CONVERSATIONAL AGENCY:
THE TRAINS-93 DIALOGUE MANAGER

David R. Traum
TECFA, FPSE
Université de Genève
9 Rte de Drize, Bat D
1227 Carouge, Switzerland.
David.Traum@tecfa.unige.ch

ABSTRACT

Designing an agent to participate in natural conversation requires more than just adapting a standard agent model to perceive and produce language. In particular, the model must be augmented with social attitudes (including mutual belief, shared plans, and obligations) and a notion of discourse context. The dialogue manager of the TRAINS-93 NL conversation system embodies such an augmented theory of agency. This paper focuses on the representation of mental state and discourse context and the deliberation strategies used in the agent model of the dialogue manager.

1 INTRODUCTION

A dialogue manager is that part of a dialogue system that connects the I/O devices and translators (whether they be spoken or typed language, a command language, menu selection, graphical presentation, etc.) to the parts that do the domain task reasoning and performance. In a simple language front-end system (e.g., for querying a database), dialogue management can be little more than a transducer from the I/O language to the task command language. However, for actual dialogue, this will also require some sensitivity to dialogue context – both for more flexible interpretation and more appropriate reaction. Figure 1 illustrates the connection of a dialogue manager to the rest of the system (in particular, the TRAINS-93 dialogue manager, described below in Section 4).

There are two main views of dialogue systems (and AI programs in general). One is as a tool designed for a particular task, the other is as an agent, with it's own *mental state*. In the case of dialogue systems, the tool view is usually as a front-end interface to a task module with which the user would like to engage in a more flexible interaction.

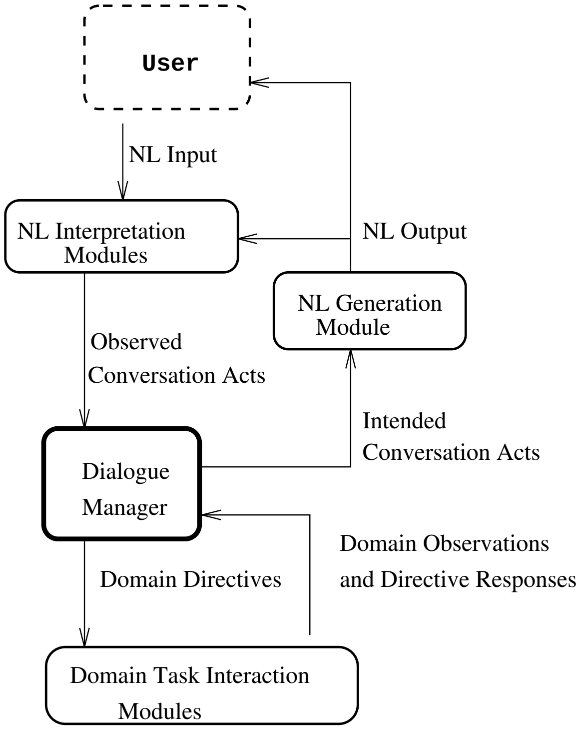

Figure 1: The TRAINS-93 System Architecture from a Dialogue Management Perspective

One claim of this paper is that it is more beneficial to design a dialogue system as an agent than as a front-end interface program. Since knowledge of many aspects of the domain task will be necessary even for accurate language interpretation, and dialogue structure is closely linked to task structure [Grosz and Sidner, 1986], for complex tasks (requiring more than a single interactional exchange to complete) a dialogue manager can not properly do it's job without at least an abstract notion of the internals of the task processing. Although the architecture in Figure 1 may look like that of a front-end system, this is misleading, since the NL interpretation modules will need to consult with the dialogue manager and the domain task modules in order to get a full contextual interpretation of the utterances, while the dialogue manager will be responsible for the manner in which information produced by the domain specialists gets reported back to the user, following the conventions of natural language conversation.

In the next section, we outline some previous work on computational agents. This is followed by a discussion of how these models of agency must be extended to handle dialogue. In section 4, an example conversational agent is presented – the TRAINS-93 dialogue manager.

2 ARTIFICIAL AGENCY

There are currently many notions of agency, as discussed in [Wooldridge and Jennings, 1995]. For this paper, we consider an agent to be an entity with it's own mental state and capacity for autonomous action. People are obviously prime examples of agents, however it can also be beneficial to view other entities (such as corporations or computer systems) as agents as well, as long as similar concepts can be fruitfully used to analyze their (past and future) behavior. McCarthy has a nice characterization of when it is useful to ascribe mental qualities to machines [McCarthy, 1990]. In particular, while some of the more emotional or particularly human qualities may not be useful, other qualities, such as beliefs and intentions can provide compact and easy to understand descriptions of the functionings of machines, as well as people.

Given that one can use mentalistic notions to describe machines, the important questions become *which* mental attitudes are most appropriate, how are they modeled, and how do they interact. [Wooldridge and Jennings, 1995] note that most theories include at least one *informational attitude* (such as knowledge or belief) and one *pro-attitude* (such as desires, goals, or intentions).

One popular formulation is the , *BDI* model, which include **Beliefs**, **Desires**, and **Intentions** as the primary mental attitudes. Figure 2 shows a schematic of such an architecture, simplified from [Bratman *et al.*, 1988].[1] In this figure, the attitudes are shown as ovals, while the processes are shown as boxes. Arrows represent the flow of influence — inputs for the processes and changes for the attitudes. The beliefs are the agent's model of the world – including both the current state, and how things have been in the past and what they are likely to be like in the future, given the performance of particular actions. By reasoning about its beliefs, the agent can derive new beliefs. Desires represent how the agent would *like* the world to be in the future. These will be the ultimate motive force in making the agent be more than just a passive observer of the world. The deliberation process will be one of deciding *which* actions should be performed so as to best achieve the desires. This process must consider the desires themselves, beliefs about how the world is now, beliefs about what kinds of actions are possible and what effects they achieve, as well as reasoning about the effects of the totality of actions under consideration.

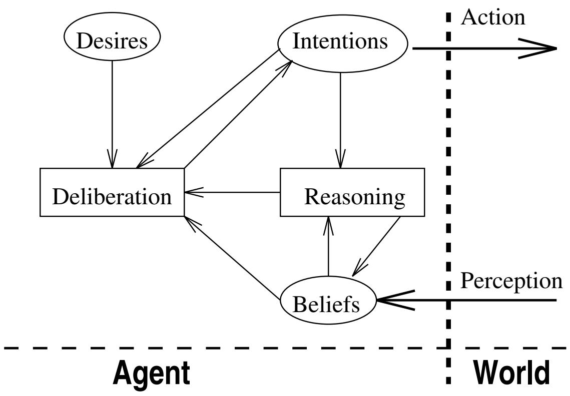

Figure 2: BDI Agent Architecture

The outcome of this deliberation process will be structured plans which the agent has decided to perform. These are the *Intentions* of the agent. According to Bratman's theory of intention [Bratman, 1987], these will play several important roles – most centrally, they contribute to the agent actually performing the action. Also, they will constrain future deliberation of an agent so as not to form conflicting intentions (if a conflicting desire is judges to be more important, a prior intention can be dropped). Also, the agent will monitor the success of actions and the achievability of intentions, replanning when needed in order to achieve the intentions. Some intentions which cannot be performed directly may also require further planning and deliberation (and adoption of intention) in order to be achieved.

The agent interacts with the world by performing actions and perceiving aspects of the world, including changes which result from these actions. Perceptions will influence the beliefs of the agent, while actions may change aspects of the world.

This general framework illustrates how mental at-

[1]The original figure included separate boxes for *means-ends reasoning* (i.e., planning), and a plan library - here we include this as part of the more general beliefs and reasoning. Also, Bratman, Pollack, and Israel included separate boxes for an opportunity analyzer and a filtering process, which here are included as part of the deliberation.

titudes and processes can be used to characterize the behavior of an agent. In [Bratman *et al.*, 1988], the focus is on the role of resource bounds on various aspects of the deliberation process. This architecture was also tested and found to perform adequately in the Tileworld domain [Pollack and Ringuette, 1990]. Others have attempted to formalize these attitudes and their relations, e..g., Rao and Georgeff have used a branching time semantics to axiomatize some of the relations of these attitudes.

REQUEST(speaker,hearer,act)
body: MB(hearer, speaker,
speaker WANT hearer DO act)
effect: hearer WANT hearer DO act

SURFACE-REQUEST(speaker,hearer,act)
effect: MB(hearer, speaker,
speaker WANT hearer DO act)

Figure 3: Allen (83) Speech Act Operators

3 CONVERSATIONAL AGENCY

While the BDI model has been useful for modeling a solitary agent, the question arises as its adequacy for handling multi-agent phenomena, and specifically dialogue. As a first attempt, one could view the perception from Figure 2 as a process of natural language interpretation, and the action as natural language generation. Austin introduced the term *speech acts* to describe the view of language as performance of actions [Austin, 1962]. Work by Allen, Cohen, and Perrault (e.g., [Cohen, 1978, Cohen and Perrault, 1979, Perrault and Allen, 1980, Allen, 1983] has brought this view within the computational community, by expressing these speech acts within AI theories of actions, with preconditions and effects relating to the mental states of the agents.

There are still several shortcomings with the BDI model of Figure 2, applied to conversation. These can be illustrated by a closer examination of AI planning operators for speech acts. Figure 3 shows one formulation of a **request** speech act, in which a speaker requests that a hearer perform some action. The body field represents a decomposition of the action into primitives; any action (or combination of actions) which achieve this body will count as performing a request action. In particular, the **surface-request** act which has this condition as its effect will be a request. A surface request, according to this formulation, is the utterance any imperative sentence, such as "Please make Orange Juice." Imperative sentences (surface-requests) are not the only way to perform requests, however. Utterance of another sentence, such as "We need to make Orange Juice", can also be a request, as long as it has the body conditions as an effect.

Getting back to Figure 2, we can see that several of the necessary aspects of the operators in Figure 3 are present, although some are missing. Most basicly, a deficiency with Figure 2, even as a solitary agent model is that the arrow from perceptions affects only beliefs. More generally, perceptions can affect desires and other attitudes as well as beliefs. This can be seen in the effects of the request act in Figure 3. A second problem is that Figure 2 shows only the agent and the world. Yet, for conversation, there are at least two agents involved (speaker and hearer in Figure 3). A model of conversational agency must include a model of the conversational partner as well as of the deliberating agent and the world. As illustrated by the operators in Figure 3, this must include beliefs about the other agent's mental state, as well as desires (or other pro-attitudes) that the other agent perform some action (such as a conversational response). In addition to simple nested attitudes about another agent, we claim that *social attitudes*, which link multiple agents, are also necessary to accurately model dialogue phenomena. **Mutual Belief** (MB) is one popular attitude, as illustrated in Figure 3.

A final missing ingredient for dialogue modeling is a notion of discourse context. Context will play an important part in deciding which of the possible acts are actually being performed. This context will include aspects of the mental state but will also include other aspects of the interaction, including the sequential patterns of action between the agents. In the rest of this section, we will discuss our own work in extending models of agency to be suitable for conversation, whereas dialogue context will be brought up again in the following section.

3.1 DISCOURSE OBLIGATIONS

Although Figure 3 correctly shows that attitudes other than belief must also be affected by action and perception, we have claimed that this desire of the hearer is proper effect of a request [Traum and Allen, 1994]. To examine this issue in more detail, we consider speech act plans from a related formalism [Cohen and Perrault, 1979]. Figure 4, from shows the planning process that an agent might use to decide to issue a request, given the desire for some action to be performed. Here, one agent, S, wants anther agent, John, to perform some action α. S realizes that if John has a desire to perform α, then John may perform it.[2] Now,

[2] In the Cohen and Perrault formalism, this link is called a *want precondition*. Using the BDI model, however, we can assume that

S must find and perform some action that will have as an effect that John wants to perform α. Here, an assumption of cooperativity is assumed, such that if John realizes that S wants him to do α, John himself will want to do α, as well. And it is assumed that the direct effect of the request is a belief that S wants John to perform α. The upshot is that S can perform this request and then rely on John's rational processes of perception, reasoning, and deliberation to intend and perform α.

$$
\begin{array}{c}
\alpha\,(\text{JOHN}) \\
\downarrow \text{ want.pr} \\
\text{JOHN WANT } \alpha\,(\text{JOHN}) \\
\uparrow \text{ effect} \\
\text{CAUSE-TO-WANT(S,JOHN, } \alpha\,(\text{JOHN})) \\
\downarrow \text{ cando.pr} \\
\text{JOHN BELIEVE S WANT } \alpha\,(\text{JOHN}) \\
\uparrow \text{ effect} \\
\text{REQUEST(S,JOHN, } \alpha\,(\text{JOHN}))
\end{array}
$$

Figure 4: Cohen & Perrault (79) Plan for Request

This model has a wide range of applicability in cooperative situations, and has also been used to explain why an answer follows a question (i.e., if α is the performance of some speech act). This account is missing something important, however. First, it associates a request with *any* discovery of a desire by another agent. While an agent *can* use indirect means to perform a request, intuitively there is a difference between helpfully acting to fulfill a discovered desire and performing an action that has actually been requested; informing of a desire is not *always* requesting that the desire be met. More crucially, consider the cases in which an agent is *not* disposed to be cooperative. Here, according to the plan in Figure 4, the cause-to-want action will be blocked, and John might not perform α. Also, even if the agent is disposed to be cooperative, perhaps he does not have the ability to perform α. As a third case, the agent might have some prior goal or intention *not* to do α. While this is fine, as far as it goes, in conversation, the agent will generally respond with *something*, even if it is not the desired action. What is it that inspires the agent to respond in these cases?

We claim that the more direct effect of a question, and the motivating factor in both the cooperative and non-cooperative setting is an **obligation** to respond. Obligations represent what an agent *should* do, according to some set of norms. The notion of obligation is a topic of much study, with some aspects are formalized as Deontic Logic [von Wright, 1951, McCarthy, 1990]. These logics allow one to define permissible, obligatory, and forbidden actions. Just because an action is obligatory with respect to a set of rules does not mean that an agent will actually perform the action. So we do not adopt the social agent model suggested by [Shoham and Tennenholtz, 1992] in which agents' behavior cannot violate the defined social laws. If an obligation is not satisfied, then this means that one of the rules must have been broken. We assume that agents generally plan their actions to violate as few rules as possible, and so obligated actions will usually occur. But when they directly conflict with the agent's personal goals, the agent may choose to violate them. Obligations are quite different from and can not be reduced to intentions and goals. In particular, an agent may be obliged to do an action that is contrary to his goals. Obligations thus form a dual to desires as motivating inputs to the deliberation procedure that leads to the adoption of intentions. An agent must consider both in deciding which actions to perform and in setting priorities for this action.

Specific obligations arise from a variety of sources. In a conversational setting, an accepted offer or a promise will incur an obligation. Also, a command or request by the other party will bring about an obligation to perform the requested action. If the obligation is to say something then we call this a *discourse obligation*. Our model of obligation is very simple. We use a set of rules that encode discourse conventions. Whenever a new conversation act is determined to have been performed, then any future action that can be inferred from the conventional rules becomes an obligation. We use a simple forward chaining technique to introduce obligations.

Some obligation rules based on the performance of conversation acts are summarized in Table 1. When an agent performs a promise to perform an action, or performs an acceptance of a suggestion or request by another agent to perform an action, the agent obliges itself to achieve the action in question. When another agent requests that some action be performed, the request itself brings an obligation to address the request: that is, either to accept it or to reject it (and make the decision known to the requester) – the requestee is not *permitted* to ignore the request. A question establishes an obligation to answer the question. If an utterance has not been understood, or is believed to be deficient in some way, this brings about an obligation to repair the utterance.

John's deliberation process can lead to his adopting an intention to perform α, given the desire (and a lack of conflicting desires).

source of obligation	**obliged action**
S_1 Accept or Promise A	S_1 achieve A
S_1 Request A	S_2 address Request: accept **or** reject A
S_1 YNQ whether P	S_2 Answer-if P
S_1 WHQ P(x)	S_2 Inform-ref x
utterance not understood or incorrect	repair utterance
S_1 Initiate DU	S_2 acknowledge DU
Request Repair of P	Repair P
Request Acknowledgement of P	acknowledge P

Table 1: Sample Obligation Rules

The model of obligations as the main effect of requests leads to both broader coverage and a more direct planning and deliberation procedure than the Cohen and Perrault model of Figure 4. Our model of the cooperative case is shown in Figure 5. Here, the obligation is the direct result, which feeds into the deliberation process. If the agent John is cooperative, without conflicting desires, then as before, the deliberative process will lead to the adoption of an intention to perform α. In the case in which there is some conflicting desire, the obligation to respond still motivates *some* action, in this case some other β, such as an evasion or refusal.

$$
\begin{array}{c}
\alpha\text{ (JOHN)} \\
\uparrow \\
\text{JOHN INTEND } \alpha\text{ (JOHN)} \\
\uparrow \text{ Deliberation} \\
\text{OBLIGED(JOHN, S, ADDRESS REQUEST(...))} \\
\uparrow \text{ effect} \\
\text{REQUEST(S,JOHN, } \alpha\text{ (JOHN))}
\end{array}
$$

Figure 5: Traum & Allen (94) Model of Requests

Obligations will thus form an important part of the reasoning process of a deliberative agent. In addition to considering **beliefs** about the world, which will govern the *possibility* of performing actions and likelihood of success, and **desires** or goals which will govern the *utility* or desirability of actions, a social agent will also have to consider **obligations**, which govern the *permissibility* of actions. Obligations, like mutual beliefs, are social attitudes, which link one agent to others in the society.

3.2 Mutual Belief and Grounding

Another deficiency of the speech act model in Figure 3 (and in fact almost all prior speech act formalisms in dialogue systems) is the way mutual belief (MB) is assumed to be the direct result of the utterance of a sentence, such as an imperative. In fact, examining actual conversation reveals that there is an elaborate process of feedback that can accompany initial utterances, and it is generally only after some sort of acknowledgment that the assumption of mutual belief is made. One reason for this is that both the language production and language interpretation are error-prone processes. A speaker cannot assume with any confidence that her contribution has been understood without some feedback from her interlocutor. Lack of understanding can be signaled with some sort of repair or request for repair. In cases in which the speaker does not receive any feedback, one can observe requests for acknowledgment or repetitions and refashionings of the original contribution in an attempt to elicit some kind of feedback. As an example of the prevalence of this kind of feedback in spoken dialogue, we observed that over half of the speaker transitions in a task-oriented dialogue corpus commenced with a simple acknowledgment, such as "ok", whereas another 30% provided direct evidence of the level of understanding by continuing with related content, while 15% followed turns consisting only of such acknowledgments. Only 2-5% of turn-transitions commenced with material unrelated to the previous turn [Traum and Heeman, 1996].

Clark and Shaefer call this process of reaching mutual belief (or common ground) *grounding* [Clark and Schaefer, 1989]. They present a descriptive model, in terms of presentation and acceptance phases that allow them to track the augmentation of common ground as the conversation proceeds. Their model is not well-suited for an on-line agent involved in dialogue, however, since it requires examination of subsequent spans of text in order to determine the boundaries of these phases.

We have built on this work, adapting it to something more useful for an on-line agent by presenting a *speech acts* approach to grounding, in which utterances are seen as actions affecting the state of grounding of contributed material [Traum and Allen, 1992, Traum, 1994]. In addition to some acts which present new material, there are *acknowledgment* acts which signal that the current speaker has understood previous material presented by the other speaker, *repairs* and *requests for repair*.

While there are short-comings of previous accounts of speech acts such as that in Figure 3, particularly in the way content is added (or is *not* added) to mutual belief, we prefer to keep as much of the previous theory intact as possible. We keep all of the *core* speech acts of previous work, recognizing, however, that they are really *multi-agent* actions, which require input from multiple participants in order to have their full effects such as mutual beliefs. We introduced a level of dialogue structure called *discourse units* (DUs), at which these core speech acts are completed. These DUs are built up by single-utterance *grounding acts*. Recognizing the fact that multiple types of action occur in conversation, we extended speech act theory to the multi-level *conversation act* theory, summarized in Table 2. As well as the grounding and core speech acts, there are also levels to model turn-taking behavior and higher order coherence of dialogue. A similar model of *meta-locutionary acts* was previously introduced by Novick [1988].

Level	Act Type	Sample Acts
`<UU`	**Turn-taking**	`take-turn` `keep-turn`
`UU`	**Grounding**	`Initiate Repair` `Ack Continue`
`DU`	**Core Speech Acts**	`Inform YNQ` `Accept Request`
`>DU`	**Argumentation**	`Elaborate Q&A`

Table 2: Conversation Act Types

The grounding acts were used as the basis for a computational account of the grounding process [Traum, 1994], in which a finite automaton was used to track the state of a DU, given a sequence of grounding acts in conversation. This model could also be used to predict possible subsequent acts as well as determine which act(s) must be performed in order to have a grounded DU (which would thus realize the effects of the constituent core speech acts).

Also presented in [Traum, 1994] was an account of grounding in terms of the perception and mental attitudes of the conversational participants. This included beliefs, intentions, and obligations, as well as nested attitudes. The reasons for engaging in particular actions were presented as relations between elements of these mental states, along with a description of the effects of these actions on the mental states of conversing agents.

4 The TRAINS-93 Dialogue Manager

The aspects of conversational agency described in the previous section have been put together to form the core of a theory of dialogue management, of which the TRAINS-93 dialogue manager is an example implementation. The TRAINS-93 System (described in detail in [Allen *et al.*, 1995]) is a large integrated natural language conversation and plan reasoning system. The dialogue manager is responsible for maintaining the flow of conversation and making sure that the conversational goals are met. For this system, the main goal is that a shared plan which meets the user's domain goals is constructed and executed in the domain.

The dialogue manager must track the state of the dialogue, determine effects of observed conversation acts, generate utterances, and send commands to the domain task reasoners when appropriate. Each utterance will generally contain acts (or partial acts) at each of the conversation act levels.

As with the BDI agency model in Figure 2 the dialogue manager agent can be seen as composed of two main parts, a *context* representation, which includes various aspects of mental state, and a deliberation and action mechanism. The actor is *reactive* to the discourse context, looping to perform only small atomic actions in response to its current state, and then updating and reacting again. This architecture makes it fairly flexible to new observations of the domain or conversation from the user as well as to its other processing. In addition, the context is used to help determine which acts have been performed, and is updated with the results of those acts.

4.1 Discourse Context

The discourse context of the dialogue manager contains representations of both the mental attitudes described previously, as well as a model of the current conversation. Several nestings of domain belief must be tracked in order to fulfill the conversational purposes in a task-oriented dialogue. Private beliefs must be maintained in order to do accurate task reasoning. In addition, the system must maintain beliefs about the user's beliefs and about mutual beliefs both to interpret user utterances and formulate its own expressions coherently. We represent these belief nestings as distinct but related belief modalities.

In addition to actual beliefs, the system must also track **proposals**. Although there will be a natural connection between these and actual beliefs, there is good reason to keep them distinct during intermediate phases in the conversation. One use is for modeling insincere utterances. Even if the interpretation of an utterance includes a claim that a certain state of affairs holds, there might be good evidence to suppose that the actual beliefs of a speaker are otherwise. Having separate modalities for proposals allows representation of any discrepancy. In addition, this will allow a more explicit representation of the method by which beliefs change through conversation – the immediate effect of a representational utterance will change only the proposal modality, and it will require an additional (mental) action to actually change the belief. This proposal modality is also useful for representing suggested courses of events that have not yet been firmly decided upon.

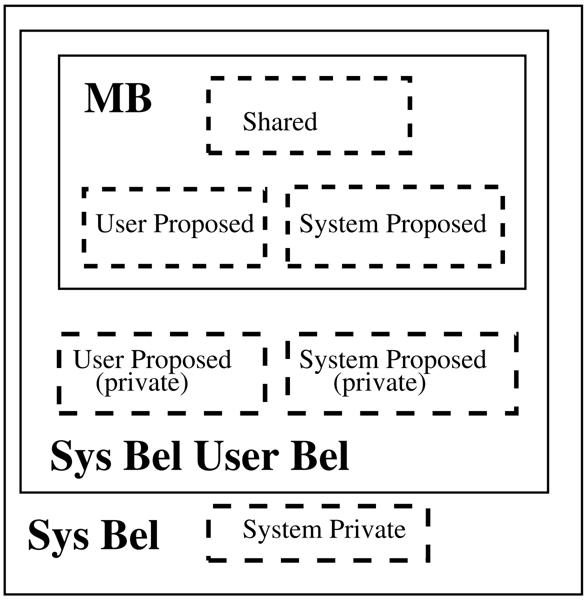

Figure 6: Belief and Proposal Contexts

Figure 6 shows the relationships between the belief and proposal modalities. The belief modalities are shown with solid boxes, with containment representing the relationships between these modalities. Some of these beliefs will concern proposals. Part of the mutual beliefs will be mutual beliefs about what the user and system have proposed. Proposals that have been accepted are also shown here as *shared*. For representative proposals (about the actual state of affairs), this *shared* modality will be undistinguished from general mutual beliefs about the world, but for directive proposals (about *actions* to be taken by the conversational participants), the shared context will include an intentional component (to perform these actions and keep them achievable) as well as a mutual belief about future eventualities. The proposal modalities in the **Sys Bel User Bel** context represent proposals that have been initiated but not yet grounded. Also, in the **Sys Bel** context, the *System Private* modality represents proposals that the system has decided to make but has not yet initiated.

For the TRAINS system, many of the utterances will include suggestions of actions, goals, and constraints to add to the current domain plan. From the point of view of the dialogue manager, *domain plans* are abstract entities which contain a number of parts. These include: the goals of the plan, the actions which are to be performed in executing the plan, objects used in the plan, and constraints on the execution of the plan. The detailed structure of TRAINS-93 domain plans and the view of plans as *arguments* are described in detail in [Ferguson, 1995].

Views of domain plans are represented as follows. The *shared* modality will include aspects of plans assumed to be shared plans – jointly intended by the system and the user. The mutually believed proposal modalities include plans proposed by one or the other party but which have not yet been accepted. The proposal modalities in **Sys Bel User Bel** represent proposals which have not yet even been acknowledged, and finally the *System Private* modality will contain plans that the system's back-end plan reasoner has constructed but which have not yet been communicated to the user. This framework allows for the representation of both the incremental construction of plans as well as conflicting proposals of what a plan should be, when the plans in different contexts have contradictory components.

The system maintains a set of high-level **Discourse Goals** representing what it hopes to achieve with the conversation. For the TRAINS domain, this is represented as a script, specifying the goals of different phases in the conversation. The TRAINS-93 script includes phases for identifying a domain goal, developing a shared plan to meet this goal, and executing the plan in the TRAINS domain. The system also maintains structures of *obligations* which have arisen according to the rules in Table 1. Also, a set of intended conversation acts is maintained, which the system will try to perform (by sending to the NL Generator, for output to the user), when it gets an opportunity. Because the system attempts to adhere to conventional interactional patterns, it does not always perform these right away, and might not get a chance to perform some of them. For example a suggestion may be preempted by a similar or conflicting suggestion by the user. Also, an answer to a question may become irrelevant if the user retracts the question.

4.1.1 Discourse Model

In addition to the mental attitudes described above, there are several aspects of the discourse interactional state that must be tracked. These represent the model of the conversation itself rather than particular mental attitudes of the participants. They are used as a resource to help generate local expectations for the flow of the conversation, as well as serving as an important tool for interpreting subsequent utterances.

Two contextual notions are useful for tracking turn-taking in casual task-oriented conversations such as those in the TRAINS domain. These are the **turn** and **local initiative**. Each of these may be said to be held by one (or none) of the participants at any given time in the conversation. The notion of who has the turn is important in deciding whether to wait for the other agent to speak, or whether to formulate an utterance. It will also shape the type of utterance that will be made, e.g. whether to use some kind of interrupting form or not.

Local initiative[3] can be glossed as providing the answer to the question of who has the most recent discourse obligation – who is expected to speak next according to the default plans for simplest satisfaction of conversational goals. In the TRAINS domain, the initiative is shared, with different participants holding it at different points in the conversation. In the initial phase, the user has the initiative while the task is conveyed. In the main part of the conversation – the construction of the plan – the initiative can lie with either party, though it generally remains with the user. In the final phase, verifying successful completion of the problem, the initiative belongs with the system.

In order to track grounding, the system will maintain a bounded stack of accessible discourse units (DUs). For each DU, the system notes the initiator and the *state* of the DU, in addition, to a representation of the constituent (partial) core speech and argumentation acts and their effects. Multiple DUs are modeled using a bounded stack structure. The structure is stack-like since, generally, new utterances will affect the most recently started DU. The stack structure is bounded to capture the constraint that DUs have limited accessibility. After enough intervening material, the older DUs are no longer directly accessible (although their content can always be reintroduced in new DUs). Uncompleted DUs which "fall off" the back of the stack are treated as if they had been cancelled – their contents are not considered grounded.

Discourse Segmentation information [Grosz and Sidner, 1986] is kept for a variety of purposes in linguistic interpretation and generation, including the ability to determine the possible referents for a referring expression and the intentional relations between utterances. The currently open segment structure will guide how certain utterances will be interpreted. In addition to general segmentation information, a structure of conversationally accessible domain objects is maintained. For the TRAINS system, this will include a set of accessible domain plans from a given segment, as well as recency pointers to parts of plans from utterances comprising the segment.

[3] Roughly the same notion as *Control* in [Walker and Whittaker, 1990], although we use a finer grained notion of utterance types.

4.2 Conversational Updates

The main way that the conversational state is updated is through the performance of conversation acts. These are briefly summarized here. Other changes to the mental state result from the deliberation process described in the next section. *Turn-taking acts* will generally only affect the turn. Grounding acts will primarily affect the grounding model – they will update the state of the DU they are a part of. Each grounding act performed as part of an utterance event will also have associated with it a (possibly empty) list of core speech acts and argumentation acts which are attempted in the utterance, which are also added to the local memory of that DU `Initiate` and `ack` acts have additional consequences. An `initiate` will add a new DU to the DU stack, often removing an old DU from the bottom of the DU stack. An `ack` will make the contents of that DU mutually believed, thus causing a transfer of contents from the SBUB modality to the MB modality, perhaps causing additional effects such as new discourse obligations or intentions.

Core speech acts have a variety of effects, such as adding new mutual beliefs or obligations, as outlined above. More details are presented in [Allen *et al.*, 1995, Traum, 1994]. Argumentation acts have an affect on the discourse coherence. This will result both in adding implicated information to the information conveyed by core speech acts, as well as affecting the discourse segmentation structure.

4.3 The Discourse Actor

In designing an agent to control the behavior of the dialogue manager, we choose a *reactive* approach in which the system will deliberate as little as possible until it can act in one way or another. It will not form complete or long-range plans about the discourse, but will proceed one step at a time, deducing and performing the next appropriate action, according to conversational conventions and its high-level goals. The TRAINS-93 actor uses the following prioritized sources for the deliberations:

1. Discourse Obligations from Table 1
2. Weak Obligation: Don't interrupt user's turn
3. Intended Speech Acts
4. Weak Obligation: Grounding
5. Discourse Goals: Proposal Negotiation
6. High-level Discourse Goals

The actor's first priority is fulfilling obligations. If there are any, then the actor will do what it thinks best to meet those obligations. If there is an obligation to address a request, the actor will evaluate whether the request is reasonable, and if so, accept it, otherwise reject it, or, if it does not have sufficient information to decide, attempt to clarify the parameters. In any case, part of meeting the obligation will be to form an intention to tell the user of the decision (e.g., the acceptance, rejection, or clarification). When this intention is acted upon and the utterance produced, the obligation will be discharged. Other obligation types are to repair an uninterpretable utterance or one in which the presuppositions are violated, or to answer a question. In question answering, the actor will query its beliefs and will answer depending on the result, which might be that the system does not know the answer.

In most cases, the actor will merely form the intention to produce the appropriate utterance, waiting for a chance, according to turn-taking conventions to actually generate the utterance. In certain cases, though, such as a repair, the system will actually try to take control of the turn and produce an utterance immediately. For motivations other than obligations, the system adopts a fairly "relaxed" conversational style; it does not try to take the turn until given it by the user unless the user pauses long enough that the conversation starts to lag. When the system does not have the turn (priority 2), the conversational state will still be updated, but the actor will not try to deliberate or act.

When the system does have the turn, the actor first (after checking obligations) examines its intended conversation acts (priority 3). If there are any, it calls the NL generator to produce an utterance.[4] System utterances are also reinterpreted (as indicated in Figure 1) and the conversational state updated accordingly. This might, of course, end up in releasing the turn. It might not be convenient to generate all the intended acts in one utterance, in which case some intended acts may be left for the future consideration. When the turn changes, only those intended speech acts that are part of the same argumentation acts as those which are uttered will be maintained as intentions – others will revert back to whatever caused the intention to be formed, although subsequent deliberation might cause the intentions to be re-adopted.

If there are no intended conversation acts, the next thing the actor considers is the grounding situation (priority 4). The actor will try to make it mutually believed (or *grounded*) whether particular speech acts have been performed. This will involve acknowledging or repairing user utterances, as well as repairing and requesting acknowledgment of the system's own utterances. Generally, grounding is considered less urgent than acting based on communicative intentions, although some grounding acts will be performed on the basis of obligations which arise while interpreting prior utterances.

If all accessible utterances are grounded, the actor then considers the negotiation of domain beliefs and intentions, represented in Figure 6 (priority 5). The actor will try to work towards a shared domain plan, adding intentions to perform the appropriate speech acts, including accepting, rejecting, or requesting retraction of user proposals, requesting acceptance of or retracting system proposals, and initiating new system proposals or counterproposals. The actor will first look for User proposals which are not shared. If any of these are found, it will add an intention to accept the proposal, unless the proposal is deficient in some way (e.g., it will not help towards the goal or the system has already come up with a better alternative). In this latter case, the system will reject the user's proposal and present or argue for its own proposal. Next, the actor will look to see if any of its own proposals have not been accepted, requesting the user to accept them if they have been simply acknowledged, or retracting or reformulating them if they have already been rejected. Finally, the actor will check its private plans for any parts of the plan which have not yet been proposed. If it finds any here, it will adopt an intention to make a suggestion to the user.

If none of the more local conversational structure constraints described above require attention, then the actor will concern itself with its actual high-level goals (priority 6). For the TRAINS system, this will include making calls to the domain plan reasoner and domain executor, which will often return material to update the system's private view of the plan and initiate its own new proposals. It is also at this point that the actor will take *control* of the conversation, pursuing its own objectives rather than responding to those of the user.

Finally, if the system has no unmet goals that it can work towards achieving, it will hand the turn back to the user or try to end the conversation if it believes the user's goals have been met as well.

4.4 EXAMPLE

The following example gives a small taste of how the dialogue manager uses this representation of context and priorities to engage in dialogue. More extended examples are presented in [Traum and Allen, 1994,

[4] If the only intention is to acknowledge, the actor will postpone the generation until it checks whether there is any other content, such as an acceptance or answer, that could be expressed in the same utterance.

Traum, 1994]. The example starts with a declarative utterance by the User:

U: "There are oranges at Corning."

At the core speech act level, this is interpreted as performing both an **inform** (about the location of oranges), and a **suggestion** that the oranges be used in the current plan. At the grounding level, this is seen as the initiation of a DU. It is also seen as keeping the turn. This has the following effects on the context - first (at priority level (4), there is an unacknowledged DU, which will require grounding. More prominently, however, the user still has the turn, so the system will just wait for the next utterance.

U: "Is a boxcar there?"

This is interpreted as asking a yes-no question, continuing the current DU, and releasing the turn. Now there is an additional core speech act in the ungrounded DU, and the system has the turn. The chosen action is now, at priority 4, to add the intention to acknowledge the content in this DU (a new item at priority 3). Forming this intention also causes the system to update its mental state with the effects of this content. In this case, the inform and suggestion will lead to items in the user-proposed context in Figure 6, at priority level 5. The YNQ leads to an obligation to answer the question, which is at priority level 1. Since the obligation is of highest priority, the system acts upon this by querying its beliefs to see if a boxcar is at Corning. This check returns negatively, which leads the system to intend to inform the user of this fact. Now, the the highest priority are the intended speech acts. These are passed to the NL generator, and a combined, acknowledgment/answer is provided with:

S: "No there isn't"

This simple example displays some of the flexibility of the reactive agency model. Given different responses or a different initial mental state, many variants of this simple dialogue could have been produced using the same rules. Most of the flexibility of plan-based approaches is maintained, while the obligation model presents a much more direct account of question answering, without any need for reasoning about or adopting the desires of the user.

5 Conclusions

The model presented here allows naturally for mixed-initiative conversation and varying levels of cooperativity. Following the initiative of the other can be seen as an *obligation-driven* process, while leading the conversation will be *goal-driven*. Representing both obligations and goals explicitly allows the system to naturally shift from one mode to the other. In a strongly cooperative domain, such as TRAINS, the system can subordinate working on its own goals to locally working on concerns of the user, without necessarily having to have any shared discourse plan. In less cooperative situations, the same architecture will allow a system to still adhere to the conversational conventions, but respond in different ways, perhaps rejecting proposals and refusing to answer questions. This architecture can handle production and recognition of acknowledgment and repair in a natural and fairly comprehensive manner, and can prompt for them when they are required. While the strict prioritization of aspects of mental state used here is too severe in general (e.g., some goals should take priority over some obligations), the framework still has a great deal of flexibility, seizing the initiative when it lags, and relinquishing it when the user presents her own goals.

Viewing dialogue systems, and in particular dialogue managers as *agents* naturally lends itself to such flexible interaction. Modeling the system's action and computation as deliberation over aspects of mental state has several advantages. First, it allows a uniform treatment of dialogue partners, be they humans or other machines. The system can reason about itself in the same way it reasons about human users. Since much of the semantics and pragmatics of natural language communication makes reference (both explicitly and implicitly) to the mental state of communicating agents, having such an explicit model makes reasoning about natural language more straightforward.

Conversation and social interaction puts extra demands on a model of agency, beyond that of simple perception. The inclusion of social attitudes, such as obligation and mutual belief, however, can lead to natural and powerful extensions to a BDI model. As demonstrated by the TRAINS-93 system, the combination of a rich notion of mental state and simple, reactive deliberation mechanisms can yield flexible and dynamic conversational behavior.

References

[Allen *et al.*, 1995] James. F. Allen, L. K. Schubert, G. Ferguson, P. Heeman, C. H. Hwang, T. Kato, M. Light, N. Martin, B. Miller, M. Poesio, and D. R. Traum. The TRAINS project: a case study in building a conversational planning agent. *Journal of Experimental and Theoretical Artificial Intelligence*, 7:7–48, 1995.

[Allen, 1983] James [F.] Allen. Recognizing intentions from natural language utterances. In Michael

Brady and Robert C. Berwick, editors, *Computational Models of Discourse*. MIT Press, 1983.

[Austin, 1962] J. A. Austin. *How to Do Things with Words*. Harvard University Press, 1962.

[Bratman *et al.*, 1988] Michael E. Bratman, David J. Israel, and Martha E. Pollack. Plans and resource-bounded practical reasoning. *Computational Intelligence*, 4:349–355, 1988.

[Bratman, 1987] Michael E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, 1987.

[Clark and Schaefer, 1989] Herbert H. Clark and Edward F. Schaefer. Contributing to discourse. *Cognitive Science*, 13:259–294, 1989. Also appears as Chapter 5 in [Clark, 1992].

[Clark, 1992] Herbert H. Clark. *Arenas of Language Use*. University of Chicago Press, 1992.

[Cohen and Perrault, 1979] Phillip R. Cohen and C. R. Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3(3):177–212, 1979.

[Cohen, 1978] Phillip R. Cohen. *On Knowing What to Say: Planning Speech Acts*. PhD thesis, University of Toronto, 1978. Reproduced as TR 118 Department of Computer Science, University of Toronto.

[Ferguson, 1995] George Ferguson. *Knowledge Representation and Reasoning for Mixed-Initiative Planning*. PhD thesis, University of Rochester, 1995. Also available as TR 562, Department of Computer Science, University of Rochester.

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

[McCarthy, 1990] John McCarthy. Ascribing mental qualities to machines. In *Formalizing Common Sense*. Ablex, 1990. Originally published in 1979.

[Novick, 1988] David Novick. *Control of Mixed-Initiative Discourse Through Meta-Locutionary Acts: A Computational Model*. PhD thesis, University of Oregon, 1988. also available as U. Oregon Computer and Information Science Tech Report CIS-TR-88-18.

[Perrault and Allen, 1980] C. Raymond Perrault and James F. Allen. A plan-based analysis of indirect speech acts. *American Journal of Computational Linguistics*, 6(3-4):167–82, 1980.

[Pollack and Ringuette, 1990] Martha E. Pollack and M. Ringuette. Introducing the tileworld: Experimentally evaluating agent architectures. In *Proceedings AAAI-90*, pages 183–189, 1990.

[Shoham and Tennenholtz, 1992] Yoav Shoham and Moshe Tennenholtz. On the synthesis of useful social laws for artificial agent societies. In *Proceedings AAAI-92*, pages 276–281, 1992.

[Traum and Allen, 1992] David R. Traum and James F. Allen. A speech acts approach to grounding in conversation. In *Proceedings 2nd International Conference on Spoken Language Processing (ICSLP-92)*, pages 137–40, October 1992.

[Traum and Allen, 1994] David R. Traum and James F. Allen. Discourse obligations in dialogue processing. In *Proceedings of the 32th Annual Meeting of the Association for Computational Linguistics*, pages 1–8, 1994.

[Traum and Heeman, 1996] David R. Traum and Peter Heeman. Utterance units and grounding in spoken dialogue. to be presented at *4th International Conference on Spoken Language Processing (ICSLP-96)*, October, 1996.

[Traum, 1994] David R. Traum. *A Computational Theory of Grounding in Natural Language Conversation*. PhD thesis, Department of Computer Science, University of Rochester, 1994. Also available as TR 545, Department of Computer Science, University of Rochester.

[von Wright, 1951] G. H. von Wright. Deontic logic. *Mind*, 60:1–15, 1951.

[Walker and Whittaker, 1990] Marilyn A. Walker and Steve Whittaker. Mixed initiative in dialogue: An investigation into discourse segmentation. In *Proceedings ACL-90*, pages 70–78, 1990.

[Wooldridge and Jennings, 1995] M. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *Knowledge Engineering Review*, 10(2), 1995.