2c) models the recognition of an utterance's contribution to the current discourse segment. When a speaker produces an utterance within a segment, a hearer must determine why the speaker said what he did. Step (2c) models the hearer's reasoning by trying to ascribe appropriate beliefs to the speaker. These beliefs are ascribed based on the hearer's beliefs about the state of the agents' SharedPlans and the steps necessary to complete them.

6.1 Modeling the Plan Augmentation Process

Figure 13 contains a high-level specification of the process of reasoning with intentional structure. It provides a framework in which to develop further mechanisms for modeling the various steps of this process. In this section, we present two such mechanisms. The first mechanism presents a method for recognizing the initiation of a new discourse segment (Step (2a) in Figure 13); the second describes an algorithm for reasoning about the contribution of an utterance to the current segment (Step (2c)). These two mechanisms are central to the augmentation process, but are not complete; they each model just one aspect of their respective steps of the process. The complete specification of these steps, as well as that of the augmentation process in general, requires further research, as is discussed in Section 10.

6.1.1 Case (2a): Initiating a New Discourse Segment.
Step (2ai) of the augmentation process involves recognizing agent G_2's intention that G_1 and G_2 form a full SharedPlan for an act β. This intention may be recognized using a conversational default rule, CDR_A, shown in Figure 14.[18] The antecedent of this rule consists of two parts: (1a) G_1 must believe that G_2 communicated her desire for the performance of act β to G_1, and (1b) G_1 must believe that G_2 believes they can together perform β. The second condition precludes the case where G_2 is stating her desire to perform the act herself

---

18 This rule extends Grosz and Sidner's (1990) original conversational default rule, CDR1.

(1a) $BEL(G_1, [communicates(G_2, G_1, Desires(G_2, occurs(\beta)), T) \wedge$

(1b) $BEL(G_2, (\exists R_\beta)CBAG(\{G_1, G_2\}, \beta, R_\beta), T)], T)$ $\overset{default}{\Longrightarrow}$

(2) $BEL(G_1, Int.Th(G_2, FSP(\{G_1, G_2\}, \beta)), T)$

Figure 14
Conversational default rule CDR_A.

or for G_1 to perform the act. If conditions (1a) and (1b) are satisfied, then in the absence of evidence to the contrary, G_1 will believe that G_2 intends that they form a full SharedPlan for β.

As given in Figure 14, CDR_A is used to recognize an agent's intention based upon its desire for the performance of a particular act β. The rule may also be used when an agent expresses its desire for a particular state of affairs P. In this case, the expressions $occurs(\beta)$[19] and β are replaced in Figure 14 by P and $Achieve(P, \{G1, G2\}, T)$ respectively.

6.1.2 Case (2c): Contributing to the Current Discourse Segment. Case (2c) of the augmentation process involves recognizing an utterance's contribution to the current SharedPlan. The SharedPlan definitions place requirements on recipes, abilities, plans, and commitments. A SharedPlan may thus be affected by utterances containing a variety of information. We will focus here, however, on utterances that communicate information about a single action β that can be taken to play a role in the recipe of the agents' plan for α. We thus do not deal with utterances concerning warnings (e.g., "Do not clog or close the stem vent under any circumstances" [Ansari 1995]) or utterances involving multiple actions that are related in particular ways (e.g., "To reset the printer, flip the switch." [Balkanski 1993]).

As with the other cases of Step (2) of the augmentation process, Step (i) of Case (c) involves ascribing a particular belief to agent G_2 regarding the relationship of her utterance to the agents' plans. For the types of utterances we are considering here, this belief is concerned with the relationship of the act β to the objective of the agents' current plan, i.e., α. In particular, G_2's reference to β is understood as indicating belief of a *Contributes* relation between β and α. *Contributes* holds of two actions if the performance of the first action plays a role in the performance of the second action (Lochbaum, Grosz, and Sidner 1990; Lochbaum 1994). It is defined as the transitive closure of the *D(irectly)-Contributes* relation. One act *D-Contributes* to another if the first act is an element of the second act's recipe (Lochbaum, Grosz, and Sidner 1990; Lochbaum 1994).[20]

An agent ascribes belief in a *Contributes* relation irrespective of his own beliefs about this relationship. Once he has ascribed this belief, he then reasons about whether he also believes β to contribute to α and in what way. Step (2cii) of the augmentation process corresponds to this reasoning. To model this step, we introduce an algorithm based on the construction of a dynamic recipe representation called a **recipe graph**

---

19 The predicate $occurs(\beta)$ is true if β was, is, or will be performed at the time associated with β as one of its parameters (Balkanski 1993).

20 The term "contributes" is overloaded in this paper. The use of *Contributes* here refers to a relation between actions. Grosz and Sidner (1986) also describe a contributes relation between DSPs that is the inverse of the dominates relation. In addition, we have been using contributes informally to refer to the inverse of a subsidiary relationship between plans.

A recipe for α is comprised of a set of immediate constituent acts ($\{\beta_1,\ldots,\beta_n\}$) and constraints ($\{\rho_1,\ldots,\rho_m\}$).
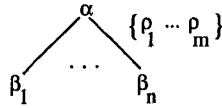
An rgraph for α is comprised of a set of constituent acts and a set of constraints.
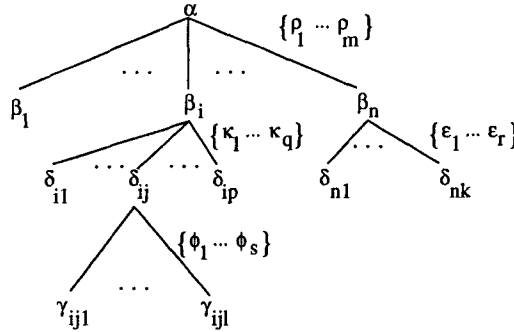
Figure 15
Graphical recipe and rgraph representations.

or **rgraph**.[21] We first describe the rgraph representation and then indicate its role in modeling G_1's reasoning concerning G_2's utterances.

Rgraphs result from composing recipes. Whereas a recipe includes only one level of action decomposition, an rgraph may include multiple levels. On analogy with parsing constructs, one can think of a recipe as being like a grammar rule, while an rgraph is like a (partial) parse tree.[22] Whereas a recipe represents information about the abstract performance of an action, an rgraph represents more specialized information by including instantiations of parameters, agents, and times, as well as multiple levels of decomposition. The graphical representations in Figure 15 contrast the structure of these two constructs.

The construction of an rgraph corresponds to the reasoning that an agent performs in determining whether or not the performance of a particular act β makes sense given the agent's beliefs about recipes and the state of its individual and shared plans. The process of rgraph construction can thus be used to model the process by which agent G_1 explains G_2's presumed belief in a *Contributes* relation. In explaining this belief, however, G_1 must reason about more than just the agents' immediate SharedPlan. In

---

21 This terminology was chosen to parallel Kautz's. He uses the term *explanation graph* or *egraph* for his representation relating event occurrences (Kautz 1990). A comparison of our representation and algorithms with Kautz's can be found elsewhere (Lochbaum 1991, 1994). In short, Kautz's work is based on assumptions that are inappropriate for collaborative discourse. In particular, Kautz assumes a model of **keyhole recognition** (Cohen, Perrault, and Allen 1982) in which one agent is observing another agent without that second agent's knowledge. In such a situation, only actual event occurrences performed by a single agent are reasoned about; Kautz's representation and algorithms include no means for reasoning about hypothetical, partially specified, or multiagent actions. In addition, in keyhole recognition, no assumptions can be made about the interdependence of observed actions. Because the agent is not aware that it is being observed, it does not structure its actions to facilitate the recognition of its motives. A separate egraph must thus be created for each observation.
22 Barrett and Weld (1994) and Vilain (1990) provide further discussion of the use of parsing in planning and plan recognition.

Assume:
  $PSP(\{G_1, G_2\}, \alpha)$,
  G_1 is the agent being modeled,
  R_α is the set of recipes that G_1 knows for α,
  H is an rgraph explaining the acts underlying the discourse up to this point,
  β is the act referred to by G_2.

0. **Initialize Hypothesis:** If β is the first act to be explained in the context of $PSP(\{G_1, G_2\}, \alpha)$, expand **H** by choosing a recipe from R_α and adding it to the rgraph.

1. **Isolate Recipe:** Let **r** be the subtree rooted at α in **H**.

2. **Select Act:** Choose an act β_i in **r** such that β_i can be identified with β and has not previously been used to explain another act. If no such act exists, then fail. Otherwise, let **r**′ be the result of identifying β with β_i in **r**.

3. **Update Hypothesis:** Let **c** = constraints(**r**′) ∪ constraints(**H**). If **c** is satisfiable, replace the subtree **r** in **H** by **r**′, otherwise, fail.

Figure 16
The rgraph construction algorithm.

particular, he must also take into account any other collaborations of the agents, as well as any individual plans of his own. In so doing, G_1 verifies that β is compatible with the rest of the acts the agents have agreed upon, as well as those G_1 intends to perform himself.[23]

The rgraph construction algorithm is given in Figure 16. It is based on the assumption that agents G_1 and G_2 are collaborating on an act α and models G_1's reasoning concerning G_2's reference to an act β. While $PSP(\{G_1, G_2\}, \alpha)$ provides the immediate context for interpreting G_2's utterance, an rgraph **H** models the remaining context established by the agents' dialogue. **H** represents G_1's hypothesis as to how all of the acts underlying the agents' discourse are related. To make sense of G_2's utterance concerning β, G_1 must determine whether β directly contributes to α while being consistent with **H**. Steps (1) and (2) of the algorithm model the immediate explanation of β, while Step (3) ensures that this explanation is consistent with the rest of the rgraph.

The algorithm in Figure 16 is nondeterministic. Step (0) involves *choosing* a recipe from G_1's recipe library, while Step (2) involves *choosing* an act from that recipe. The failures in Steps (2) and (3) do not imply failure of the entire algorithm, but rather failure of a single nondeterministic execution.

In Step (0) of the algorithm, G_1's hypothesis rgraph is initialized to some recipe that he knows for α. As will be discussed in Section 6.2.3, this recipe may involve physical actions, such as those involved in lifting a piano, as well as information-gathering actions, such as those involved in satisfying a knowledge precondition. At the start of the agents' collaboration, G_1 may or may not have any beliefs as to how the agents will perform α. If he believes that the agents will use a particular recipe, the hypothesis rgraph is initialized to that recipe. Otherwise, a recipe is selected arbitrarily from G_1's recipe library. The initial hypothesis will be refined, and possibly replaced, on the basis of G_2's utterances.

In Step (1) of the algorithm, the recipe for α is first isolated from the remainder of the rgraph. This recipe, **r**, represents G_1's current beliefs as to how the agents are going to perform α. Step (2) of the algorithm involves identifying β with a particular

---

23 On the basis of this reasoning, G_1 thus attributes belief in more than just a *Contributes* relation to G_2. In particular, G_1 assumes that G_2 also believes that β is compatible with the other acts the agents have agreed upon.

act β_i in r resulting in a new recipe r'. If an appropriate β_i can be found, it provides an explanation for G_2's reference to the act β. If an appropriate β_i cannot be found, then r cannot be the recipe that G_2 has in mind for performing α. The algorithm thus fails in this case and backtracks to select a different recipe for α. The new recipe must account for β as well as all of the other acts previously accounted for by r.

Step (3) of the algorithm ensures that the recipe and act chosen to account for α and β are compatible with the other acts the agents have already discussed in support of α or the objectives of their other plans. This is done by adding the constraints of the recipe r' to the constraints of the rgraph H and checking that the resulting set is satisfiable. For G_1 to agree to the performance of β, the recipe r' must be both internally and externally consistent. That is, the constraints of the recipe must be consistent themselves, as well as being consistent with the constraints of the recipes that G_1 believes the agents will use to accomplish their other objectives.[24]

The rgraph construction algorithm fails to produce an explanation for an act β in the context of a PSP for α if the algorithm fails for all of the nondeterministic possibilities. This failure corresponds to a discrepancy between agent G_1's beliefs and those G_1 has attributed to agent G_2. The failure thus indicates that further communication and replanning are necessary.

6.2 Dialogue Analyses

To further elucidate the augmentation process, we now return to the dialogues given in Section 1 and show that the processes presented in this section capture the properties highlighted by the informal analyses given in the Introduction. We present each analysis from the perspective of one of the two discourse participants. Each analysis thus indicates the type of reasoning that is required for a system to assume the role of that participant in the dialogue.

6.2.1 Example 1: Subtask Subdialogues.
The dialogue in Figure 17 (repeated from Figure 1) contains two subtask subdialogues. In Section 1 we noted that an OCP must recognize the purpose underlying each subdialogue, as well as the relationship of each purpose to the preceding discourse, in order to respond appropriately to the ICP. The OCP's recognition of DSPs and their interrelationships is modeled by Case (2a) of the augmentation process in Figure 13. We illustrate its use by modeling the Apprentice's reasoning concerning the Expert's first utterance in the second segment in Figure 17, i.e.,

(2a)   E: Now remove the pump.

At this point in the agents' discourse, the stack S consists only of a PSP to replace the air compressor's pump and belt. This PSP corresponds to the overall discourse in Figure 17. The SharedPlan corresponding to the first embedded segment has been completed at this point in the discourse and is thus no longer in focus.

---

24 Another distinction between our work and Kautz's (1990) relates to Step (3) of the algorithm in Figure 16 and the use of constraints. Whereas rgraphs include an explicit representation of constraints, Kautz's egraphs do not. Constraints are used to guide egraph construction, but are not part of the representation itself. As a result, Kautz's algorithms can only check for constraint satisfaction locally. In our algorithm, that would correspond to checking the satisfiability of a recipe's constraints before adding it to an rgraph, but not afterwards. By checking the satisfiability of the constraint set that results from combining the recipe's constraints with the rgraph's constraints, the rgraph construction algorithm is able to detect unsatisfiability earlier than an algorithm that checks constraints only locally.

E: Replace the pump and belt please.

A: OK, I found a belt in the back.

   Is that where it should be?

      . . .   [A removes belt]

A: It's done.

E: Now remove the pump.

   . . .

E: First you have to remove the flywheel.

   . . .

E: Now take the pump off the base plate.

A: Already did.
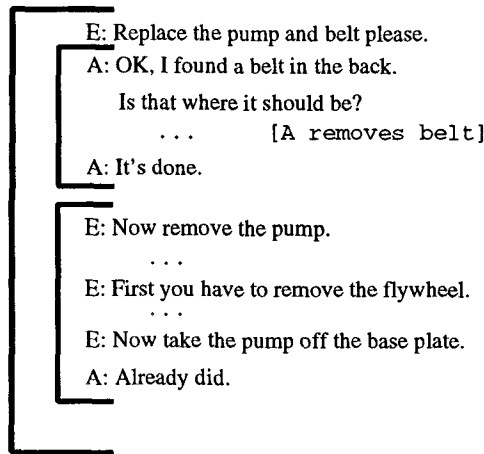
Figure 17
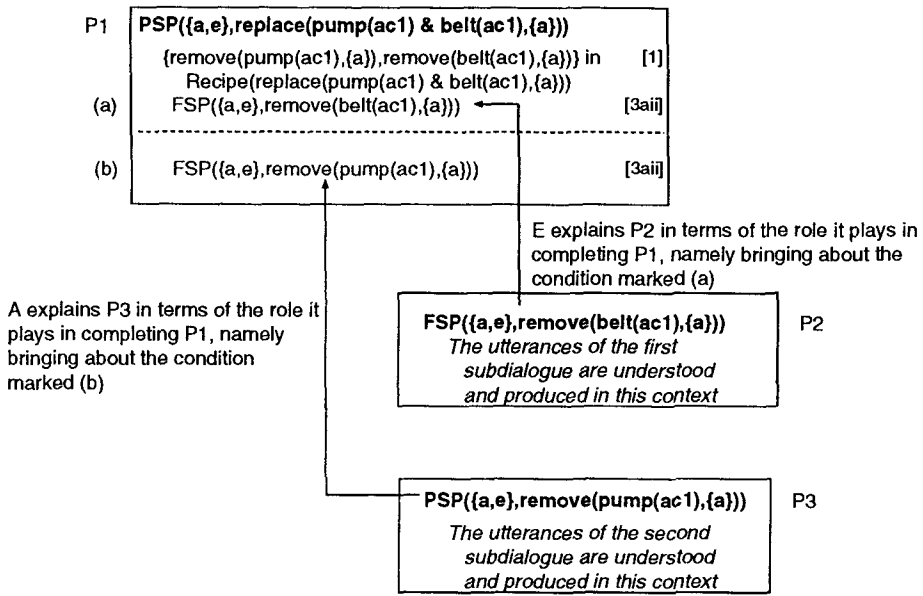Sample subtask subdialogues (Grosz 1974).

In utterance (2a), the Expert expresses her desire that the action $remove(pump(ac1), \{a\})$ be performed, where $ac1$ represents the air compressor the agents are working on. The Apprentice's reasoning concerning this utterance may be modeled using CDR_A. Condition (1a) of CDR_A is satisfied by the communication of this utterance to the Apprentice. Condition (1b) is satisfied by the context surrounding the agents' collaboration. Because the Expert is in another room and can only instruct the Apprentice as to which actions to perform, the Expert's utterance cannot be expressing her intention to perform the desired action herself. In addition, because the Apprentice and Expert are both aware that the Apprentice does not have the necessary expertise to perform the action himself, the Apprentice can assume that the Expert must believe the agents can perform the act together, thus satisfying Condition (1b) and sanctioning the default conclusion, $Bel(a, Int.Th(e, FSP(\{a,e\}, remove(pump(ac1), \{a\}))))$. Thus, on the basis of the Expert's utterance and her presumed beliefs concerning the agents' capabilities to act, the Apprentice may reason that the Expert is initiating a new discourse segment with this utterance. The purpose of this segment is recognized as:

$DSP_3 = Int.Th(e, FSP(\{a,e\}, remove(pump(ac1), \{a\})))$.

Once the Apprentice recognizes the DSP of this new discourse segment, he must determine its relationship to the other DSPs underlying the discourse. Subsidiary relationships between plans provide the basis for modeling the Apprentice's reasoning. In particular, if the Apprentice believes that a plan for removing the pump would further some other plan of the agents', then he will believe that DSP_3 is dominated by the DSP involving that other plan.

As discussed in Section 5.1.1, the subsidiary relation in question in this example derives from the constituent plan requirement of the SharedPlan definition. The Apprentice will succeed in recognizing the relationship of the second subdialogue to the remainder of the discourse, if he believes that removing the pump of the air compressor could be an act in the agents' recipe for replacing its pump and belt. If the Apprentice does not have any beliefs about the relationship between these two acts, he may choose to assume the necessary *D-Contributes* relation on the basis of the Expert's utterance and the current discourse context, or he may choose to query the Expert further.

The rgraph construction algorithm may be used to model the Apprentice's reasoning. In particular, Steps (1) and (2) of the algorithm in Figure 16 model the reasoning

P1 | **PSP({a,e},replace(pump(ac1) & belt(ac1),{a}))**

{remove(pump(ac1),{a}),remove(belt(ac1),{a})} in   [1]
Recipe(replace(pump(ac1) & belt(ac1),{a}))
(a)   FSP({a,e},remove(belt(ac1),{a}))   [3aii]

(b)   FSP({a,e},remove(pump(ac1),{a}))   [3aii]

E explains P2 in terms of the role it plays in
completing P1, namely bringing about the
condition marked (a)

A explains P3 in terms of the role it
plays in completing P1, namely
bringing about the condition
marked (b)

FSP({a,e},remove(belt(ac1),{a}))   P2
*The utterances of the first
subdialogue are understood
and produced in this context*

PSP({a,e},remove(pump(ac1),{a}))   P3
*The utterances of the second
subdialogue are understood
and produced in this context*

Figure 18
Analysis of the dialogue in Figure 17.

necessary for determining that a *D-Contributes* relation holds between two actions. If the OCP is able to infer such a *D-Contributes* relation, he will thus succeed in determining the subsidiary relationship necessary for explaining a subtask subdialogue. If the OCP is unable to infer such a relationship, then the algorithm will fail. This failure indicates that the OCP may need to query the ICP further about the appropriateness of her utterance. For example, as we noted in Section 1, if the OCP has reason to believe that the proposed subtask will not in fact play a role in the agents' overall task, then the OCP should communicate that information to the ICP. In addition, if the OCP has reason to believe that the performance of the subtask will conflict with the agents' other plans and intentions, then the OCP should communicate that information as well. The latter reasoning is modeled by Step (3) of the rgraph construction algorithm. Step (3) ensures that the subtask is consistent with the objectives of the agents' other plans.

Figure 18 contains a graphical representation of the SharedPlans underlying the discourse in Figure 17. It is a snapshot representing the Apprentice's view of the agents' plans just after he explains the initiation of segment (3). Each box in the figure corresponds to a discourse segment and contains the SharedPlan used to model the segment's purpose. The plan used to model DSP_3 is marked P3 in this figure, while the plans used to model DSP_1 and DSP_2 are labeled P1 and P2, respectively. We will continue to follow the convention of co-indexing DSPs with the SharedPlans used to model them in the remainder of this paper.

The information represented within each SharedPlan in Figure 18 is separated into two parts. Those beliefs and intentions that have been established at the time of the snapshot are shown above the dotted line, while those that remain to be established, but are used in determining subsidiary relationships, are shown below the line. Because the last utterance of segment (2) signals the end of the agents' SharedPlan for removing the belt, the FSP for that act occurs above the dotted line. The agents' plan for removing the belt is complete and thus no longer in focus at the start of segment (3). We have included it in the figure for illustrative purposes. The index in square brackets

(1)   User: Show me the generic concept called "employee".

(2) System: OK.   <system displays network>

(3)   User: I can't fit a new ic below it.

(4)   Can you move it up?

(5) System: Yes.   <system displays network>

(6)   User: OK, now make an individual employee concept
            whose first name is ...

Figure 19
A sample correction subdialogue (Sidner 1983; Litman 1985).

to the right of each constituent indicates the clause of the FSP definition from which the constituent arose.

Subsidiary relationships between plans are represented by arrows in the figure and are explained by the text that adjoins them. Plans P2 and P3 are thus subsidiary to plan P1 because of the constituent plan requirement (Clause (3aii)) of the FSP definition. These subsidiary relationships indicate that DSP_2 and DSP_3 are both dominated by DSP_1.

6.2.2 Example 2: Correction Subdialogues. The dialogue in Figure 19 (repeated from Figure 2) contains an embedded correction subdialogue. As discussed in Section 5.1.2, we take the purpose underlying the entire dialogue to be modeled using a SharedPlan to modify a KL-ONE network,

(P4) $PSP(\{u,s\}, modify\_network(NetPiece, Data, Loc, \{u,s\}))$.

We will assume the role of the System in analyzing this example.

The System may have many recipes for modifying a network. One may involve deleting a concept from the network, one may involve changing the data in part of the network, and one may involve adding new data to the network. These three possibilities are depicted in Figure 20. At the beginning of the dialogue, the System may have no prior beliefs as to which of these recipes, if any, he and the User will follow to modify the network. The rgraph construction algorithm is used to model the System's reasoning and, as indicated in Step (0), will select one of these recipes nondeterministically. If the chosen recipe fails to account for the User's utterances, then it cannot be the recipe that the User has in mind for modifying the network. The algorithm will then backtrack at that point to select a different recipe for modifying the network. For illustrative purposes, we will assume that the System initially believes that he and the user are following the first recipe in Figure 20; this recipe involves deleting data from the network. The rgraph that results after the System has explained utterance (1) is shown in Figure 21.

The User's utterance in (3) indicates that she has encountered a problem with the normal execution of the subtasks involved in modifying a network. The System's reasoning regarding this utterance may be modeled using CDR_A. On the basis of the User's utterance and her presumed beliefs concerning the agents' capabilities regarding freeing up space on the screen, the System may reason that the User is initiating a new discourse segment with this utterance. The purpose of this segment is recognized as:

$$DSP_5 = Int.Th(u, FSP(\{u,s\}, Achieve(freespace\_for(Data, below(ge1)), \{u,s\})))$$

modify_network(NetPiece,Loc,G,T)

{type(NetPiece,kl-one_network),
type(Loc,screen_location),
T1<T2}

display(NetPiece,G1,T1)   delete(Data,Loc,G2,T2)

modify_network(NetPiece,Loc,G,T)

{type(NetPiece,kl-one_network),
type(Loc,screen_location),
T1<T2}

display(NetPiece,G1,T1)   change(Data,Loc,G2,T2)

modify_network(NetPiece,Loc,G,T)

{type(NetPiece,kl-one_network),
type(Loc,screen_location),
empty(Loc),freespace_for(Data,Loc)
T1<T2}

display(NetPiece,G1,T1)   put(Data,Loc,G2,T2)

Figure 20
Recipes for modifying a network.

modify_network(ge1,Loc,{u,s})

{type(ge1,kl-one_network),
type(Loc,screen_location)}

display(ge1,{s})   delete(Data,Loc,{u})
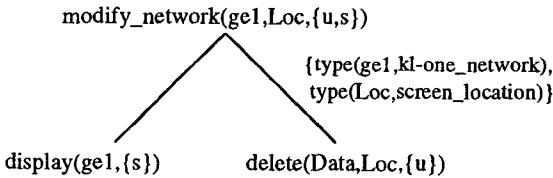
Figure 21
Initial rgraph explaining utterances (1)–(2) of the dialogue in Figure 19.

where $ge1$ represents "the generic concept called 'employee'." To explain the User's initiation of the subdialogue, the System must determine how the SharedPlan in DSP$_5$ will further the agents' plan in (P4).

The System's current beliefs as to how the agents will modify the network, as represented by the rgraph in Figure 21, do not provide an explanation for the User's utterance in (3). The System's recipe does not include any type of "fit" act. The rgraph construction algorithm thus fails at this point and backtracks to nondeterministically select a different recipe for modifying the network. Suppose that this time the third recipe in Figure 20 is selected; this is the recipe that includes adding data to the network. The rgraph that results from using this recipe to explain the User's first utterance is shown in Figure 22. This new rgraph also provide an explanation for the User's utterance in (3); the "fit" act referred to by the User corresponds to the "put" act in the rgraph. In addition, the constraints of the recipe, along with the requirements of the ability operators, provide the explanation for the new discourse segment.

As discussed in Section 5.1.2, an agent G's ability to perform an act β depends in part on its ability to satisfy the constraints of the recipe in which β is a constituent. Thus, to perform the act $put(Data, below(ge1), \{u\})$, the User must be able to satisfy the constraints $empty(below(ge1))$ and $freespace\_for(Data, below(ge1))$. The need to satisfy the latter constraint provides the System with an explanation for DSP$_5$. In particular,
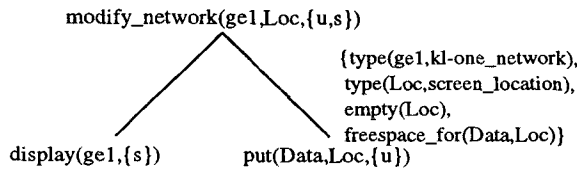
553

modify_network(ge1,Loc,{u,s})

{type(ge1,kl-one_network),
type(Loc,screen_location),
empty(Loc),
freespace_for(Data,Loc)}

display(ge1,{s})   put(Data,Loc,{u})

Figure 22
Second rgraph explaining utterances (1)–(2) of the dialogue in Figure 19.

P4 | PSP({u,s},modify_network(ge1,below(ge1),{u,s}))
{{display(ge1,{s}), put(Data,below(ge1),{u})},   [1]
{empty(below(ge1)), freespace_for(Data,below(ge1))}} in
Recipe(modify_network(ge1,below(ge1),{u,s}))

(a) | BCBA(u,put(Data,below(ge1),{u}),R,   [2ai]
{empty(below(ge1)),freespace_for(Data,below(ge1))})

U engages S in P5 because   S explains P5 in terms of the role it plays in completing P4,
she needs to satisfy (a)   namely bringing about the condition marked (a)

PSP({u,s},Achieve(freespace_for(Data,below(ge1)),{u,s})) | P5
move(ge1,up,{s}) in   [1]
Recipe(Achieve(freespace_for(Data,below(ge1)),{u,s}))

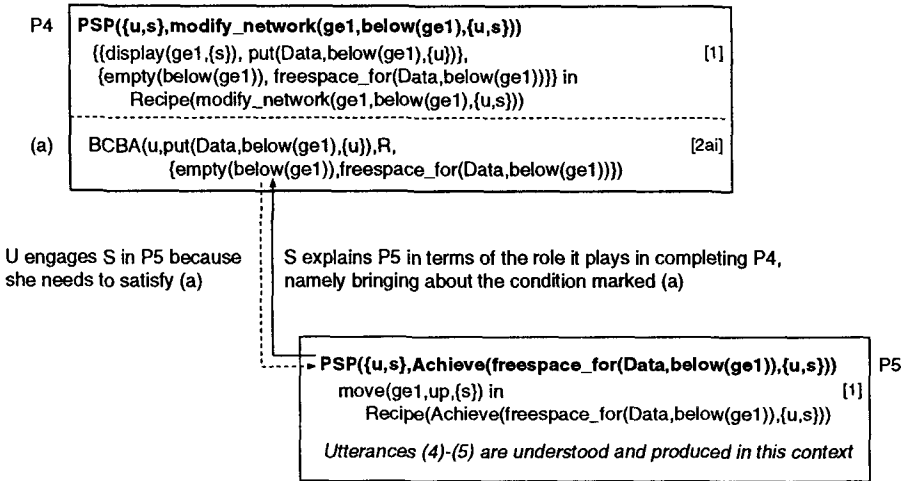*Utterances (4)-(5) are understood and produced in this context*

Figure 23
Analysis of the dialogue in Figure 19.

the System can reason that the User initiated the new discourse segment in order to satisfy one of the ability requirements of the agents' SharedPlan to modify the network. The SharedPlan in DSP$_5$ is thus subsidiary to that in (P4) by virtue of the BCBA requirements of the latter plan. Figure 23 summarizes our analysis of the dialogue. Whereas subtask subdialogues are explained in terms of constituent plan requirements of SharedPlans (Clause (3aii)), correction subdialogues are explained in terms of ability requirements (Clause (2ai)).

Once the System recognizes, and explains, the initiation of the new segment, it will interpret the User's subsequent utterances in the context of its DSP, rather than the previous one. It will thus understand utterance (4) to contribute to freeing up space on the screen, rather than to modifying the network. This reasoning is modeled by Case (3a) of the augmentation process as follows: First, on the basis of its explanation of DSP$_5$, the System will take the agents to have a PSP for the act *Achieve(freespace_for(Data, below(ge1)), {u, s})*. This plan is marked (P5) in Figure 23 and is pushed onto the stack **S** above the plan in (P4). As a result, the System will now take the agents to be focused on the plan in (P5), rather than that in (P4), and thus will interpret the User's subsequent utterances in terms of the information they contribute towards completing the plan in (P5), rather than that in (P4).

The User's utterance in (4) makes reference to an act *move(ge1, up, {s})*. Using the rgraph construction algorithm, this act is understood to directly contribute to the objective of the plan in (P5), i.e., *Achieve(freespace_for(Data, below(ge1)), {u, s})*. The resulting rgraph is shown in Figure 24. This rgraph provides an explanation for utterance (4) in the context of all of the acts involved in the agents' plans.
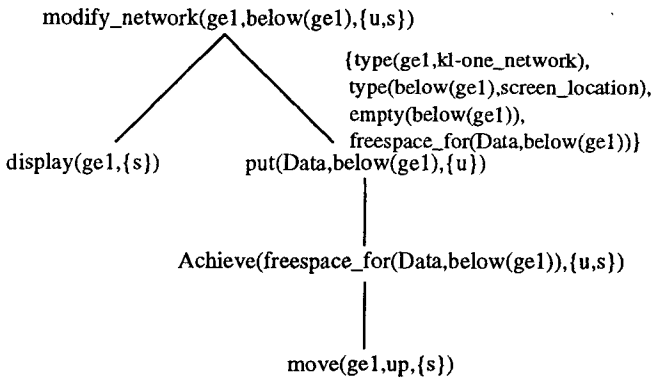
modify_network(ge1,below(ge1),{u,s})

{type(ge1,kl-one_network),
type(below(ge1),screen_location),
empty(below(ge1)),
freespace_for(Data,below(ge1))}

display(ge1,{s})   put(Data,below(ge1),{u})

Achieve(freespace_for(Data,below(ge1)),{u,s})

move(ge1,up,{s})

Figure 24
Rgraph explaining utterances (1)–(4) of the dialogue in Figure 19.

As noted in Section 1, the System's response to the User's request in (4) should take the context of the agents' entire discourse into account and not simply the context of freeing up space on the screen. In particular, the System should not clear the currently displayed network from the screen to help the User perform the task of putting up some new data, but rather should leave the displayed network visible. The discourse context modeled by the SharedPlans in (P4) and (P5), as well as the rgraph in Figure 24, enables the System to respond correctly. In particular, by examining the plans currently in focus and determining what needs to be done to complete them, the System can reason that it should perform an act in support of $Achieve(freespace\_for(Data, below(ge1)), \{u, s\})$. The System will most likely select the requested act of moving $ge1$ up, but if it decides to modify that act in some way or to select a different act, the new act must be compatible with the other acts the agents have agreed upon. By inserting the new act into the rgraph and determining that the resulting rgraph constraints will not be violated by this addition, the System can ensure that its response is in accord with the larger discourse context.

6.2.3 Example 3: Knowledge Precondition Subdialogues. The dialogue in Figure 25 (repeated from Figure 3) contains two embedded knowledge precondition subdialogues. We will assume the role of the Network Presenter, NP, in analyzing this example.
    The overall purpose of the dialogue may be represented as:

$$DSP_6 = Int.Th(nm, FSP(\{nm, np\}, maintain(node39, \{nm, np\})))$$

and can be recognized on the basis of NM's utterance in (1) and CDR_A. The purpose of the first subdialogue in Figure 25 can be represented as:

$$DSP_7 = Int.Th(np,$$
$$FSP(\{nm, np\},$$
$$Achieve(has.recipe(\{nm, np\}, maintain(node39, \{nm, np\}), R), \{nm, np\}))).$$

This first subdialogue is initiated by agent NP, the agent whose reasoning we are modeling. We must thus account for NP's generation of an utterance in this example, rather than his interpretation of another agent's utterance. As will be discussed in Section 10, the use of SharedPlans in generation is an area for future research; however, the basic principles used in interpretation apply here as well. The current state of the agents' plans provides the basis for an agent's communication.
    DSP_7 represents NP's intention that the agents determine a means of diverting network traffic. As discussed in Section 5.1.3, for a group of agents G to have a col-

(1) NM: It looks like we need to do some maintenance on node39.

(2) NP: Right.

(3)   How shall we proceed?

(4) NM: Well, first we need to divert the traffic to another node.

(5) NP: Okay.

(6)   Then we can replace node39 with a higher capacity switch.

(7) NM: Right.

(8) NP: Okay good.

(9) NM: Which nodes could we divert the traffic to?

(10) NP: [puts up diagram]

(11)   Node41 looks like it could temporarily handle the extra load.

(12) NM: I agree.

(13)   Why don't you go ahead and divert the traffic to node41
and then we can do the replacement.

(14) NP: Okay.

(15)   [NP changes network traffic patterns]

(16)   That's done.

...

Figure 25
Sample knowledge precondition subdialogues. (Adapted from Lochbaum, Grosz, and Sidner [1990].)

laborative plan for an act α, the group must have mutual belief of a recipe for α. It is this requirement that leads NP to initiate the first subdialogue; deciding upon a means of performing the objective of the agents' collaboration is a necessary first step to furthering that collaboration. The plan in DSP_7 to agree on a recipe for maintaining node39 thus contributes to the plan in DSP_6 to do the maintenance, and is therefore subsidiary to it. Figure 26 provides a graphical representation of this relationship.

Once NM agrees to the subsidiary collaboration, either explicitly or implicitly as in utterance (4), NP will assume that the agents have a partial SharedPlan to obtain the recipe:

(P7) $PSP(\{nm, np\},$
          $Achieve(has.recipe(\{nm, np\}, maintain(node39, \{nm, np\}), R), \{nm, np\}))$

NP will thus produce his next utterances in the context of the SharedPlan in (P7), rather than that in DSP_6 and will assume that NM will do the same.

To make sense of NM's utterance in (4), NP must provide an explanation for it in the context of the agents' SharedPlan in (P7). The rgraph construction algorithm is used in modeling NP's reasoning. Whereas in the case of a subtask subdialogue, the algorithm makes uses of recipes for performing a subtask, in the case of a knowledge precondition subdialogue, it makes use of recipes for satisfying a knowledge precondition. Figure 27 contains two recipes an agent might know to obtain a recipe for an act α. The first is a single-agent recipe that involves looking up a procedure for α in a manual. The second recipe is a multiagent recipe that involves the agents communicating to come to agreement about the acts and constraints that will comprise their recipe for α.

We use these recipes to model NM's reasoning concerning utterance (4) as follows: In Step (0) of the rgraph construction algorithm, a recipe for the act $Achieve(has.recipe(\{nm, np\}, maintain(node39, \{nm, np\}), R))$ is first selected from NP's recipe library. For illustrative purposes, we will assume that the second recipe in Figure 27 is selected.
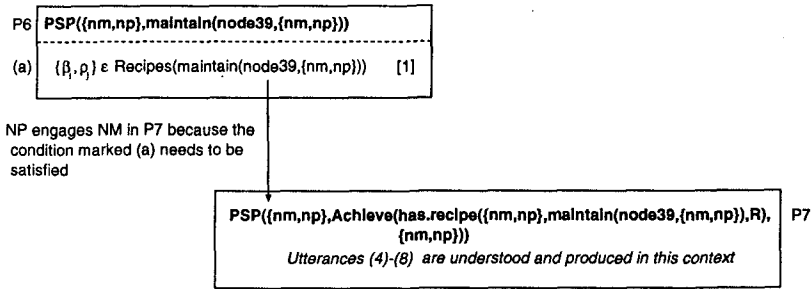
```
P6 │ PSP({nm,np},maintain(node39,{nm,np}))
   │ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─ ─
(a)│  {β,ρ} ε Recipes(maintain(node39,{nm,np}))      [1]
```

NP engages NM in P7 because the
condition marked (a) needs to be
satisfied

```
PSP({nm,np},Achieve(has.recipe({nm,np},maintain(node39,{nm,np}),R),   P7
{nm,np}))
Utterances (4)-(8) are understood and produced in this context
```

Figure 26
Analysis of the first subdialogue in Figure 25.

$$Achieve(has.recipe(G, \alpha, R, T), G, T)$$
$$\{Bel(G, R \; \varepsilon \; Recipes(\alpha), T)\}$$
$$look\_up(G, R, Manual, T)$$

$$Achieve(has.recipe(\{G1, G2\}, \alpha, R, T), \{G1, G2\}, T)$$
$$\{MB(\{G1, G2\}, R \varepsilon \; Recipes(\alpha), T),$$
$$MB(\{G1, G2\} \; Exists \; R[\{\beta_i, \rho_j\} \subseteq R \varepsilon \; Recipes(\alpha)], T)\}$$
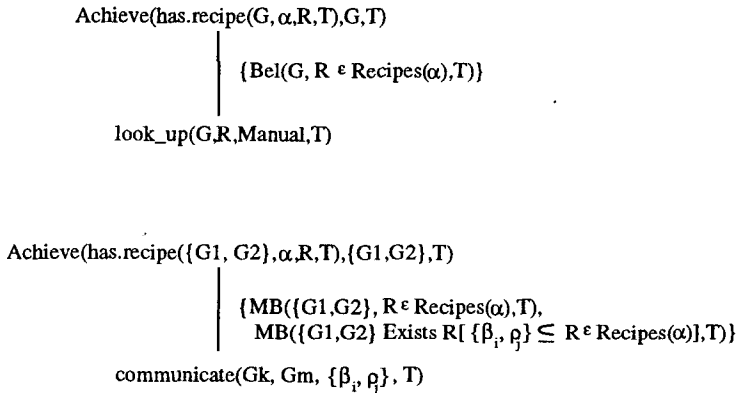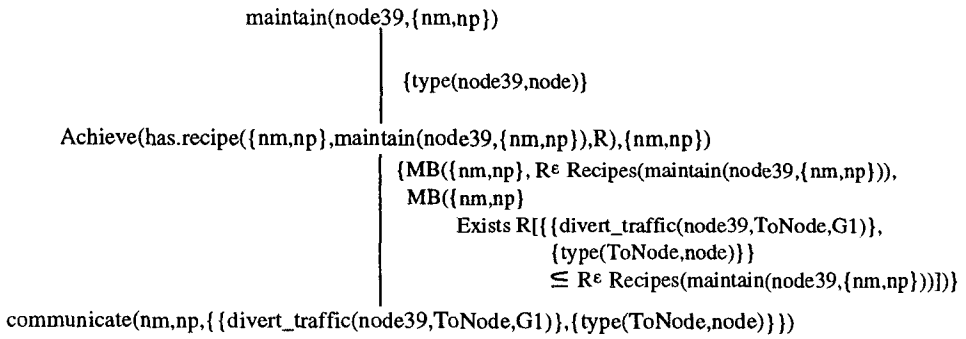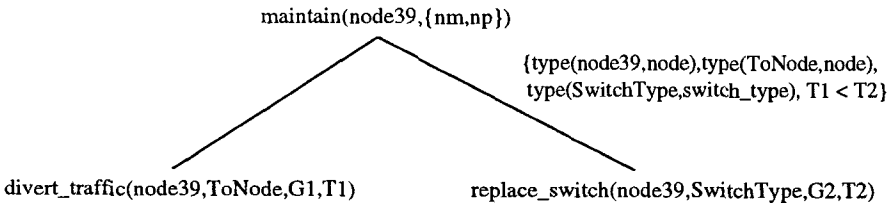$$communicate(Gk, Gm, \{\beta_i, \rho_j\}, T)$$

Figure 27
Recipes for obtaining recipes.

Next, we try to identify NM's communicative act in utterance (4) with some act in that recipe, and succeed by appropriately instantiating the *communicate* act. NP is thus able to make sense of NM's utterance based on his beliefs about ways of obtaining recipes. Now, however, he must decide whether the act that NM is proposing to include as part of their recipe for maintaining node39 is compatible with his beliefs about ways of performing that act. This reasoning is modeled by Step (3) of the augmentation process in which the constraints of the rgraph are checked for satisfiability. The recipe for obtaining recipes that was selected in Step (0) of the algorithm indicates that to have a recipe for maintaining node39, the agents must have mutual belief that some set of acts and constraints constitute a recipe for that act. If NP does not believe that the act *divert_traffic(Node1, Node2, G)* should play a role in maintaining node39, then the constraint will not hold and the algorithm will fail. NP will then communicate his dissent to NM and possibly propose an alternative act. In this instance, however, NP is in agreement with NM, as evidence by his "Okay" in utterance (5). The rgraph that results from his reasoning is shown in Figure 28.

To produce utterance (6), NP must reason about the state of the agents' Shared-Plans and determine what needs to be done to complete them. At this point in the discourse, the agents are focused on obtaining a recipe for maintaining node39 and have agreed that the act of diverting network traffic will be included in that recipe. NP might thus propose the performance of another act as part of their recipe. He does this in utterance (6). In utterance (7), NM agrees to the inclusion of that act.

maintain(node39,{nm,np})

{type(node39,node)}

Achieve(has.recipe({nm,np},maintain(node39,{nm,np}),R),{nm,np})

{MB({nm,np}, Rε Recipes(maintain(node39,{nm,np}))),
MB({nm,np}
        Exists R[{ {divert_traffic(node39,ToNode,G1)},
                        {type(ToNode,node)} }
                ⊆ Rε Recipes(maintain(node39,{nm,np}))])}

communicate(nm,np,{ {divert_traffic(node39,ToNode,G1)},{type(ToNode,node)} })

Figure 28
Rgraph explaining utterances (1)–(5) of the dialogue in Figure 25.

maintain(node39,{nm,np})

{type(node39,node),type(ToNode,node),
type(SwitchType,switch_type), T1 < T2}

divert_traffic(node39,ToNode,G1,T1)   replace_switch(node39,SwitchType,G2,T2)

Figure 29
Rgraph explaining utterances (1)–(8) of the dialogue in Figure 25.

To produce utterance (8), NP must once again reason about the state of the agents' plans. If he believes that diverting network traffic from node39 and then replacing that node with a higher capacity switch will result in maintaining node39, then he will believe that the agents have completed their SharedPlan in (P7) to obtain a recipe for maintaining the node. His utterance in (8) indicates that this is the case. Unless agent NM indicates her disagreement, NP will thus assume that the agents have completed their SharedPlan in (P7) and will update his beliefs accordingly. First, he will remove the SharedPlan in (P7) from further consideration; the agents have completed that plan and have thus satisfied the corresponding discourse purpose. The plan in (P7) is thus popped from NP's representation of the intentional structure. Second, NP will update his beliefs about the dominating plan in DSP_6 based on the knowledge gained during the subdialogue. In particular, the recipe that was decided upon to maintain node39 will be added to the plan and the rgraph will be updated accordingly. Figure 29 contains the rgraph representing the new discourse context after utterance (8).

Utterance (9) indicates the initiation of a new discourse segment, the purpose of which can be recognized as:

$DSP_8 = Int.Th(nm,$
$\qquad FSP(\{nm, np\},$
$\qquad\qquad Achieve(has.sat.descr(\{nm, np\}, ToNode, \mathcal{F}(divert\_traffic, ToNode)),$
$\qquad\qquad\qquad \{nm, np\}))).$

using CDR_A. As with the other types of subdialogues discussed above, once agent NP recognizes this DSP, he must determine its relationship to the other DSPs underlying the discourse. In this instance, the only other DSP is that underlying the entire discourse. To model agent NP's reasoning, we must thus determine the relationship of the SharedPlan in DSP_8 to that in DSP_6. The knowledge precondition requirements of the latter plan provide that explanation.
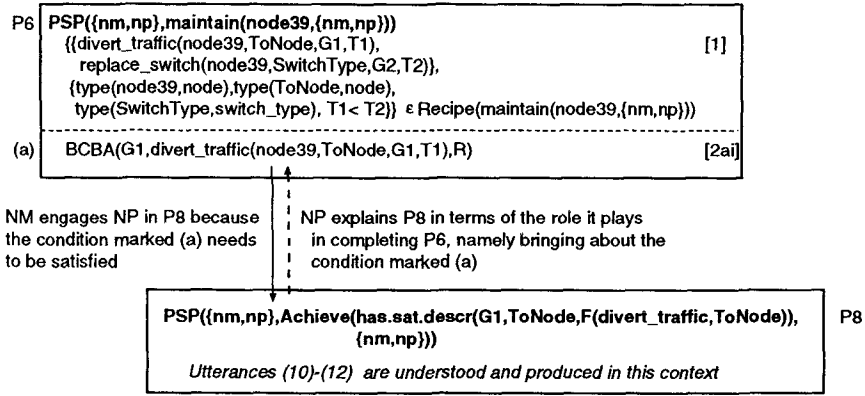
P6 | PSP({nm,np},maintain(node39,{nm,np}))
{{divert_traffic(node39,ToNode,G1,T1),   [1]
  replace_switch(node39,SwitchType,G2,T2)},
  {type(node39,node),type(ToNode,node),
  type(SwitchType,switch_type), T1< T2}} ε Recipe(maintain(node39,{nm,np}))

(a) | BCBA(G1,divert_traffic(node39,ToNode,G1,T1),R)   [2ai]

NM engages NP in P8 because | NP explains P8 in terms of the role it plays
the condition marked (a) needs | in completing P6, namely bringing about the
to be satisfied | condition marked (a)

PSP({nm,np},Achieve(has.sat.descr(G1,ToNode,F(divert_traffic,ToNode)),   P8
{nm,np}))

Utterances (10)-(12) are understood and produced in this context

Figure 30
Analysis of the second subdialogue in Figure 25.

$$Achieve(has.sat.descr(G,p_i, F(\bar{\alpha},p_i),T),\{G,G2\},T)$$

$$\{has.sat.descr(G,D,F(\bar{\alpha}, p_i),T)\}$$
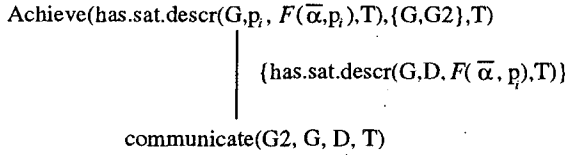
$$communicate(G2, G, D, T)$$

Figure 31
A recipe for obtaining a parameter description.

As discussed in Section 5.1.3, an agent G's ability to perform an act β depends in part on its ability to identify the parameters of β. Thus to perform the act *divert_traffic* (*node39, ToNode, G1*) as part of the agents' SharedPlan to maintain node39, the agents must be able to identify the *ToNode* parameter of the act. The need to identify this parameter thus provides NP with an explanation for DSP$_8$. In particular, NP can reason that NM initiated the new discourse segment in order to satisfy one of the ability requirements of the agents' SharedPlan to maintain node39. The SharedPlan in DSP$_8$ is thus subsidiary to that in DSP$_6$ by virtue of the BCBA requirements of the latter plan. Figure 30 summarizes our analysis of the subdialogue.

Once NP recognizes, and explains, the initiation of the new segment, he will produce his subsequent utterances in the context of its DSP, rather than the previous one, and will expect NM to do the same. The rgraph construction algorithm is used in modeling NP's reasoning. Whereas in the previous example, the algorithm makes use of recipes for obtaining recipes, in this case it makes use of recipes for obtaining parameter descriptions. Figure 31 contains an example of such a recipe. The recipe is derived from the definition of *has.sat.descr* in Figure 8 and represents that an agent G can bring about *has.sat.descr* of a parameter p_i by getting another agent $G2$ to give it a description D of p_i. The recipe's constraints, however, require that D be of the appropriate sort, according to the constraint $\mathcal{F}(\bar{\alpha},p_i)$, for the identification of the parameter to be successful (Appelt 1985; Kronfeld 1986, 1990; Hintikka 1978).

Given the discourse context represented by Figure 30 then, NP should respond to NM's utterance in (9) on the basis of his beliefs about ways in which to identify parameters. For example, if NP knows the recipe in Figure 31, then he might respond to NM by communicating some node description to her. As we noted in Section 1, however, the description that NP uses must be one that is appropriate for the current circumstances. In particular, NP should respond to NM with a description that will

enable both of the agents to identify the node for the purposes of diverting network traffic. The rgraph in Figure 29 and the constraints of the recipe in Figure 31 provide the necessary context for modeling NP's behavior. Because NP knows that the agents are trying to divert network traffic as part of maintaining node39, as represented by the rgraph in Figure 29, he should first choose a node that is appropriate for that circumstance. For example, he might choose a node that is spatially close to node39, rather than one that, while lightly loaded, is more distant. After selecting the node, NP should then choose a means of identifying it for NM. For example, he might present her with a diagram of the network and then tell her how to identify the particular node on the diagram; NP's response in utterances (10) and (11) takes this form. It would not be appropriate, however, for NP to respond to NM with some internal node name, or with a description like "the node with the lightest traffic," unless he believed that NM could identify the node on the basis of that description. The constraints of the recipe in Figure 31 model this requirement. They represent that the description communicated by an agent should be one that will allow the other agent to identify the object in question for the purpose of the act to be performed.

7. Comparison with Grosz and Sidner's Theory

Grosz and Sidner (1990) have argued that a theory of DSP recognition depends upon an underlying theory of collaborative plans. Although SharedPlans provide that latter theory, the connection between SharedPlans and DSPs was never specified. In this paper, we have presented a SharedPlan model for recognizing DSPs and their interrelationships. We now show that this model satisfies the requirements set out by Grosz and Sidner's (1986) theory of discourse structure. We first discuss the process by which intentional structure is recognized. Next, we discuss the way in which intentional structure interacts with the attentional state component of discourse structure. And finally, we discuss the contextual use of intentional structure in interpretation.

7.1 Recognizing Intentional Structure
7.1.1 Recognizing Discourse Segments and their Purposes. In their paper on discourse structure, Grosz and Sidner give several examples of the types of intentions that could serve as DSPs (Grosz and Sidner 1986, 179):

1. Intend that some agent intend to perform some physical task.

2. Intend that some agent believe some fact.

3. Intend that some agent believe that one fact supports another.

4. Intend that some agent intend to identify an object.

5. Intend that some agent know some property of an object.

Intentions such as these, as well as segment beginnings and endings, might be recognized on the basis of linguistic markers, utterance-level intentions, or knowledge about actions and objects in the domain of discourse (Grosz and Sidner 1986).

In our model, DSPs take the form $Int.Th(ICP, FSP(\{ICP, OCP\}, \beta))$. This type of DSP addresses several problems with the above examples—problems that motivated Grosz and Sidner's (1990) subsequent work on SharedPlans—namely the case of one agent intending another to do something and the so-called master/slave assumption. We recognize DSPs using the conversational default rule, CDR_A. This rule provides a means of recognizing the initiation of new segments and their purposes based on the

propositional content of utterances. Although this use of CDR_A is admittedly limited—it requires an ICP to communicate the act that it desires to collaborate on at the outset of a segment—other sources of information, such as those cited above, could also be incorporated into the model to aid in the recognition of new segments and their corresponding SharedPlans.

SharedPlans can also be used in recognizing the completion of discourse segments. Case (2b) of the augmentation process in Figure 13 outlines the required reasoning. A discourse segment is complete when all of the beliefs and intentions required to complete its corresponding SharedPlan have been established. This use of SharedPlans also appears at first glance to be of limited use—the mental attitudes required of a full SharedPlan may not all be explicitly established over the course of a dialogue or subdialogue. However, the OCP may be able to infer the completion of a SharedPlan, and thus the corresponding segment, in combination with information from other sources. For example, suppose an OCP has some reason to expect the end of a segment based on a linguistic signal such as an intonational feature (e.g., as described by Grosz and Hirschberg [1992]). If additionally the OCP is able to ascribe the various mental attitudes "missing" from the SharedPlan that corresponds to that segment, then the OCP has further evidence for the segment boundary. These mental attitudes may be ascribed on the basis of those of the OCP's beliefs that are in accord with the mental attitudes comprising the SharedPlan (Pollack 1986a; Grosz and Sidner 1990).

7.1.2 Recognizing Relationships between Discourse Segments. Once an OCP recognizes the initiation of a new discourse segment, it must determine the relationship of that segment's DSP to the other DSPs underlying the discourse (Grosz and Sidner 1986). In our model, relationships between SharedPlans provide the basis for determining the corresponding relationships between DSPs. An OCP must determine how the SharedPlan used to model a segment's DSP is related to the other SharedPlans underlying the discourse. The information that an OCP considers in determining this relationship is delineated by the beliefs and intentions that are required to complete each of the other plans. In this way, our model provides a more detailed account of the relationships that can hold between DSPs than did Grosz and Sidner's original formulation.

One DSP dominates another if the second provides part of the satisfaction of the first. In our model, subsidiary relationships between SharedPlans provide a means of determining dominance relationships between DSPs. If one plan is subsidiary to another, then the DSP that is modeled using the first plan is dominated by that modeled using the second. One DSP satisfaction-precedes another if the first must be satisfied before the second. This relationship corresponds to a temporal dependency between SharedPlans. When one SharedPlan must be completed before another, the DSP that is modeled using the first satisfaction-precedes that modeled using the second.

7.2 Relationship to Attentional State
The attentional state component of discourse structure is an abstraction of the discourse participants' focus of attention; it is modeled using a stack of focus spaces, one for each segment. Each focus space contains its segment's DSP, as well as those objects, properties, and relations that become salient over the course of the segment. One of the primary roles of the focus space stack is to constrain the range of DSPs to which a new DSP can be related; a new DSP can only be dominated by a DSP in some space on the stack.

In our model, a segment's focus space contains a DSP of the form $Int.Th(ICP, FSP(\{ICP, OCP\}, \beta))$. The operations on the focus space stack depend upon subsidiary rela-

tionships between SharedPlans in the same way that Grosz and Sidner (1986) describe the operations as depending upon DSP relationships. As each SharedPlan corresponding to a discourse segment is completed, the segment's focus space is popped from the stack. Only those SharedPlans in some space on the stack are candidates for subsidiary relationships. The use of the SharedPlan stack **S** in the augmentation process of Figure 13 reflects the operations of the focus space stack.

7.3 The Contextual Role of Intentional Structure

An utterance of a discourse can either begin a new segment of the discourse, complete the current segment, or contribute to it (Grosz and Sidner 1986). Each of these possibilities is modeled by a separate case within the augmentation process given in Figure 13. The initiation and completion of discourse segments was discussed in Section 7.1. Our discussion here is thus restricted to the case of an utterance's contributing to a discourse segment.

Under Grosz and Sidner's theory, each utterance of a discourse segment contributes some information towards achieving the purpose of that segment. In our model, each utterance is understood in terms of the information it contributes towards completing the corresponding SharedPlan. The FSP definition in Figure 5 constrains the range of information that an utterance of a segment can contribute towards the segment's SharedPlan. Hence, if an utterance cannot be understood as contributing information to the current SharedPlan, then it cannot be part of the current discourse segment. That is, the utterance must begin a new segment of the discourse or complete the current segment, but it cannot contribute to it. In this way, our model provides a more detailed account of the role that intentional structure plays as context in interpreting utterances than did Grosz and Sidner's original formulation.

Because each utterance of a discourse segment contributes some information towards the purpose of that segment, the segment's DSP may not be completely determined until the last utterance of the segment. However, as Grosz and Sidner (1986) have argued, the OCP must be able to recognize initially at least a generalization of the DSP so that the proper moves of attentional state can be made. Although CDR_A provides a limited method of recognizing new segments and their purposes, it does conform to this aspect of Grosz and Sidner's theory. In particular, the initial purpose of a segment, as recognized by CDR_A, is quite generally specified; it consists only of the intention that the agents form a SharedPlan. However, as the utterances of a discourse segment provide information about the details of that plan, the segment's purpose becomes more completely determined. In particular, the purpose comes to include the mental attitudes required of a full SharedPlan and established by the dialogue. Additionally, although the objective of the agents' plan may only be abstractly specified when it is initially recognized, it too may be further refined by the utterances of the segment.

8. Comparison with Previous Plan-Based Approaches

Early work on plan recognition in discourse (Allen and Perrault 1980; Cohen, Perrault, and Allen 1982) focused on the problem of reasoning about single utterances.[25] Subsequent work (Sidner and Israel 1981; Sidner 1983, 1985; Carberry 1987) extended the earlier approaches to recognize speaker's intentions across multiple utterances. All of

---

25 More recent work in the area of single utterance reasoning includes that of Cohen and Levesque (1990) and Perrault (1990). Their work provides a detailed mental state model of speech act processing and is thus focused at a different level of granularity than the work discussed in this paper.

(1)   User: Show me the generic concept called "employee".

(2) System: OK.   <system displays network>

(3)   User: I can't fit a new ic below it.

(4)   Can you move it up?

(5) System: Yes.   <system displays network>

(6)   User: OK, now make an individual employee concept
              whose first name is ...

Figure 32
A sample correction subdialogue (Sidner 1983; Litman 1985).

these approaches were based on a data-structure view of plans and were designed to recognize utterance-level intentions. More recent work (Litman 1985; Litman and Allen 1987; Lambert and Carberry 1991; Ramshaw 1991) has been concerned with the problems introduced by discourses containing subdialogues. However, the more recent work has followed in the tradition of the previous work and as a result continues to produce an utterance-to-utterance-based analysis of discourse, rather than one based on discourse structure. We now review these approaches and show that they are aimed at recognizing a different type of intention than that discussed in this paper.

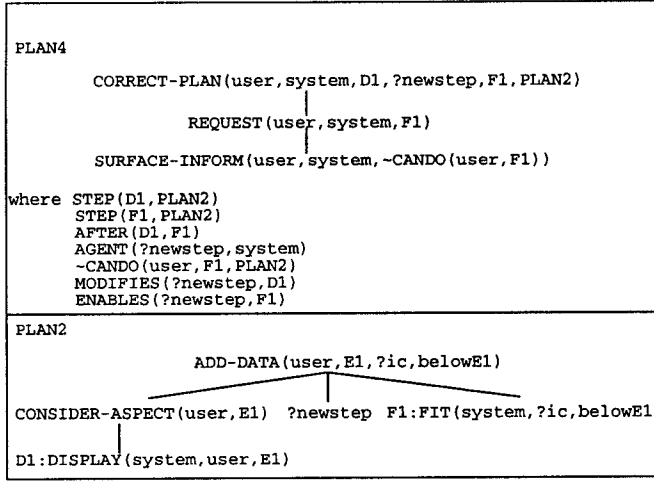### 8.1 The Approach of Litman and Allen

To model clarification and correction subdialogues, Litman and Allen propose the use of two types of plans: **discourse plans** and **domain plans** (Litman 1985; Litman and Allen 1987). Domain plans represent knowledge about a task, while discourse plans represent conversational relationships between utterances and plans. For example, an agent may use an utterance to *introduce, continue,* or *clarify* a plan.

In Litman and Allen's model, the process of understanding an utterance entails recognizing a discourse plan from the utterance and then relating that discourse plan to some domain plan; the link between plans is captured by the constraints of the discourse plan. For example, under Litman and Allen's analysis, utterance (3) of the dialogue in Figure 32 (repeated from Figures 2 and 19) is recognized as an instance of the CORRECT-PLAN discourse plan; with the utterance, the User is correcting a domain plan to add data to a network.
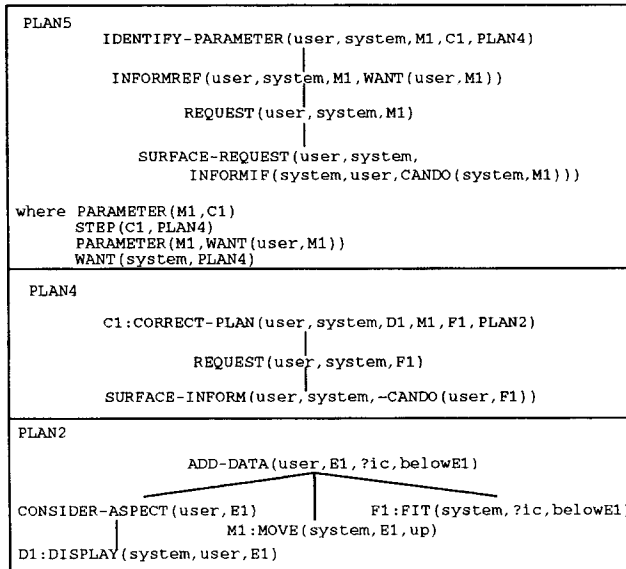
Litman and Allen use a stack of plans to model attentional aspects of discourse. The plan stack after processing utterance (3) is shown in Figure 33. The CORRECT-PLAN discourse plan on top of the stack indicates that the user and system are correcting a problem with the step labeled D1 in PLAN2 (the DISPLAY act of the ADD-DATA domain plan) by inserting a new step into PLAN2 (?newstep) before the step labeled F1 (the FIT step).

The plan stack after processing the User's subsequent utterance in (4) is shown in Figure 34. The IDENTIFY-PARAMETER discourse plan indicates that utterance (4) is being used to identify the ?newstep parameter of the CORRECT-PLAN discourse plan.

The boxes in Figures 33 and 34 do not correspond to discourse segments, but rather to individual utterances. PLAN5 in Figure 34 was introduced by utterance (4) in the dialogue, PLAN4 by utterance (3). The two utterances are linked together by the parameter M1, corresponding to the MOVE act in PLAN2. Although this analysis serves as a method of relating the two utterances, it provides only an utterance-to-utterance-based model of discourse processing. Intuitively, utterances (3)–(5) as a

```
PLAN4
            CORRECT-PLAN(user,system,D1,?newstep,F1,PLAN2)
                              |
                   REQUEST(user,system,F1)
                              |
              SURFACE-INFORM(user,system,~CANDO(user,F1))

where STEP(D1,PLAN2)
      STEP(F1,PLAN2)
      AFTER(D1,F1)
      AGENT(?newstep,system)
      ~CANDO(user,F1,PLAN2)
      MODIFIES(?newstep,D1)
      ENABLES(?newstep,F1)

PLAN2
                  ADD-DATA(user,E1,?ic,belowE1)
                         _____
CONSIDER-ASPECT(user,E1)   ?newstep  F1:FIT(system,?ic,belowE1)
       |
D1:DISPLAY(system,user,E1)
```

Figure 33
Plan stack after processing utterance (3) of the dialogue in Figure 32 (Litman 1985).

```
PLAN5
           IDENTIFY-PARAMETER(user,system,M1,C1,PLAN4)
                              |
          INFORMREF(user,system,M1,WANT(user,M1))
                              |
               REQUEST(user,system,M1)
                              |
             SURFACE-REQUEST(user,system,
                   INFORMIF(system,user,CANDO(system,M1)))
where PARAMETER(M1,C1)
      STEP(C1,PLAN4)
      PARAMETER(M1,WANT(user,M1))
      WANT(system,PLAN4)

 PLAN4
           C1:CORRECT-PLAN(user,system,D1,M1,F1,PLAN2)
                              |
                  REQUEST(user,system,F1)
                              |
             SURFACE-INFORM(user,system,~CANDO(user,F1))

PLAN2
                  ADD-DATA(user,E1,?ic,belowE1)
                         _____
CONSIDER-ASPECT(user,E1)   |      F1:FIT(system,?ic,belowE1)
                        M1:MOVE(system,E1,up)
       |
D1:DISPLAY(system,user,E1)
```

Figure 34
Plan stack after processing utterance (4) of the dialogue in Figure 32 (Litman 1985).

whole are concerned with correcting a problem; utterance (3) identifies the problem, while utterance (4) suggests a method of correcting it. Under Litman and Allen's analysis, however, utterance (3) is used to correct a problem and utterance (4) is used to identify a parameter in a discourse plan. This type of analysis cannot capture the contribution of a subdialogue to the overall discourse in which it is embedded. Each utterance is simply linked to one that precedes it, irrespective of how the utterances aggregate into segments.

In contrast to Litman and Allen's approach, our approach accurately reflects the compositional structure of discourse; utterances are understood in the context of discourse segments, and segments in the context of the discourse as a whole.[26] Our analysis of the dialogue in Figure 32 was discussed in Section 6.2.2 and is summarized by Figure 23. Under our analysis, utterance (3) introduces a new discourse segment, the purpose of which is to satisfy a constraint that there be enough free space on the screen to add a new concept. This new segment is recognized and explained based on the ability requirements of SharedPlans. Utterance (4) of the dialogue is understood in the context of this new discourse segment. In particular, the act of moving the generic concept up is understood as a means of satisfying the constraint.

In more recent work, Litman and Allen have augmented their model with a notion of "discourse intentions." "Discourse intentions are purposes of the speaker, expressed in terms of both the task plans of the speaker (the domain plans) and the plans recursively generated by these plans (the discourse plans)" (Litman and Allen, 1990, 376). For example, the discourse intention underlying utterance (4) can be glossed as:

User intends that System intends that
    System identify the ?newstep parameter
        of the CORRECT-PLAN discourse plan.

Because Litman and Allen's discourse and domain plans are recognized on the basis of a single utterance, their discourse intentions are actually utterance-level intentions, and not the type of discourse-level intentions discussed in this paper.

8.2 Other Approaches

Lambert and Carberry (1991) have revised Litman and Allen's dichotomy of plans into a trichotomy of **discourse, problem-solving,** and **domain plans**. Their discourse plans represent means of achieving communicative goals, while their problem-solving plans represent means of constructing domain plans. The Build-Plan operator in Figure 9 is an example of a problem-solving plan; it is used to represent the process by which two agents build a plan for one of them to do an action. The body of the operator requires that the agents (i) Build-Plans for the subacts of that action and (ii) Instantiate-Var(iable)s of those subacts.

In Lambert and Carberry's model, the process of understanding an utterance entails recognizing a tripartite structure of plans. Beginning from the surface-level form of an utterance, their system recognizes plans on the discourse level until a plan at that level can be linked to one on the problem-solving level; plans on the problem-solving level are then recognized until one can be linked to a plan on the domain level; further plans may then be recognized on that level.

---

26 Although there may be several possible segmentations of a discourse, just as there may be several possible parses of a sentence, there is general agreement that utterances do cluster into segments. The point here is that our analysis reflects this segmentation, whereas Litman and Allen's is utterance-to-utterance based and thus does not.

As a model of subdialogue understanding, Lambert and Carberry's approach suffers from problems similar to that of Litman and Allen's. In particular, Lambert and Carberry's analysis is still utterance-to-utterance based; subdialogues are not recognized as separate units, nor is a subdialogue's contribution to the discourse in which it is embedded recognized. This is also true of Lambert and Carberry's (1992) more recent work on modeling negotiation subdialogues. Although Lambert and Carberry emphasize the importance of recognizing the initiation of negotiation subdialogues, and work through an example involving an embedded negotiation subdialogue, they do not indicate how these subdialogues are actually recognized as such. The only possibility hinted at in the text (i.e., that the discourse act Address-Believability accounts for them) results in a discourse segmentation that does not accurately reflect the purposes underlying their sample dialogue.
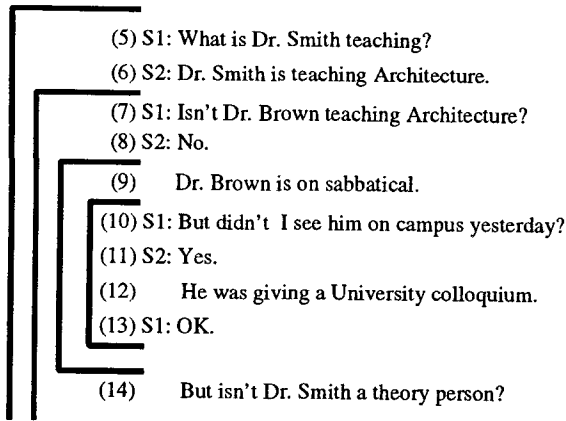
Figures 35 and 36 contain the sample dialogue used by Lambert and Carberry (1992). In Figure 35, the dialogue is segmented as suggested by Lambert and Carberry's analysis, while in Figure 36 it is segmented to more accurately reflect the purposes underlying the discourse. The subdialogues marked (b) and (d) in Figure 36 are both initiated by S1 and are each concerned with a different aspect of the accuracy of S2's utterance in (6). Segments (b) and (d) are thus siblings both dominated by segment (a) in Figure 36. Under Lambert and Carberry's analysis, however, these two subdialogues are not recognized as separate units. That they should be can be seen by the coherent discourses that remain if either is removed from the dialogue.

In addition, although the process of plan construction provides an important context for interpreting utterances, trying to formalize this mental activity under a data-structure approach results in a model that conflates recipes and plans (Pollack 1990). For example, each of Lambert and Carberry's domain act operators requires as a precondition that the agent have a plan to use that operator to perform the act. That requirement, however, results in the paradoxical situation whereby a recipe for an act α requires having a plan for α that uses that recipe. As another example, the Build-Plan operator in Figure 9 requires as a precondition that each agent know the referents of the subactions that one of the agents needs to perform to accomplish α. However, considering that determining how to perform an act is part of constructing a plan to perform that act, it is odd that a recipe for building a plan for α requires knowing the subactions of α as a precondition of its use. The fact that these inconsistencies do not seem to pose a problem for Lambert and Carberry's model is testament to its data-structure nature; the plan chaining behavior of their reasoner on the various types of operators is such that no circularities arise.
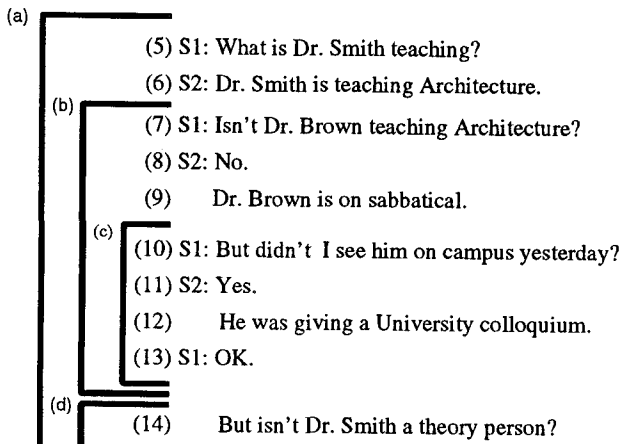
Ramshaw (1991) has augmented Litman and Allen's two types of plans with a different third type, **exploration plans**. This type of plan is added to distinguish those domain plans an agent has adopted from those it is simply considering adopting. In this model, understanding an utterance entails recognizing a discourse plan from the utterance and then relating that plan to a plan on either the exploration level or the domain level, as determined by the form of the utterance and the plan structures built from previous utterances. Like the previous approaches, however, Ramshaw's model is still utterance-to-utterance based. The three-level structure he manipulates on the basis of each user query does not account in any way for the structure of discourse.

9. Conclusion

In this paper, we have developed a computational model for recognizing the intentional structure of a discourse and using that structure in discourse processing. Shared-Plans are used both to represent the components of intentional structure, i.e., discourse

(5) S1: What is Dr. Smith teaching?

(6) S2: Dr. Smith is teaching Architecture.

(7) S1: Isn't Dr. Brown teaching Architecture?

(8) S2: No.

(9)   Dr. Brown is on sabbatical.

(10) S1: But didn't I see him on campus yesterday?

(11) S2: Yes.

(12)   He was giving a University colloquium.

(13) S1: OK.

(14)   But isn't Dr. Smith a theory person?

Figure 35
Lambert and Carberry's analysis.

(a)

(5) S1: What is Dr. Smith teaching?

(6) S2: Dr. Smith is teaching Architecture.

(b)

(7) S1: Isn't Dr. Brown teaching Architecture?

(8) S2: No.

(9)   Dr. Brown is on sabbatical.

(c)

(10) S1: But didn't I see him on campus yesterday?

(11) S2: Yes.

(12)   He was giving a University colloquium.

(13) S1: OK.

(d)

(14)   But isn't Dr. Smith a theory person?

Figure 36
Our analysis.

segment purposes and their interrelationships, and to reason about the use of intentional structure in utterance interpretation.

We have also shown that our work differs from previous plan-based approaches to discourse processing by providing a model for recognizing and reasoning with discourse-level intentions, rather than utterance-level intentions. The previous approaches address the problem of recognizing the propositional content of an utterance from its surface form, but provide only an utterance-to-utterance-based analysis of discourse. In contrast, we begin from propositional content and present a model of discourse processing that derives from discourse structure.

10. Future Directions

There are three main areas in which the research presented in this paper could be extended. The first involves the augmentation process, the second its use in modeling intentional structure, and the third its use in building collaborative agents.

10.1 The Augmentation Process

The augmentation process given in Figures 10 and 13 provides a novel framework for utterance interpretation based on discourse structure. This framework outlines the required steps of the interpretation process and provides constraints on the types of algorithms that may be used to model them. The rules and algorithms presented in Section 6.1 provide a means of modeling the central steps involved in the interpretation process, but are only a beginning. Further research is required to develop algorithms to model the remaining steps. For example, Case (2c) of the augmentation process models the process by which an agent recognizes the contribution of an utterance to the SharedPlan currently in focus. In elaborating this case, we concentrated on just one type of utterance. In particular, we focused on utterances that communicate information about a single action and reasoned only about that action, and not the other information communicated by the utterance. The augmentation process could thus be extended to include reasoning about other types of utterances, as well as to include reasoning about the information contained in those utterances. For example, Balkanski (1993) has shown that multi-action utterances convey a wealth of information about a speaker's beliefs and intentions. That information should also be taken into account in interpreting the agent's utterances.

Step (3b) of the augmentation process deals with the situation in which an agent does not understand, or disagrees with, its collaborative partner's utterances. The recognition of this case is modeled by the failure of the rgraph construction algorithm. This failure indicates that the algorithm was unable to produce an explanation for an act and thus that further communication and replanning are necessary. Our implementation of the algorithm (Lochbaum 1994) models one possible behavior of the agent in such circumstances. In particular, the implementation outputs the recipe it was trying to use to explain the act, along with an explanation for its failure. This information can be viewed as a starting point from which the agents may engage in a negotiation process. The details of that negotiation process are the subject of future research.

As we alluded to earlier in the paper, the process of constructing a SharedPlan may also be used to aid in the generation of utterances. Figure 37 provides a high-level specification of this process. It is based on the assumption that agents G_1 and G_2 are collaborating on an act α and indicates how the requirements of collaboration constrain the range of information that G_1 must consider in formulating his utterances. These requirements are maintained in G_1's **agenda**.[27] As indicated in Step (4) of Figure 37, G_1's agenda indicates those beliefs and intentions that are required for the agents to have a full SharedPlan for α, but that are absent from their current partial SharedPlan.

On the basis of his agenda, G_1 chooses an item to which to direct his attention, decides what he wants to say about that item, and then does so (Step (5b) of the process in Figure 37). The question of how the information on G_1's agenda is organized is the subject of future research. We have not specified the process by which an agent chooses what to communicate from among the possible options, or how it then does so.

Once G_1 has communicated some particular information to G_2, he waits for her response. If G_2 indicates her agreement with G_1, either explicitly or implicitly, G_1 then updates his beliefs about the agents' PSP to reflect the information he communicated (Step (6)).

The augmentation process given in Figure 37 provides a specification of the generation process at a much higher level of detail than previous work in generation. It

---

27 We follow Grosz and Kraus's (1993) terminology in our use of this term.

Assume:
 $PSP(\{G_1, G_2\}, \alpha)$,
 G_1 is the agent being modeled.
 G_1 is the speaker and must decide what to communicate.

4. G_1 inspects his beliefs about the state of the agents' PSP to determine what beliefs and intentions the agents must establish to complete it. Call this set G_1's *Agenda*.

5. (a) If the Agenda is empty, then G_1 believes that the agents' PSP is complete and so communicates that belief to G_2.

   (b) Otherwise, G_1
      i. chooses an item from the Agenda to establish,
      ii. decides upon a means of establishing it,
      iii. communicates his intent to G_2.

6. Unless G_2 disagrees, G_1 assumes mutual belief of what he communicated and updates his beliefs about the state of the agents' PSP accordingly.

Figure 37
The SharedPlan augmentation process—generation.

is concerned with participating in an extended discourse, while most work in generation has been concerned with generating short, isolated monologues. Moore and Paris (1993), however, have begun to look at the problem of participating in longer discourses and in particular at the problem of responding to follow-up questions. They have argued that while previous work in generation (e.g., the work of McKeown [1985], McCoy [1989], Paris [1988], and Hovy [1991]) has been concerned with *what* information to communicate and *how*, responding to follow-up questions also requires maintaining a record of *why* the information is being communicated. Without such a representation, the system cannot respond effectively when a hearer does not understand or accept its utterances. It remains to be determined how the process in Figure 37 meshes with previous work in generation. Our suspicion, however, is that as with interpretation, the augmentation process provides a model of discourse-level intentions, while work such as Moore and Paris's (1993) is really providing a model of utterance-level intentions. Both types of information are necessary for generating extended discourses, but serve different purposes.

10.2 Modeling Intentional Structure

In our model, SharedPlans and relationships among them provide the basis for computing intentional structure. We take DSPs to be of the form $Int.Th(ICP, FSP(\{ICP, OCP\}, \beta))$ and relationships between DSPs to depend upon subsidiary and temporal relationships between the corresponding SharedPlans. DSPs that do not involve SharedPlans would thus seem to present a problem for our model; however, many such DSPs may still be explained in terms of SharedPlans. For example, consider DSPs of the form "Intend that some agent intend to perform some physical task," as proposed by Grosz and Sidner (1986, 179). It is possible to explain this type of DSP in terms of the Int.To requirement of the FSP definition (Clause (2a) in Figure 5). According to that requirement, each of the single-agent acts in the agents' recipe must be intended by one of the two collaborating agents. This requirement might lead one of the agents, say G_1, to engage in a subdialogue to convince the other agent, G_2, to adopt such an intention. The DSP of this subdialogue would be represented as $Int.Th(G_1, Int.To(G_2, \beta_i))$ and corresponds to the English gloss given above. Although this DSP does not involve a SharedPlan, it is still motivated and explained by the

requirements of the FSP definition. Further research is required to completely develop this extension.

10.3 Building Collaborative Agents

Although issues in discourse processing provided the original motivation for Shared-Plans, Grosz and Kraus's more recent work (1993, 1996) has also shown the importance of the formalism to building collaborative agents. The work presented in this paper also contributes to that aspect of SharedPlans. The SharedPlan definitions delineate the information about which collaborating agents must communicate, whether they communicate in a natural language or an artificial one. The model of discourse processing developed in this paper provides a means of processing the agents' communications regardless of the form in which they occur. Rich and Sidner's (1998) work on COLLA-GEN demonstrates the use of the model with an artificial discourse language (Sidner 1994).

References

Allen, James F. 1983. Recognizing intentions from natural language utterances. In M. Brady and R. C. Berwick, editors, *Computational Models of Discourse*. MIT Press, Cambridge, MA, pages 107–166.

Allen, James. F. and C. Raymond Perrault. 1980. Analyzing intention in utterances. *Artificial Intelligence*, 15:143–178.

Ansari, Daniel. 1995. Deriving Procedural and Warning Instructions from Device and Environment Models. Master's thesis, University of Toronto.

Appelt, Douglas. 1985. Some pragmatic issues in the planning of definite and indefinite noun phrases. In *Proceedings of the 23rd Annual Meeting, Association for Computational Linguistics*, pages 198–203, Chicago, IL.

Appelt, Douglas and Amichai Kronfeld. 1987. A computational model of referring. In *Proceedings of IJCAI-87*, pages 640–647, Milan, Italy.

Balkanski, Cecile T. 1993. *Actions, Beliefs and Intentions in Multi-Action Utterances*. Ph.D. thesis, Harvard University.

Barrett, Anthony and Daniel S. Weld. 1994. Task decomposition via plan parsing. In *Proceedings of AAAI-94*, pages 1117–1122, Seattle, WA.

Brachman, Ronald J. and James G. Schmolze. 1985. An overview of the KL-ONE knowledge representation system. *Cognitive Science*, 9:171–216.

Bratman, Michael E. 1992. Shared cooperative activity. *The Philosophical Review*, 101:327–341.

Carberry, Sandra. 1987. Pragmatic modeling: Toward a robust natural language interface. *Computational Intelligence*, 3:117–136.

Cohen, Philip R. and Hector J. Levesque. 1990. Rational interaction as the basis for communication. In P. R. Cohen, J. L. Morgan, and M. E. Pollack, editors, *Intentions in Communication*. MIT Press, Cambridge, MA, pages 221–255.

Cohen, Philip R. and C. Raymond Perrault. 1979. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3:177–212.

Cohen, Philip R., C. Raymond Perrault, and James F. Allen. 1982. Beyond question-answering. In W. Lehnert and M. Ringle, editors, *Strategies for Natural Language Processing*. Lawrence Erlbaum Associates, Hillsdale, NJ, pages 245–274.

Fikes, Richard E. and Nils J. Nilsson. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208.

Grice, H. P. 1969. Utterer's meaning and intentions. *Philosophical Review*, 68(2):147–177.

Grosz [Deutsch], Barbara J. 1974. The structure of task-oriented dialogs. In *IEEE Symposium on Speech Recognition: Contributed Papers*, pages 250–253, Pittsburgh, PA.

Grosz, Barbara J. and Julia Hirschberg. 1992. Some intonational characteristics of discourse structure. In *Proceedings of the International Conference on Spoken Language Processing*, pages 429–432, Banff, Alberta, Canada.

Grosz, Barbara J. and Sarit Kraus. 1993.
Collaborative plans for group activities.
In *Proceedings of IJCAI-93*, pages 367–373,
Chambery, Savoie, France.

Grosz, Barbara J. and Sarit Kraus. 1996.
Collaborative plans for complex group
action. *Artificial Intelligence*, 86(2):269–357.

Grosz, Barbara J. and Candace L. Sidner.
1986. Attention, intentions, and the
structure of discourse. *Computational
Linguistics*, 12(3):175–204.

Grosz, Barbara J. and Candace L. Sidner.
1990. Plans for discourse. In P. R. Cohen,
J. L. Morgan, and M. E. Pollack, editors,
Intentions in Communication. MIT Press,
Cambridge, MA, pages 417–444.

Hintikka, Jaakko 1978. Answers to
questions. In H. Hiz, editor, *Questions*. D.
Reidel, Dordrecht, Holland, pages
279–300.

Hobbs, Jerry R. 1985. Ontological
promiscuity. In *Proceedings of the 23rd
Annual Meeting, Association for
Computational Linguistics*, pages 61–69,
Chicago, IL.

Hovy, Eduard. 1991. Approaches to the
planning of coherent text. In C. L. Paris,
W. R. Swartout, and W. C. Mann, editors,
*Natural Language Generation in Artificial
Intelligence and Computational Linguistics*.
Kluwer Academic Publishers, Boston,
MA, pages 83–102.

Kautz, Henry A. 1990. A circumscriptive
theory of plan recognition. In P. R. Cohen,
J. L. Morgan, and M. E. Pollack, editors,
Intentions in Communication. MIT Press,
Cambridge, MA, pages 105–134.

Kronfeld, Amichai. 1986. Donnellan's
distinction and a computational model of
reference. In *Proceedings of the 24th Annual
Meeting, Association for Computational
Linguistics*, pages 186–191, New York, NY.

Kronfeld, Amichai. 1990. *Reference and
Computation*. Cambridge University Press,
Cambridge, England.

Lambert, Lynn and Sandra Carberry. 1991.
A tripartite plan-based model of dialogue.
In *Proceedings of the 29th Annual Meeting,
Association for Computational Linguistics*,
pages 47–54, Berkeley, CA.

Lambert, Lynn and Sandra Carberry. 1992.
Modeling negotiation subdialogues. In
*Proceedings of the 30th Annual Meeting,
Association for Computational Linguistics*,
pages 193–200, Newark, DE.

Litman, Diane J. 1985. *Plan Recognition and
Discourse Analysis: An Integrated Approach
for Understanding Dialogues*. Ph.D. thesis,
University of Rochester.

Litman, Diane J. and James F. Allen. 1987. A
plan recognition model for subdialogues

in conversations. *Cognitive Science*,
11:163–200.

Litman, Diane J. and James F. Allen. 1990.
Discourse processing and commonsense
plans. In P. R. Cohen, J. L. Morgan, and
M. E. Pollack, editors, *Intentions in
Communication*. MIT Press, Cambridge,
MA, pages 365–388.

Lochbaum, Karen E. 1991. An algorithm for
plan recognition in collaborative
discourse. In *Proceedings of the 29th Annual
Meeting, Association for Computational
Linguistics*, pages 33–38, Berkeley, CA.

Lochbaum, Karen E. 1994. *Using
Collaborative Plans to Model the Intentional
Structure of Discourse*. Ph.D. thesis,
Harvard University. Available as
Technical Report TR-25-94, Center for
Research in Computing Technology,
Division of Applied Sciences.

Lochbaum, Karen E., Barbara J. Grosz, and
Candace L. Sidner. 1990. Models of plans
to support communication: An initial
report. In *Proceedings of AAAI-90*, pages
485–490, Boston, MA.

McCarthy, John and Patrick J. Hayes. 1969.
Some philosophical problems from the
standpoint of artificial intelligence. In
B. Meltzer and D. Michie, editors, *Machine
Intelligence 4*. Edinburgh University Press,
Edinburgh, pages 463–502.

McCoy, Kathleen F. 1989. Generating
context sensitive responses to
object-related misconceptions. *Artificial
Intelligence*, 41(2):157–195.

McKeown, Kathleen R. 1985. Discourse
strategies for generating natural language
text. *Artificial Intelligence*, 27:1–42.

Moore, Johanna D. and Cécile L. Paris. 1993.
Planning text for advisory dialogues:
Capturing intentional and rhetorical
information. *Computational Linguistics*,
19(4):651–694.

Moore, Robert C. 1985. A formal theory of
knowledge and action. In J. R. Hobbs and
R. C. Moore, editors, *Formal Theories of the
Commonsense World*. Ablex Publishing
Corp., Norwood, NJ, pages 319–358.

Morgenstern, Leora. 1987. Knowledge
preconditions for actions and plans. In
*Proceedings of the 10th International Joint
Conference on Artificial Intelligence
(IJCAI-87)*, pages 867–874, Milan, Italy.

Morgenstern, Leora. 1988. *Foundations of a
Logic of Knowledge, Action, and
Communication*. Ph.D. thesis, New York
University.

Paris, Cécile L. 1988. Tailoring object
descriptions to the user's level of
expertise. *Computational Linguistics*,
14(3):64–78.

Perrault, C. Raymond. 1990. An application of default logic to speech act theory. In P. R. Cohen, J. L. Morgan, and M. E. Pollack, editors, *Intentions in Communication*. MIT Press, Cambridge, MA, pages 161–186.

Pollack, Martha E. 1986a. *Inferring Domain Plans in Question-Answering*. Ph.D. thesis, University of Pennsylvania.

Pollack, Martha E. 1986b. A model of plan inference that distinguishes between the beliefs of actors and observers. In *Proceedings of the 24th Annual Meeting, Association for Computational Linguistics*, pages 207–214, New York, NY.

Pollack, Martha E. 1990. Plans as complex mental attitudes. In P. R. Cohen, J. L. Morgan, and M. E. Pollack, editors, *Intentions in Communication*. MIT Press, Cambridge, MA, pages 77–103.

Ramshaw, Lance A. 1991. A three-level model for plan exploration. In *Proceedings of the 29th Annual Meeting, Association for Computational Linguistics*, pages 39–46, Berkeley, CA.

Rich, Charles and Candace L. Sidner. 1998. COLLAGEN: A collaboration manager for software interface agents. *User Modeling and User-Adapted Interaction*.

Sacerdoti, Earl D. 1977. *A Structure for Plans and Behavior*. North-Holland, Amsterdam, Netherlands.

Searle, John R. 1990. Collective intentions and actions. In P. R. Cohen, J. L. Morgan, and M. E. Pollack, editors, *Intentions in Communication*. MIT Press, Cambridge, MA, pages 401–416.

Sidner, Candace L. 1983. What the speaker means: The recognition of speakers' plans in discourse. *Computers and Mathematics with Applications*, 9:71–82.

Sidner, Candace L. 1985. Plan parsing for intended response recognition in discourse. *Computational Intelligence*, 1(1):1–10.

Sidner, Candace L. 1994. An artificial discourse language for collaborative negotiation. In *Proceedings of AAAI-94*, pages 814–819, Seattle, WA.

Sidner, Candace L. and David J. Israel. 1981. Recognizing intended meaning and speakers' plans. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI-81)*, pages 203–208, Vancouver, British Columbia, Canada.

Vilain, Marc. 1990. Getting serious about parsing plans: A grammatical analysis of plan recognition. In *Proceedings of AAAI-90*, pages 190–197, Boston, MA.