

A Conversation Acts Model for Generating Spoken Dialogue Contributions

AMANDA J. STENT*¹

¹*Department of Computer Science, SUNY at Stony Brook, NY, USA
stent@cs.sunysb.edu*

*We gratefully acknowledge James Allen, George Ferguson and the other members of the TRIPS group at the University of Rochester, who are always generous with help and advice. We would also like to thank the anonymous reviewers for their helpful comments. This work was funded by DARPA research grant no. F30602-98-2-0133, ONR research grant no. N00014-95-1-1088, and NSF grant no. EIA-0080124.

Contact information

Department of Computer Science
SUNY Stony Brook
Stony Brook, NY 11794-4400
USA
stent@cs.sunysb.edu

Abstract

In this paper, we describe a generation system for spoken dialogue that not only produces coherent, informative and responsive dialogue contributions, but also explicitly models human styles of interaction. This generation system is based on conversation acts theory. It has been implemented in the TRIPS spoken dialogue system, and includes components that plan content, perform surface generation for different modalities, and coordinate output production. We discuss our implementation, and describe an evaluation of the generation output.

1. Introduction

This paper addresses the question of how to enable task-oriented dialogue systems to plan more “natural” dialogue contributions; dialogue behavior that, at least in particular types of situations, seems to the human user of a dialogue system as though it could come from another human. Much previous research has addressed the question of generating coherent, informative and responsive dialogue contributions (e.g. [Chu-Carroll and Carberry, 1998, Lochbaum, 1998, Power, 1979]); this work is concerned with explicitly modeling human styles of interaction.

Our work on spoken dialogue generation is performed in the context of the TRIPS system, a task-oriented spoken-dialogue system at the University of Rochester. TRIPS uses rich models of intention recognition and context, and can handle fairly complex task-oriented dialogue in a variety of domains, including scheduling, planning and advice-giving domains [Allen et al., 2001a]. TRIPS represents years of work by many researchers, and offers a unique opportunity to explore generation for dialogue without having to solve the many other problems related to dialogue processing (such as speech recognition, parsing and intention recognition). Furthermore, as the domains and tasks handled by the system have become more complex, the need for extensive, fast, and flexible generation has become ever more apparent [Stent, 1999].

In one of the TRIPS domains, the Monroe domain, users interact with the system to handle health and public safety emergencies in Monroe county, NY. We have collected a corpus of human-human dialogues in the Monroe domain. The tasks used for this collection were difficult enough that they could not typically be solved by one person working alone in under ten minutes [Stent, 2000a]. Because of the complexity of the domain and tasks, these dialogues exhibit a high degree of mixed initiative behavior, with both speakers contributing to the solution of the task and to the maintenance of the collaboration. We used this corpus to study the structure of dialogue contributions in human-human task-oriented dialogues, and to develop our model of generation for spoken dialogue.

Extracts from two of the Monroe dialogues are shown in figures 1 and 2[†]. In the first dialogue, the participants have just finished making a plan to deal with broken water mains and downed power lines resulting from a snow storm. They had to work under quite severe constraints: some of the repair locations were roads that had not yet been plowed, and some of the repair locations had to be dealt with quickly because there were disabled people relying on power at those locations. This extract comes from the end of the dialogue; one participant is summarizing the plan for the other.

In the second dialogue extract, the participants are in the middle of forming a

[†]The dialogue extracts in this paper are segmented into turns and utterances. Generally, the utterances are numbered in the transcript. In our examples, overlapping speech is indicated with ‘+’ signs and silences or pauses are indicated with <SIL>. Sometimes, the participants sneeze, laugh, or smack their lips; these sounds are given in brackets.

utt272 u yeah only one digger we're <SIL> sort of limited okay
 utt273 well i'll so i'll go over the <SIL> the plan as we
 have it <SIL> which <SIL> is sort of unfortunate
 utt274 + so we're sen- + <SIL> we're sending one electric crew
 <SIL> <lipsmack> <SIL> from <SIL> r g and e
 <SIL> to <SIL> east and three ninety <SIL> via
 <SIL> east and five ninety <SIL> via <SIL>
 three ninety <SIL> and five ninety
 utt275 s + okay +
 utt276 u + and they'll + be they'll get there they'll finish all
 the work they'll be done within an hour and a half
 <SIL> so that'll be fine for those peo- <SIL>
 those sick <SIL> people
 utt277 s + yes +
 utt278 u <lipsmack> <SIL> we'll send the <SIL> another
 electric crew <SIL> to <SIL> two fifty two
 <SIL> and <SIL> the river <sniff>
 utt279 <lipsmack> via three ninety and three eighty three
 utt280 that should also be pretty quick that'll take an hour
 <SIL> that takes care of all the sick <SIL> people
 utt281 s um <SIL> one correction is i think <SIL> that
 <SIL> the uh <SIL> you have the times reversed
 <SIL> oh no <SIL> you're right

Figure 1: First Monroe dialogue extract

plan to deal with disturbances, injuries and a fire around downtown Rochester. They are dispatching police units, medical crews and fire truck; the police units must be dispatched before any of the other emergency vehicles.

The complex coordination behavior exhibited by both participants in these dialogues is evident even in these short extracts. Although there is quite a bit of overlapping speech the speakers are not talking past one another. Each participant contributes to the shared information and acknowledges the other's contributions (e.g. utterances 274 and 275 in the first extract, utterances 78 and 79 in the second). Also, we can see that quite complex discourse structure is necessary to communicate the relationships between tasks, actions and solutions, and time and space constraints. Finally, there are interesting initiative shifts in both these extracts (utterances 280 and 281 in the first extract, and utterances 85–89 in the second).

Traum and Hinkelman, in their work on spoken dialogue structure, identify a hierarchy of dialogue behaviors or *conversation acts*, including *turn-taking*, *grounding*, *speech*, and *argumentation* acts [Traum and Hinkelman, 1992]. Examples of each act type are given in table 1. *Turn-taking acts* are acts that

utt78 u and then <breath> sent <SIL> an ambulance to
 <SIL> <breath> from rochester <SIL> general
 utt79 s right
 utt80 right i presume there that we'll have the rioting under
 control at that <SIL> point
 utt81 <SIL> um but we're still going to have at least um
 <SIL> two walking wounded there to be + dealt with +
 utt82 u + right +
 utt83 s and we're still going to have three walking wounded at
 east main street to be dealt with
 utt84 u that's right
 utt85 s um
 utt86 since we don't have any other medical <SIL> people at
 the time i suggest we go ahead and just dispatch the rest
 of our <SIL> um <SIL> police units <SIL> um
 utt87 + to the + other <SIL> three locations + yes +
 utt88 u + um +
 utt89 + what about <SIL> + the uh <lipsmack> <SIL> the
 fire hydrant
 utt90 s <click> <SIL> yeah well we need to get <SIL> police in
 there though we can get the water + crew + right in after them
 utt91 u + okay +

Figure 2: Second Monroe dialogue extract

regulate which speaker may speak at any point in the conversation [Sacks et al., 1974]. *Grounding acts* are acts that establish that both participants are “up to date”, that is, that each participant knows what is known and believed by both at any point in the conversation [Clark and Schaefer, 1989]. *Core speech acts* correspond to the traditional notion of a speech act: the action performed in speaking [Searle, 1969]. Finally, *argumentation acts* are sets of speech acts related in some way, either functionally (e.g. by means of rhetorical relations [Mann and Thompson, 1987] or adjacency pairs [Sacks et al., 1974]) or structurally (e.g. a list of items).

In conversation acts theory, the performance of acts at lower levels is necessary to the performance of acts at higher levels: “the more conventional and intentional level acts are conditionally generated by the performance of appropriate acts at lower levels, given the proper context” [Poesio and Traum, 1997]. For example, in order for an *inform* act to be performed, one speaker must *initiate* the *inform* (perform the core speech act *inform*) and the other must ground it by *accepting* it. Both the initiation and the response require the speaker to take and then release the turn. Two observations follow from this aspect of conversation acts theory:

- Because performance of an act at a higher level (e.g. a speech act) entails performance of one or more acts at each lower level (e.g. a turn-taking act), the acts at the lower levels do not necessarily need to be explicitly realized using language in every case. We will refer to this phenomenon as *subsumption*; the performance of a lower-level act can be subsumed in (performed in the process of) the production of a higher-level act.
- If an act at a lower level is performed on its own (resulting in the production of language for that act only), it should typically precede in the output the performance of higher-level acts that could have subsumed it. It cannot, logically speaking, be performed after the higher-level act; if it is not performed before the higher-level act, because it will be performed in the process of performing the higher-level act. For example, a *take-turn* act can be performed before performing an *assert*, or can be subsumed in the *assert*. We will refer to this constraint as the *ordering constraint* on conversation act-based generation.

Conversation acts theory has never been used for spoken dialogue generation. However, it provides a nice account of discourse structure, in which the relationships between different dialogue behaviors are clearly defined and collaboration-maintaining behaviors are granted equal status with intentional and rhetorical structures.

Act type	Sample acts
turn-taking	take-turn, keep-turn, release-turn, assign-turn
grounding	initiate, continue, acknowledge, repair, cancel
core speech acts	inform, yes/no question, suggest, request, accept, reject
argumentation	elaborate, summarize, clarify, question-answer, convince

Table 1: Conversation act types

In our work on conversation act-based generation for dialogue, we set out to answer two questions:

- If conversation acts theory is used as a discourse analysis tool, to what extent does it provide an adequate accounting of human behavior in actual dialogues?
- Can the key concepts of conversation acts theory be used to build a generation system for spoken dialogue, and is the output of this system comparable to human behavior in actual dialogues?

The second question is the key topic of this paper. However, the first is important because it would be a waste of time to build a generation system based on

a theory that could not adequately account for actual human dialogue behavior. Therefore, in this paper we first briefly discuss our analysis of human-human task-oriented dialogues in the Monroe domain, which demonstrates that conversation acts theory can adequately (if not perfectly) account for many aspects of these dialogues. We then describe a generation system we built for TRIPS that implements the ideas of conversation acts theory. Finally, we describe an evaluation we performed of this system, comparing its output to dialogue behavior produced by humans in the Monroe corpus.

1.1. Related work

The generation task has long been framed as a planning problem. The input to planning is communicative intentions, for example the intention of explaining a process, answering a question, or correcting a misconception. The output may include speech, text, graphics and gestures. The process itself is typically divided into two stages: deciding what to communicate, and deciding how to communicate it.

The first stage of generation is called content planning or strategic generation, and may include the selection of communicative intentions to pursue, the selection of propositional content, and the construction of discourse structure or discourse plans. In most dialogue situations, the entire discourse cannot be planned beforehand; rather, the dialogue participants take turns constructing parts of the dialogue and the overall dialogue structure is a byproduct of their collaboration. We will call these small units of discourse, each of which is a structured set of conversation acts, *contributions*. In this work, we focus on the construction of contributions from sets of communicative intentions; most of the content that can be communicated is given as input. We call this process *contribution planning*.

The second stage of generation is called surface generation or tactical generation, and (for language generation) includes the selection of lexical items and their organization into appropriate syntactic structures. Although it is not the focus of this paper, we do briefly describe our text generation component, since it is part of our model of conversation acts theory.

Most generation systems use a single-layer model of discourse structure and track the rhetorical or intentional structure of the discourse. (Of course other sources of discourse information, such as a discourse context and focus information, may be used by the generation process.)

In the text generation field, some researchers have examined different types of texts in a particular genre, such as explanatory texts or descriptive texts, to try to identify underlying genre-specific structures (e.g. [McKeown, 1985, Cawsey, 1993]). Other researchers have looked at texts across different genres, identifying general relationships that hold between different parts of texts (e.g. [Mann and Thompson, 1987, Marcu, 1997]). Once these structures or relationships have been

identified, they may be used as rules to guide the process of content planning (e.g. [Andre and Rist, 1996, Hovy, 1993, Maybury, 1993, Moore and Paris, 1992]).

The collaborative nature of dialogue imposes practical and theoretical constraints that make the application of many text generation techniques infeasible. Accordingly, most research on dialogue structure starts with an examination of single exchanges between dialogue participants. Most models of dialogue structure define a set of basic exchanges and describe how these can be combined to form the overall intentional or rhetorical structure of a dialogue.

Chu-Carroll and Brown, in their content planning system for task-oriented dialogues, use a propose-evaluate-modify model to describe the intentional structure of task-oriented dialogues. Each user utterance is considered to be a proposal for an addition to the shared plan being built by the dialogue participants. The system evaluates each proposal, deciding whether to accept or reject it, or suggest modifications. If it cannot decide whether to accept, it may initiate an information-sharing sub-dialogue to learn more about the proposal [Chu-Carroll and Carberry, 1995, 1998].

The dialogue system COLLAGEN uses Grosz and Sidner's stack based model of discourse structure, in which the attentional and intentional structures of dialogue are tracked simultaneously [Grosz and Sidner, 1986, Rich and Sidner, 1998]. There are two basic relations between parts of a dialogue in this model: two parts may be nested one within the other (for example, a clarification sub-dialogue will be a nested sub-dialogue), or they may be adjacent to each other (for example, when the topic shifts). Lochbaum, in her dissertation, proposed an algorithm for content planning based on COLLAGEN [Lochbaum, 1998].

POPEL is a content planning system for dialogue that uses a model based on rhetorical relations to decide what content to include in an utterance, how it should be ordered and whether a pointing gesture should be used. POPEL operates in a feedback loop with the surface generation component, providing information about how descriptions should be generated and how items in a sentence should be ordered, among other things [Reithinger, 1992].

In early work on the structure of dialogue, Power describes a system that uses an adjacency pair model of dialogue for interpretation and generation [Power, 1979]. An adjacency pair is a functionally-related pair of utterances by different speakers; examples include *proposal-accept* and *question-answer*. Power's system could interact with other copies of itself in planning dialogues in a simple world. More recent research by Grau and Vilnat is based on a similar idea [Grau and Vilnat, 1997]. Both of these generation systems explicitly model the collaborative nature of dialogue, but they still only track one level of dialogue behavior.

Our approach to generation differs from previous approaches to generation for dialogue in modeling different levels of dialogue behavior simultaneously. By explicitly accounting for turn-taking and grounding behaviors as well as speech acts and the higher-level discourse structure, we aim to improve the naturalness of generated dialogue contributions.

2. Conversation acts analysis of Monroe corpus dialogues

The Monroe corpus [Stent, 2000a] consists of 20 human-human dialogues, altogether comprising approximately 6.6 hours of data and 8200 utterances. We selected 8 of these dialogues at random, ensuring only that we had a range of tasks and speakers. We then annotated these 8 dialogues for turn-taking behavior, grounding and speech acts, and argumentation acts. Because an utterance may contain elements from each of these levels, the different behaviors were annotated separately. For each annotation, there were at least two annotators. Altogether, 7 annotators were involved in this project.

Our reasons for performing these annotations were as follows:

- To test whether conversation acts theory is an adequate explanatory theory for spoken dialogue. That is, we wanted to check whether conversation acts theory can account in an unforced way for every utterance in human-human dialogues; whether the different levels of dialogue behavior are necessary; and whether they interact in the way the theory predicts.
- To identify conversational patterns. We were specifically interested in how often turn-taking and grounding acts are not subsumed in higher-level acts; in how different conversation acts are combined in a single turn or small sequence of turns; and in the combinations of conversation acts that may be realized in a single utterance.
- To identify words or phrases that signal particular conversation acts, particularly words or phrases commonly used to perform turn-taking and grounding acts.

2.1. Turn-taking acts

We labeled turn-taking behavior using a simple manual of our own creation [Stent, 2001]. Essentially, this manual defines:

- the tags to be used to label *take-turn*, *release-turn*, *assign-turn*, *steal-turn* and *keep-turn* acts
- the tags to be used to label words or phrases in an utterance
- how to label backchannels, which do not change the turn holder
- how to label overlap when it is not the result of an interruption that seemed to perform strictly turn-taking functions

Because the participants in these dialogues could not see each other, we did not have to worry about most non-verbal turn-taking behavior such as gaze and gesture. The annotators listened to the dialogues, so they were able to pick up on intonational cues.

Turn-taking is the dialogue behavior that fits least well into conversation acts theory. To think of turn-taking as an act-based behavior at all requires taking a view of this phenomenon very different from those described in most of the

turn-taking literature (e.g. [O’Connell et al., 1990, Oreström, 1983, Sacks et al., 1974]). First, there is the awkward fact that an utterance (or even a single word) can perform several turn-taking functions. For example, an acknowledgment can perform both a *take-turn* and a *release-turn* act. Also, it can be difficult to find situations where there is no doubt that the speaker is only performing some turn-taking act. This task is not made easier by the fact that conversation acts theory does not address primarily non-verbal communication, and much of turn-taking behavior is determined by gaze, gesture and intonation [Cassell et al., 1998, Novick et al., 1996, Oreström, 1983]. Finally, turn-taking is the only dialogue behavior included in conversation acts theory in which time plays a very important role [Bull, 1998].

Discounting disagreements arising from resegmentation of utterances, the Kappa scores for our annotation of turn-taking behavior are .77 for turn-taking functions, namely *take-turn*, *keep-turn*, and *steal-turn*; and .78 for the turn-releasing functions *release-turn* and *assign-turn* (these scores are both significant at $p < .1$). Almost all the disagreement came from three sources: disagreement about whether an utterance was a backchannel or an acknowledgment; disagreement about whether an instance of overlap was due to an interruption; and disagreement about whether an utterance that “trailed off” was a prompt (*assign-turn*) or not (*release-turn*). Each of these types of disagreement is difficult to resolve, since a determination depends on subtle intonational cues. These Kappa scores for this annotation are not quite high enough to judge the annotation reliable, and the reasons for the disagreements illustrate some of the difficulties with an act-based accounting of turn-taking behavior.

Because turn-taking fits so awkwardly into conversation acts theory, we might ask if it should be included at all. One way to test this is to see if there are any utterances or parts of utterances in the spoken dialogues we dealt with that cannot be explained by any means other than as turn-taking behavior. There are indeed some behaviors for which the most plausible explanation seems to require a model of turn-taking behavior:

- prompts – Such as “right?”, “okay?”, “got it?”.
- verbal fillers – For example, “um”, “uh” (although these can also express uncertainty).
- some backchannels – Most backchannels can be analyzed as expressing understanding or at least hearing, but some are redundant according to that explanation. For example, sometimes a speaker produces two backchannels in a row, with a pause after each to allow someone else to speak. In these cases the speaker seems to be expressing unwillingness to take the turn.
- certain repetitions – When speakers overlap at the start of a turn, typically one abandons their utterance. The other will often repeat the first part of theirs [Schegloff, 1987].

When we labeled the Monroe dialogues for speech and grounding acts using DAMSL [Allen and Core, 1997], most of the utterances labeled with the *Other-forward-function* tag fell into one of these categories. This means that even a dialogue system that does not model prosody, to be natural, must still model turn-taking to the extent of being able to produce certain types of utterances.

Cue	Turn-taking acts signaled
um	keep-turn, take-turn, release-turn
<lipsmack>	take-turn, keep-turn
<click>	take-turn, keep-turn
well	keep-turn, take-turn
oh	keep-turn, take-turn
uh	keep-turn, take-turn
so	keep-turn, take-turn
just a second	keep-turn, take-turn
okay	take-turn, keep-turn
isn't that so	assign-turn
say that again	assign-turn
you know	assign-turn
I'm ready	release-turn

Table 2: Language used to explicitly perform turn-taking acts

Table 2 shows a sample of the words and phrases annotators labeled as explicitly performing one turn-taking act or another. For each one, the turn-taking acts most frequently performed by that word or phrase are listed. This list is not exhaustive, but is intended merely to provide examples. An interesting observation is that sometimes the same cue can have very different functions, with most of the difference coming from intonation or placement in the turn (consider the uses of “um”, for example). We used the less ambiguous verbal cues in our templates for surface generation of various turn-taking acts.

2.2. Grounding and speech acts

The grounding and speech act levels of dialogue are well understood and fit well in a larger act-based framework such as conversation acts theory. We labeled the 8 Monroe dialogues for grounding and speech acts using DAMSL [Allen and Core, 1997]. DAMSL tags fall into 13 dimensions that are in three layers: forward looking functions, backward looking functions, and information level. Tag dimensions that involve forward-looking functions indicate the type of speech act the utterance is performing; examples include *assert*, *info-request*, *action-directive* and *commit*. Tag dimensions that involve backward-looking functions indicate how this utterance relates to previous utterances and include answers to questions, indications of degree of understanding, and indications of (dis-)agreement. The *Understanding* tag marks some grounding functions; we did not distinguish between the *initiate* and *continue* grounding acts in this annotation.

Information-level tags indicate the dialogue level of the contents of the utterance (task, task-management or communication-management). Each tag dimension is marked separately from the others.

Forward-looking functions		
Tag dimension	Tags	Kappa
Conventional	Opening, Closing	.882
Explicit-performative	Yes, No	.452
Exclamation	Yes, No	.699
Influence-on-listener	Action-directive, Open-option	.883
Influence-on-speaker	Offer, Commit	.831
Info-request	Yes, No	.902
Other-forward-function	Yes, No	.881
Statement	Assert, Reassert, Other	.876
Backward-looking functions		
Tag dimension	Tags	Kappa
Agreement	Accept, Accept-part, Hold, Maybe, Reject-part, Reject	.888
Answer	Yes, No	.856
Understanding	Signal-non-understanding, SU-Acknowledge, SU-Repeat-rephrase, SU-Completion, Correct-misspeaking	.914

Table 3: Kappa scores for forward- and backward-looking functions for annotation of eight Monroe county dialogues

Table 3 shows the DAMSL tag dimensions and tags for forward- and backward-looking functions. The Kappa scores for annotator agreement for each tag on the Monroe corpus are also given (all are significant at $p < .01$ except for the *Performative* dimension, which is significant at $p < .1$). The low reliability for the *Explicit-performative* and *Exclamation* tag dimensions is due to having very little data for these two tags. The Kappa scores for the other tag dimensions indicate that those annotations are reliable.

Table 4 shows the relative frequencies of all tags in the Monroe corpus, and where known, in the Switchboard and TRAINS corpora [Jurafsky et al., 1998][‡]. In DAMSL, a tag is typically not marked unless the corresponding act is explicitly performed – so, for example, the *SU-Acknowledge* tag is not marked unless there is an explicit acknowledgment. These data indicate that grounding acts are performed explicitly more often than conversation acts theory would perhaps predict (about 30% of the utterances have an *Understanding* dimension tag); but that the subsumption of grounding in higher-level acts is still very prevalent. In particular, the *initiate* and *continue* grounding acts are never per-

[‡]It should be noted that the TRAINS results are based largely on annotations done using a previous version of DAMSL, and the Switchboard results on annotations done using a modification of DAMSL, so we had to reinterpret some of the tags in each case.

Tag dimension	Tag	Monroe	Switchboard	TRAINS
Communicative status	Unintelligible	.6%	1%	3.4%
	Abandoned	8%	5%	2%
	Self-talk	.8%	.1%	2.4%
Information level	Task	80%		74%
	Task-management	2%		4.1%
	Communication-mgt.	14%		18.3%
Statement	Assert	40%	49%	36.1%
	Reassert	10%		4.9%
	Other-statement	1%		4.8%
Influence-on-listener	Action-directive	8%	.4%	10.4%
	Open-option	5%	<.1%	2.4%
	Info-request-directive			15%
	Info-request	10%	c. 4.6%	.4%
Influence-on-speaker	Offer	5%	<.1%	5.8%
	Commit	12%	<.1%	19.2%
Conventional			<1.3%	0%
	Opening	.3%		2.5%
	Closing	.3%		0%
Explicit-performative		.3%		0%
Exclamation		1%		.4%
Other-forward-function		5.7%		7%
Agreement	Accept	23%	5%	30.8%
	Accept-part	.2%	<.1%	0%
	Maybe	.2%	<.1%	.2%
	Hold	1.2%		3.2%
	Reject-part	.2%	<.1%	0%
	Reject	.6%	.2%	2.2%
Understanding	Signal-non-understanding	.5%	.1%	1.1%
	SU-Acknowledge	24.5%	23%	26.3%
	SU-Repeat-rephrase	3.5%	.8%	2.5%
	SU-Completion	.7%	.4%	1.6%
	Correct-misspeaking	.3%		0%
	Answer	8%	3%	14.4%

Table 4: Frequency of act occurrence: Monroe vs. Switchboard

formed on their own (although they may be signaled by discourse cues such as “so” and “and then”, which may also signal certain rhetorical relations).

Tag	Frequency of occurrence at specified position		
	Single-utterance turns	Multiple-utterance turns, first utt.	Multiple-utterance turns, last utt.
Assert	33%	19%	28%
Reassert	25%	18%	30%
Other-statement	26%	21%	34%
Info-request	39%	10%	41%
Action-directive	32%	16%	29%
Open-option	28%	18%	34%
Offer	29%	18%	34%
Commit	38%	20%	23%
Other-forward-function	30%	31%	15%
Accept	50%	30%	11%
Hold	68%	29%	0%
SU-Acknowledge	51%	33%	9%
SU-Repeat-rephrase	45%	33%	10%
SU-Completion	43%	33%	10%
Answer	56%	27%	11%

Table 5: Frequency of occurrence of tags occurring more than twenty times

Table 5 shows how often the most frequently-occurring DAMSL tags appear in the turn-initial and turn-final positions. In general, these results confirm our intuitions about the ordering constraint of conversation acts theory. For example, *Understanding* dimension tags most often appear at the start of a turn, and info-request tags at the end of one. However, some of these numbers may seem a little odd. For example, 11% of *Accept* acts occur in the last utterance of a multiple-utterance turn. These are turns entirely made up of *Accept* acts, or of *SU-Acknowledge* and *Accept* acts. Similarly, those *Info-requests* that do not occur right before a speaker change appear in sequences of questions, or are rhetorical questions, abandoned questions or questions that the speaker asks him/herself.

In DAMSL, each tag type is independently marked, so that a single utterance may end up labeled with several tags (e.g. answers to wh-questions may be labeled both *assert* and *answer*). We have used these tags, individually and in combination, to build a variety of models of these dialogues, including n-gram models and “turn-length” models which include sequences of varying lengths that do not cross a speaker change [Stent, 2001].

There are two interesting observations arising from this annotation. One relates to the large number of grounding utterances, especially acknowledgments, produced by both conversation partners. With these we were able to obtain a variety of linguistic forms for use in surface generation. The frequency of these

acts in human-human spoken dialogue highlights the importance of modeling them correctly in a generation system.

The second interesting observation relates to the large number of assertions (and sequences of assertions) in these dialogues. Without a model of argumentation acts, it is impossible to understand how these assertions came to be produced, and therefore how to generate them.

2.3. Argumentation acts

We made two attempts to annotate the Monroe corpus for argumentation acts. Our purpose in performing this annotation was partly to identify the most common argumentation act types. More importantly, we were interested in exploring how the different types of argumentation act fit together; the argumentation act level of conversation acts theory is described by Traum and Hinkelman in fairly vague terms as a heterogeneous collection of disparate elements of discourse structure [Traum and Hinkelman, 1992].

In our first annotation, we used only rhetorical relations. There were several problems with this approach, which we outline in [Stent, 2000b]. We decided we needed to more clearly define the basic units (the dialogue segmentation) for this annotation, and to pick a more complete set of argumentation acts including some specific to dialogue.

For our second annotation, we used a manual we created, which includes adjacency pairs, rhetorical relations and a few structural schemas (e.g. *list*).

Neither of these annotation projects was particularly successful, although the second was much more successful than the first. There are three basic issues. First, our dialogues are long and complex. Second, because we opted to include the greatest variety of argumentation acts possible, our annotation scheme is also complex. We provided several aids to the annotators to reduce this complexity, but our annotators still suffered badly from fatigue. Finally, even when a dialogue has been annotated it is difficult to determine inter-annotator agreement. The different act types are not completely independent, and there are many places in these dialogues where there seem to be at least two valid discourse structures.

In our dialogues, annotators tended to agree about the very high and very low levels of discourse structure. For example, in most dialogues the annotators marked at least two top-level segments: one for the description of the situation, and one for the construction of the plan. They tended to agree to within 1-2 utterances on this boundary, even though all annotators constructed their trees bottom-up rather than top-down. At the lowest level, the annotators tended to agree about pairs of utterances that formed adjacency pairs.

All the adjacency pairs appear in our dialogues. Not surprisingly, *assertion-acknowledge*, *question-answer*, *proposal-accept* and *summons-response* occur most frequently. The most frequently occurring rhetorical relations included some used when describing situations (e.g. *background*, *circumstance*, *non-volitional cause*, *object:attribute* and *generalization:specific*), some used during planning (e.g. *en-*

ablement, justify, volitional cause, solutionhood, evidence, evaluation and correction) and some used when summarizing the plan (e.g. *restatement, summary and sequence*). In our generation system we use a model of adjacency pairs and have content structuring rules for most of these rhetorical relations.

Although each element of the argumentation act level (adjacency pairs, rhetorical relations, structural schemas) is fairly well-understood, it is difficult to conceptualize any of these elements in terms of conversational acts. First, the emphasis in each of these theories is on the relationships between units of discourse, rather than on acts which are in some sense atomic. To put this another way, it is strange to think of a speaker or writer forming the explicit intention of creating a *background* or other rhetorical relation for the sake of producing that relation. It is more natural to think of a speaker or writer intending to communicate some fact, and deciding that in order to be convincing some support for the fact should also be provided.

Second, the roles of the dialogue participants are not as clearly defined in this level. Turn-taking and grounding acts are typically performed by one dialogue participant only, with the other observing. Speech acts are initiated by one dialogue participant and grounded by the other. Adjacency pairs are also typically initiated by one participant and completed by the other. However, rhetorical relations and structural relations were first thought of in the context of the generation of text monologues, where there is only one active participant. Although both dialogue participants can contribute to the production of a *summary* or other rhetorical relation, their respective roles and motivations for doing so are not very well defined.

2.4. Summary

Using the levels of dialogue behavior outlined in conversation acts theory, we were able to account for every utterance in a subset of the Monroe dialogues. Our results broadly confirm the key ideas of conversation acts theory: that dialogue is composed of several distinct act types, which form a hierarchy in which the production of acts at lower levels enables the production of acts at higher levels. However, the turn-taking and argumentation act levels of conversation acts theory seem to fit less well in an act-based framework than the grounding and speech act levels. Also, there is clearly more work to be done in terms of providing an adequate accounting of the argumentation act level.

3. Conversation acts based generation system

The key insights of conversation acts theory are the following:

- Spoken dialogue comprises several distinct types of behavior (or act types), namely turn-taking, grounding, speech acts and argumentation.
- These behaviors are organized in a hierarchy, with lower-level behaviors contributing to the performance of higher-level ones.

An important assumption of conversation acts theory is that these dialogue behaviors are act-based; that is, that they are formed of discrete actions which one can presumably plan using traditional AI planning techniques.

In the previous section, we showed that using conversation acts theory, we could account for every utterance in a set of complex human-human task-oriented dialogues. We also showed that conversation acts theory offers a useful model for how different dialogue behaviors interact. A reasonable hypothesis is that we can use this theory as the basis for a generation system that plans human-like spoken dialogue contributions.

In this section, we describe a generation system for spoken dialogue that uses the insights of conversation acts theory. We look first at where this system fits in the architecture of the TRIPS dialogue system. Then we describe a contribution planning component, which directly models the subsumption principle and ordering constraint of conversation acts theory. Finally, we briefly consider surface generation for different conversation act types, focusing on the generation of language.

TRIPS consists of a heterogeneous set of components that share information using KQML messages [Allen et al., 2001a]. The logical architecture of the current version of TRIPS is shown in figure 3. Viewed one way, there are three basic types of processing in this system: understanding (the upper left section of the diagram), reasoning (the bottom section of the diagram) and generation (the upper right section of the diagram). Viewed another way, there are two types of processing in TRIPS. In the top half of the diagram, the focus is on reasoning about the discourse: how to understand some language or gesture, how to maintain discourse context, how to produce appropriate discourse contributions. In the bottom half, the focus is on problem solving: planning, reasoning about the domain and the world, simulation, and the maintenance of the system’s goals. TRIPS maintains a careful separation between discourse reasoning and problem solving, both for reasons of portability and for conceptual clarity [Allen and Litman, 1990].

Knowledge source	Use
Shared system ontology	Definition of semantic representation, source of information about the domain
Discourse Context component	Information about the ongoing dialogue
Task Manager component	Maps between the abstract plan representation and domain-specific actions, objects and states

Table 6: Knowledge sources in TRIPS

The “core” components of TRIPS, in terms of determining system behavior, are the Interpretation Manager, the Behavioral Agent, and the Generation Manager. All three are event-driven components, incorporating input messages into their ongoing processing. These components use the same representations for

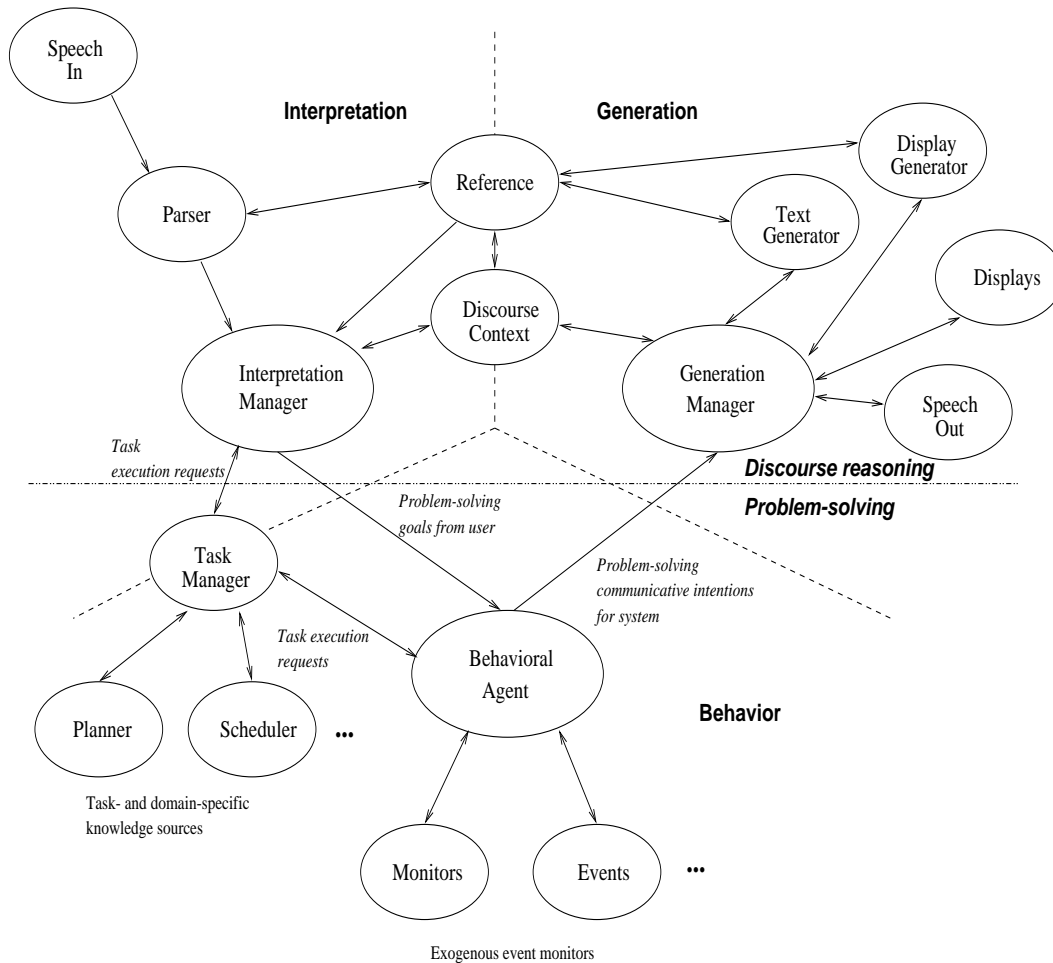


Figure 3: Logical architecture of the TRIPS system

discourse, planning and domain information [Allen et al., 2001b]. They all have access to the same knowledge sources, shown in table 6. This means that each component’s basic processing is domain independent. It also means that there is a well-defined syntax for the inputs and outputs of each component.

The Interpretation Manager performs interpretation in the context of the dialogue so far and of the planning state. It receives information about the dialogue context from the Discourse Context component, and information about the planning state from the Task Manager. When it has determined the discourse-related and problem solving-related intention(s) for each user utterance, it updates the Discourse Context and Behavioral Agent with that information.

The Behavioral Agent is responsible for managing the system’s problem solving. Its behavior is determined by its model of the user’s goals and plans, by its own goals and plans, and by its observations of the dynamic, changing world. It can form intentions to act in the world, to pursue some goal or plan, or to communicate some goal, plan, fact, or situation. For example, it may observe

that a multiple-vehicle accident has occurred at the mall. As a result, it may decide to inform the user of the situation, and to form the goal of getting the victims to the hospital.

The Generation Manager is responsible for managing and coordinating system contributions to the dialogue. It receives the system's interpretation of user utterances from the Interpretation Manager via the Discourse Context. It also receives problem solving-related communicative intentions, such as to communicate a particular fact or a particular task, from the Behavioral Agent. Its task is to organize these disparate pieces of knowledge, select appropriate conversation acts, and combine them into a dialogue contribution for production the next time the system gets the turn.

Both the Interpretation Manager and the Generation Manager reason about and contribute to the formation of the dialogue structure. The Interpretation Manager's responsibilities end with the determination of the user's intentions; the Generation Manager's responsibilities begin with deciding how to fulfill its discourse obligations to respond to what the user has said [Traum and Allen, 1994].

The Interpretation Manager, Behavioral Agent and Generation Manager together do what, in a traditional dialogue system, would be done by the dialogue manager. Because they are independent, concurrently running processes, we are better able to experiment with different styles of interaction, and we are also able to add more incremental processing to the system.

For more information about TRIPS, see [Allen et al., 2000, 2001b].

Generation stage	Description of processing and identification of responsible TRIPS component(s)
intention formation	formation of system's discourse-related communicative intentions (Generation Manager) and of problem solving-related communicative intentions (Behavioral Agent)
content selection	selection of content to be expressed; content may be related to the domain or to planning (Behavioral Agent), or to the dialogue (Generation Manager)
content structuring	selection and organization of dialogue acts into one or more coherent turns (Generation Manager)
sentence planning	organization of content into simple semantic forms (Generation Manager); addition of focus and ordering information (Text Generator)
surface generation	construction of syntactic structure and text (Text Generator), or of display plans (Display Generator)

Table 7: Comparison of the TRIPS generation system and a standard generation architecture

Table 7 shows the stages in a standard generation architecture [Reiter, 1994]. Many end-to-end dialogue systems combine dialogue management and content planning, and perform surface generation using template-based generation (e.g. [Rich and Sidner, 1998, Stent et al., 1999]). We do not take this approach, but also do not follow the standard generation architecture completely.

In TRIPS, because the system maintains a very clean separation between problem solving and discourse reasoning, the Behavioral Agent and the Generation Manager share responsibility for intention formation and content selection. As the dialogue progresses, the Behavioral Agent forms intentions to act and communicate based on progress in problem solving, and the Generation Manager forms communicative intentions based on discourse obligations arising from user input. The communicative intentions formed by the Behavioral Agent are necessarily intentions to communicate some fact, goal or process related to the problem solving or the domain; that is, intentions to communicate some propositional content. The communicative intentions formed by the Generation Manager may require content (for example, the intention to answer a question); typically, this content is either present in the discourse context or comes from the Behavioral Agent. If the content to be communicated relates to the discourse (for example, if the system was unable to understand part of the user's utterance), the Generation Manager may construct the content itself.

The Generation Manager is responsible for mapping communicative intentions to conversation acts and organizing these conversation acts into coherent and natural dialogue contributions. This process is similar to content structuring in that it is the organization of basic elements into a coherent discourse structure. In this case, however, the basic elements are sets of communicative intentions and conversation acts. We call this process *contribution planning*.

The Generation Manager is also responsible for transforming content into semantic logical forms, each of which can typically be realized in one utterance. However, the Text Generator is responsible for ordering the information within the utterance, selecting lexical items and determining the form of referential expressions. The Display Generator is responsible for selecting parts of the semantic form for display and selecting the display objects to use.

In the following sections, we describe in detail how we used the insights of conversation acts theory in contribution planning and surface generation processes for TRIPS.

3.1. Contribution planning

The input to the contribution planning process is communicative intentions, perhaps with associated content. However, these communicative intentions are not all formed at once. In the context of an ongoing dialogue communicative intentions may be formed at any time. For example, if the system cannot hear the user, or cannot understand what the user said, the Generation Manager may form a discourse-related communicative intention to ask the user to speak up or speak more clearly. If the system believes that the user has asked a question, the Generation Manager will typically form a discourse-related communicative intention to acknowledge and answer that question; however, it may not have an answer immediately. Also, the Behavioral Agent may form a problem solving-

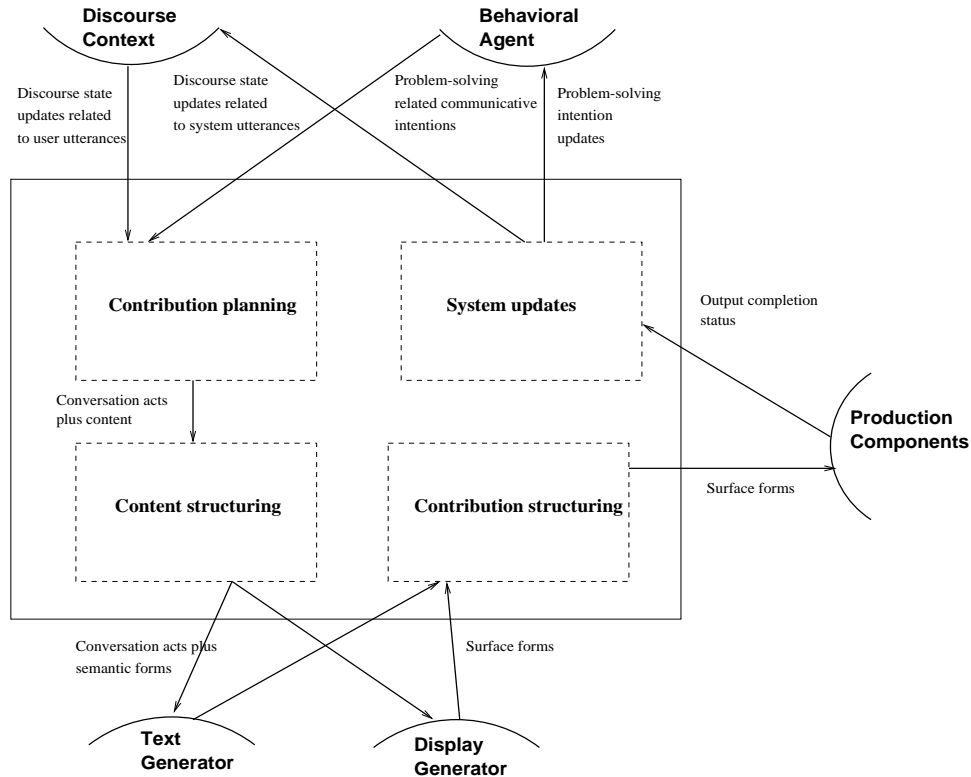


Figure 4: Tasks performed by the Generation Manager

related communicative intention to present a situation or plan regardless of what the user is currently saying.

It is important, for several reasons, that the Generation Manager not wait for a pre-set signal (such as a message from the Interpretation Manager that the user's utterances have been fully interpreted) before beginning to plan a contribution. Practically speaking, the system has a very small amount of time after the user stops speaking before it must begin to respond. Also, the system may need to interrupt the user. Finally, the system may decide to say something even if the user is not speaking. The way to perform contribution planning is therefore to be constantly planning based on what is known so far, in the manner of a person who uses the time provided by a conversational partner's talking to plan her next witty remark.

Figure 4 shows the different tasks performed by the Generation Manager. In our implementation each new communicative intention or discourse state update triggers contribution planning, which may lead to the planning of one or more new conversation acts. A second stage, here called *contribution structuring*, filters and orders the output from the surface generation modules and is responsible for deciding which acts are actually produced. In short, contribution planning over-generates and contribution structuring prunes. Each stage incorporates slightly different aspects of conversation acts theory: contribution planning maps inten-

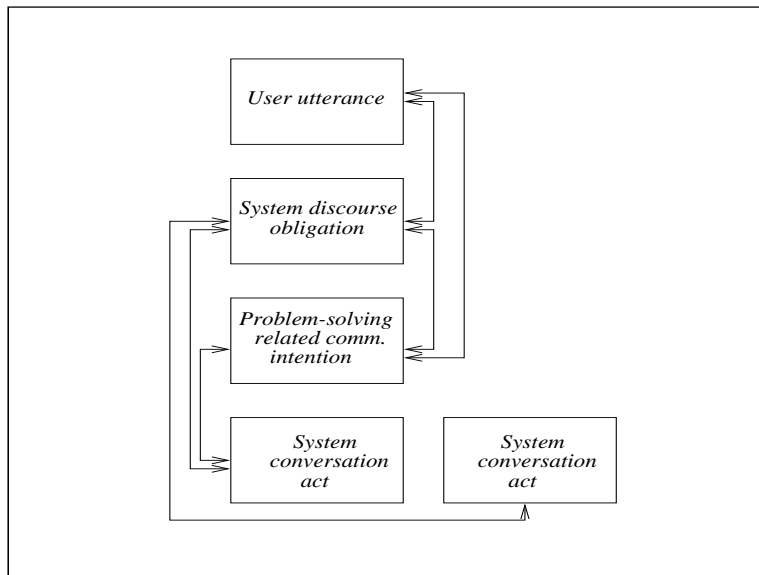


Figure 5: Possible links between user utterances, discourse obligations, problem solving-related communicative intentions and conversation acts in a contribution

tions onto levels of dialogue behavior and individual conversation acts, while contribution structuring uses the ordering and subsumption relationships that hold between the different levels of conversation act.

The basic unit in the Generation Manager's state is the current contribution. A contribution is typically one turn long but may span several system turns as long as the intervening user utterances do not change the problem solving state of the system (and therefore possibly make previous communicative intentions unnecessary). The Generation Manager stores quite a lot of information about each system contribution, including:

- all user utterances that are responded to in this contribution
- all discourse obligations (communicative intentions arising from user input [Traum and Allen, 1994]) that are fulfilled in this contribution
- all problem solving-related communicative intentions that may be satisfied in this contribution
- all output conversation acts that could be or have been produced in this contribution
- all output from other modalities that could be or have been produced in this contribution

In addition, each utterance, discourse obligation, and problem solving-related communicative intention has quite a lot of information linking it to other utterances and/or goals in the contribution state (figure 5).

3.1.1. Top-down contribution planning

The contribution planning process decides which conversation acts to use to fulfill each communicative intention, using a set of rules that model various conversation acts. Most of these rules are for argumentation acts, including adjacency pairs and rhetorical relations; but some are for acts at the grounding or turn-taking level. Each rule contains a set of conditions, and a set of actions to perform if all the conditions are met.

The conditions in a contribution planning rule may specify that certain discourse obligations and/or problem solving related communicative intentions must be present in the turn state, and may also specify that variables internal to the Generation Manager (such as initiative holder) must have a certain value. The “actions” are really conversation acts with associated top-level syntactic form and content. The content is selected from the content associated with the discourse obligations or problem solving goals in the conditions. Rule actions may also change the values of internal Generation Manager variables.

Figure 6 shows example content planning rules. The first rule says that if the user has just released the turn, the system may perform a *take-turn* act. The second says that if there is an obligation to respond to a *wh-question*, and that question led to problem-solving goal ?psid, and there is a problem solving-related communicative intention to communicate content ?answer in order to complete that same problem-solving goal, then the system can respond by *asserting* the specified content in answer to the question. The third rule says that regardless of whether or not there is an outstanding discourse-related intention, if there is a problem solving-related intention to summarize some information, that can be done by *asserting* the information using both speech and displays. The content structuring process (described in the next section) segments the content from these last two rules into utterance-length chunks; if more than one semantic form is output from that process, the *assert* conversation act applies to all of them.

Each time the contribution state is updated, the contribution planning process finds all the contribution planning rules that apply and weights them according to the degree to which the rule conditions cover the entire contribution state. It then selects one of the rules from the set using a randomness factor. After the content structuring algorithm arranges the content, the conversation act(s), content and top-level syntactic category are sent to the surface generators for each modality selected by the rule (the default is speech), and any other rule actions are executed.

There is no backtracking during contribution planning; if the modality-specific generators fail to return surface forms, another rule is not selected. This is because the contribution state can change quite dramatically over a short period of time with the addition of new goals, new planned utterances and so on. However, if the state does change in particular ways (e.g. a discourse obligation or a problem solving-related communicative intention is added, a timer times out) then

```

(contribution-planning-rule
 :obligations {(respond-to :who system
                            :what (release-turn :who user))}
 :conversation-acts {(take-turn :syntax s)})

(contribution-planning-rule
 :obligations {(respond-to :who system
                            :what
                            (wh-question :who user
                                          :what ?question
                                          :why (initiate :what ?psid)))}
 :problem-solving-intentions {(tell :who system
                                   :what ?answer
                                   :why (complete :what ?psid))}
 :conversation-acts {(assert :what ?answer :syntax s)})

(contribution-planning-rule
 :problem-solving-intentions {(tell :who system
                                   :what ?content
                                   :why (summarize :what ?psid))}
 :conversation-acts {(assert :semantics ?content
                            :in-relationship-to (summarize :what ?psid)
                            :modalities (speech display)
                            :syntax s)})

```

Figure 6: Example contribution planning rules

the content planner will be called again, so that planning may happen several times for the same contribution. If this results in inconsistent dialogue acts, the contribution structuring stage will resolve the issue.

3.1.2. Content structuring

After a conversation act has been selected and content assigned to it, the content must be transformed into one or more semantic logical forms. The process that does this uses the shared TRIPS ontology, which includes a mapping between domain and planning information (actions, states and objects) and the role-based semantic representation used in TRIPS. For example, if the piece of content to be communicated is that ambulance 4 will go to Strong Memorial hospital, the content structuring process may find that this type of action can be mapped onto the semantic action `lf-take`, which has required roles for `agent` and `object` and optional roles for `source` and `destination`. The `agent` role will be filled by the semantic description for `ambulance-4`, which specifies that this is a reference to an object of type ambulance. The `object` role will be filled by the semantic description for `person-5`, `person-6`, which specifies that this is a reference to

a set of objects of type person. The *destination* role will be filled by the semantic description for *Strong Memorial Hospital*, which specifies that this is a reference to an object of type hospital. There is no additional information to be communicated for this example, so only one semantic form is output. Focus information will be added to the semantic descriptions in the text generation component. The final semantic form for this piece of context (simplified from the TRIPS-internal representation) might be:

```
(LF-TAKE (AGENT
  (DESCRIPTION (TYPE ambulance)
    (REFERS-TO ambulance-4)))
(OBJECT
  (DESCRIPTION (TYPE person)
    (REFERS-TO (person-5 person6))))
(DESTINATION
  (DESCRIPTION (TYPE hospital)
    (REFERS-TO strong-memorial-hospital))))
```

Each conversation act, with associated content and top-level semantic label, is sent to the surface generators specified in the contribution planning rule. If the text or display generator replies with a surface form, the contribution state will be updated. Surface forms are linked to the conversation acts they realize, and the conversation acts themselves specify in which modalities they must be realized. When all the modality-specific surface forms for a conversation act are present in the contribution state, the contribution structuring process is called.

3.1.3. Contribution structuring

The contribution structuring process uses a set of rules that specify how the different levels of conversation act interact, and how conversation acts within the same level may be organized. Because of the subsumption relationship holding between the different levels of conversation act, the contribution structuring process has some freedom to choose which surface forms to produce. For example, if it has a surface form realizing a *take-turn* act and one realizing a speech or grounding act it may choose not to output the surface form for the turn-taking act. If it has surface forms for a grounding act and a speech act, it may choose to produce only the speech act. From our analysis of the Monroe corpus, we know approximately how often lower-level acts are subsumed in higher-level ones, and how often they are explicitly produced.

The ordering constraint of conversation acts theory imposes some structure on where acts may appear in the contribution. For example, if the contribution structuring algorithm decides to output the surface form for a grounding act, that should typically precede the production of any speech acts in that turn.

Finally, our analysis of the Monroe corpus dialogues gives some information about how to combine acts at the same level. There are some general rules; for

```

(contribution-structuring-rule
 :conditions ((time > 5)
              (this-act :act (not {release-turn assign-turn})
                        :level turn-taking :produced false)
              (other-acts (not {(act :act ?_ :level ?_
                                     :produced true)})))
 :level turn-taking
 :results output
 :actions {(plan-new-act keep-turn)})

(contribution-structuring-rule
 :conditions ((this-act :act accept :level speech-act
                       :produced false)
              (other-acts {(act :act acknowledge :level ?_
                               :produced true)})))
 :level speech-act
 :probability .33
 :results output)

```

Figure 7: Example contribution structuring rules

example, if there are two acts at the same level, of the same type and with the same content (e.g. two *acknowledgments*), then only one need be produced. There are also specific ordering and selection constraints for the production of turn-taking, grounding and speech acts, although these are different for each level of conversation act. For example, if the user said two things, only one of which the system understood, the contribution structuring process may end up with two grounding acts, one an *acknowledgment* and the other a *signal-non-understanding* act. The most specific act (the *signal-non-understanding*) is the one that should be produced. If the contribution structuring algorithm has two speech acts, a question and an assertion, the assertion should precede the question in the turn unless there is a rhetorical relation holding between the two speech acts.

Each contribution structuring rule has constraints, actions and a weight (defaulting to 1). Three types of constraints can be specified in the conditions of contribution structuring rules: timing constraints, constraints on the current act of interest, and constraints on the other dialogue acts in this turn. There are two types of timing constraints: constraints on the time since the user stopped speaking, and constraints on the time since the system's last utterance. Time is measured in seconds. These timing constraints are necessary to model the time-dependent aspect of turn-taking behavior. Constraints that may be specified on a conversation act include constraints on its act level, its act type, and whether it has been produced. Also, it is possible to check for the total number of con-

versation acts in the contribution state that belong to a certain act level, have a certain act type, or have been produced.

There are two “action” fields for these rules: the *:result* field specifies whether output will be produced for this act, and the *actions* field may specify other actions to be taken.

The weights for the rules used in contribution structuring are derived from the annotation of the Monroe corpus, and they indicate the relative frequency of certain types of acts or certain combinations of acts. For example, turn-taking acts are usually performed only verbally only if there is nothing else to say and the system has run out of time to start the turn. However, in the corpus turn-taking acts appeared roughly 10% of the time, so about 1 time in 10 the system may produce a turn-taking act even if there is other pending output in the current turn state.

Example contribution structuring rules are given in figure 7. The conditions for the first rule say that the time since the user last stopped speaking must be greater than 5 seconds, that this act must be some act that takes or keeps the turn, and that there can be no other already produced turn-taking act in this turn. If these conditions are met, the system will produce this turn-taking act and start planning a keep-turn act. The second rule says that an *accept* act will be explicitly produced after an *acknowledge* act with probability 1/3.

The algorithm for rule selection in the contribution structuring process is similar to that in the contribution planning process. Rules are selected from among the rules that match the most closely roughly according to their weight. The rule’s actions are then performed; typically, these are either to produce output or to simply update the discourse context or problem solving state without producing output.

3.1.4. Discourse state update

When production of a system utterance is complete, the Generation Manager receives a status update from the relevant modality-specific production components. It then updates the other TRIPS components. As with user utterances, the words, syntactic structure and semantic form of each system utterance are sent to the Discourse Context. These may give rise to new user obligations (just as the system is obliged to respond to the user’s utterances, so the system believes that the user is obliged to respond to its utterances). Also, if the conversation act realized by the system utterance fulfills one of the system’s discourse obligations, the Discourse Context is updated to indicate that the relevant user utterances have been responded to. Finally, if the conversation act realized by the system’s utterance fulfills a problem solving-related communicative intention, the Behavioral Agent is also updated.

Some of the problem solving-related communicative intentions require substantial output to be completely fulfilled. If the system is not able to communicate all the required information (for example, it is interrupted by the user, the

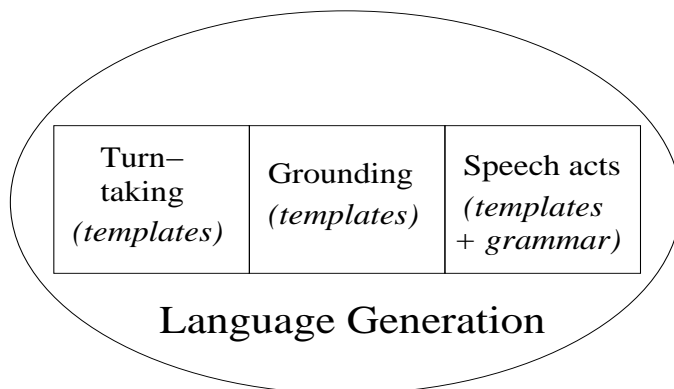


Figure 8: Structure of the Text Generator

user asks a question, or the Behavioral Agent cancels the intention), then it tells the Behavioral Agent how much of the content it was able to communicate.

3.2. Text generation

The modeling of the different levels of dialogue behavior does not stop at contribution planning and structuring, but continues through surface generation. There are two reasons for this. First, we wished to continue to use the subsumption principle during surface generation. Second, our analysis of the Monroe corpus revealed that some important generalizations can be made about conversational behavior that affect surface generation and can contribute significantly to the efficiency, portability and generality of a generation system.

The template-based generation used in previous versions of TRIPS was too limited to work effectively in larger and more complex domains. We needed a surface generator that would allow us to rapidly expand language coverage while maintaining efficiency, and that would permit us to produce surface forms for different types of conversation acts. There are some excellent syntactic generation systems already in existence, but a combination of factors led us to reject each one, even though each was superior in some way to what we could create on our own. Some were too slow, some would have required extensive and complex modification of the semantic representation, and some included components, such as context components, already to be found in TRIPS.

In the approach we have taken, we have tried to marry portability and efficiency. As illustrated in figure 8, we aimed for a Brooksian architecture where individual behaviors are modeled separately and in parallel, so that simpler behaviors such as turn-taking will continue to function even if more complex behaviors break down [Brooks, 1991]. We use templates, which are efficient to process, for behaviors that do not vary much from one domain to another (such as turn-taking and grounding). We use a lexicalized grammar for surface generation for speech acts, to obtain the most general and portable system possible. Finally, we have incorporated considerable randomness into our surface genera-

tion to provide much-needed variation, leading to more natural and interesting system output.

As we saw in the previous section, the content structuring process in the Generation Manager organizes content into semantic forms for realization as individual utterances. It is important for this to take place in the Generation Manager, since that component tracks the planning process from intention to production, and maintains the contribution plans as they are built. The Text Generation component also performs some aspects of sentence planning. It orders the information in the semantic form and adds default values for some roles. It also queries the Discourse Context for focus information, and uses that to determine the type of each referential description. In short, the Generation Manager determines what will be included in each utterance, and the Text Generator determines what needs to be known about each part of the utterance and how the utterance will be structured.

After this initial processing of the input semantic form, the conversation act, semantic form and top-level syntactic label are input to each surface generation process. The surface generation processes for turn-taking and grounding acts use templates, for several reasons. The production of these types of acts typically does not involve forming complex structures to represent a complex semantics. The utterances that produce these types of acts are typically formulaic and do not vary much from one domain to another, so the templates do not have to be modified when the domain changes. Finally, speed is very important for producing these types of acts [Stent, 1999].

Example templates from our system are given in figure 9. Each contains a partly or fully specified surface form and a set of conditions. If the conversation act, syntactic form and input semantic form match the conditions, the syntactic tree and output semantic form will be returned to the Generation Manager. The first template is for a *take-turn* act; the output words are “um”. The second is a template for a *signal-non-understanding* act. The output words are “Sorry, I don’t know how to do that”.

We use a feature-based lexicalized tree adjoining grammar (TAG) to perform surface generation for most speech acts [Joshi and Schabes, 1997]. The use of a grammar allows us to do the sort of complex reasoning about semantics, syntax and pragmatics we require, and also gives greater flexibility and portability (at the cost of a little efficiency and more initial development time). A TAG-based formalism lends itself to a combination approach such as the one we have chosen, since one can use the same structures (trees) to represent templates and grammar rules.

The grammar we use is a fairly small feature-structure based lexicalized TAG, constructed at run time from a variety of sources. To facilitate grammar creation and re-use, we have a set of about 100 tree templates for different types of lexical items, which are filled in with specific information for each lexical item at run-time (as in [XTAG Research Group, 1995]). The lexical items themselves are obtained semi-automatically. Some come from the description of the domain

```

(template
  :input-semantic (take-turn)
  :output-semantic (take-turn)
  :type s
  :act-type turn-taking
  :kids
    ((word :name um
           :type word)))

(template
  :input-semantic (reasoning-failed (in-response-to (request)))
  :output-semantic (reasoning-failed (in-response-to (request)))
  :type s
  :act-type grounding
  :kids
    ((non-terminal :type s :kids
      ((word :name sorry :type discoursecue)))
     (punc :name comma)
     (non-terminal :type s :kids
      ((non-terminal :type np :kids
        ((word :name i :type pro)))
       (non-terminal :type vp :kids
        ((non-terminal :type vp :kids
          ((word :name do :type vaux)
           (non-terminal :type vp :kids
            ((word :name not :type vaux)
             (non-terminal :type vp :kids
              ((word :name know :type v))))))))))
       (non-terminal :type advp :kids
        ((word :name how :type adv)))
       (non-terminal :type s :kids
        ((non-terminal :type vp :kids
          ((non-terminal :type vp :kids
            ((word :name to :type vaux)
             (non-terminal :type vp :kids
              ((word :name do :type v))))))
           (non-terminal :type np :kids
            ((word :name that :type pro))))))))))
    ))))

```

Figure 9: Example templates

provided in the shared system ontology. Others were obtained by automatic off-line extraction from the TRIPS parser's lexicon. Finally, many were obtained by semi-automatic extraction from the Monroe corpus transcripts, which were first tagged for part of speech using Brill's POS tagger [Brill, 1992]. Not counting proper nouns or other lexical items from the shared system ontology, there are about 750 lexical items in the smallest version of the lexicon and 1500 in the largest.

We use a subset of the XTAG feature set in our grammar [XTAG Research Group, 1995]. We follow [Stone and Doran, 1997] in extending the scope of our rules to include semantic and pragmatic information, although we do not use a feature-based approach to the representation of these types of information.

This grammar does contain a few "back doors", in case the system has to produce something not in its lexicon. There is a set of non-lexicalized trees for proper nouns, numbers (of course), adjectives and adverbs. These are used very rarely, since our lexicon provides extensive coverage for our domain. Eventually, our lexicon will be fully integrated into the system's ontology and there will no longer be a coverage issue. Finally, in addition to the grammar we have a small set of templates for producing greetings, closings and thanks, which use highly idiomatic language.

The generation algorithm for all three conversation act generation processes is essentially a top-down depth-first search with backtracking, keyed off the semantic tree. In our system, there is considerable randomness built into the generation process. Each grammar tree or template has a weight associated with it (the default is 1). At each point in the generation process, all the applicable trees are found and their weights normalized. Those that most closely match the semantic form are singled out. One is selected from among them at random, so that each tree has a chance of being chosen proportional to its weight. This guarantees variety in the system's utterances, which is important if one is not to bore the user (especially when the user has the initiative and the system is only responding).

The output from the text generation process for each level of dialogue behavior is a fully specified syntactic parse tree, which is linked to the fully specified semantic form. The actual text is constructed from the parse tree using a simple model of morphology. All three items are then sent back to the Generation Manager.

3.3. Display generation

For spoken dialogues humans have very strict conventions defining appropriate and natural linguistic behavior, and this work is an examination of how to make explicit those conventions in a generation system. The equivalent conventions for non-linguistic communication are not as well understood. Indeed, for some types of non-linguistic communication, such as that involving the graphical user interface, the conventions are still being defined.

TRIPS has traditionally used displays to supplement speech or facilitate the

communication of certain types of information, not to add personality to the system (*c.f.* [Cassell et al., 1999]). The system uses displays to present information that would take a long time to communicate using language, that should persist in the forefront of the discourse context to help the user perform planning, or that should be emphasized. For example, TRIPS draws routes on maps, and summarizes plans on a plan viewer, as planning proceeds. Route and plan descriptions involve the communication of temporal and spatial information. These types of information can be difficult to express using language and difficult to remember. A visual representation is concise and may persist, making it easier for the user to see the connections between the different parts of a plan.

TRIPS also uses the map, the plan viewer and other displays to communicate information about sets of objects that it would be ineffective to communicate using language. For example, if the user asks where the ambulances are and there are ten ambulances, it is silly to describe all their locations verbally when a display is available.

Finally, TRIPS uses the displays to emphasize certain objects or situations, in order to express focus information or urgency. In answering a question, it may highlight the objects in the answer on the map. If something changes in the world that necessitates immediate re-planning, this can be highlighted on the display so that it serves as a constant reminder to the user.

Because the displays in TRIPS are used to supplement the spoken communication rather than as a primary communication modality, we were not concerned with modeling the different levels of conversation acts in display generation. The Display Generator receives the same information as the Text Generator: one or more conversation acts, with associated content. It identifies domain and planning objects in the semantic forms (e.g. vehicles, people, actions, situations) and forms a display plan for production with the speech output. The display plan contains a set of actions, each of which manipulates one or more display objects in some way. For example, a display object for a planning action may be added to the plan display, or an icon representing an ambulance may be highlighted.

Coordination in TRIPS occurs at the conversation act level: a display plan for a particular dialog act will not be produced until the corresponding language plan has also been received by the Generation Manager. Because we do not currently produce gestures, we do not have to worry about more fine-grained coordination such as that described in [Cassell et al., 2000].

3.4. Example

We conclude the discussion of our generation system by presenting an example that demonstrates the flexibility of our approach to contribution planning. Rather than show the message traffic between TRIPS components, we will show communicative intentions and pieces of discourse information as they appear in the contribution state, using a frame-like notation.

Assume that the user asks “Where are the ambulances?”. The first piece of

discourse-related information that the Generation Manager receives is that the user has released the turn. As a result, the following goal is added to the contribution state:

```
(goal :type discourse-obligation
      :id discourse-oblig1
      :respond-to (release-turn :who user)
      :who system)
```

This goal is a discourse obligation for the system to respond to the user's action of releasing the turn. For ease of reference, every item in the contribution state is assigned a unique identifier.

After a communicative intention or discourse obligation is added to the contribution state, the contribution planning algorithm is called. The only rule that matches the current contribution state says that if the user has released the turn, the system should take the turn. Accordingly, the Generation Manager adds the following conversation act to the contribution state:

```
(conversation-act :type turn-taking
                  :name take-turn
                  :id system-conversation-act1
                  :semantics nil
                  :syntax s
                  :fulfils discourse-oblig1
                  :who system)
```

It also sends a message to the Text Generator asking it to generate a *take-turn* act. The turn-taking generation process in the Text Generator may decide that an appropriate realization of a *take-turn* act is "um". This information is returned to the Generation Manager, which updates the contribution state:

```
(conversation-act :type turn-taking
                  :name take-turn
                  :id system-conversation-act1
                  :semantics nil
                  :syntax s
                  :words 'um'
                  :fulfils discourse-oblig1
                  :who system)
```

Meanwhile, the Generation Manager may have been informed that the user intended to ask a question about the location of ambulances in the world. This gives rise to a new discourse obligation:

```
(goal :type discourse-obligation
      :id discourse-oblig2)
```

```

:respond-to
  (wh-question :who user
               :what (at-loc
                     (description (class set)
                                   (type ambulance))
                     (description (class wh)
                                   (type location))))
  :who system)

```

At this point, the only contribution planning rule that matches the contribution state says that a question can be responded to (and the turn taken) if the question is acknowledged. The grounding process of the Text Generator produces a surface form for an *acknowledge* act, which is added to the contribution state:

```

(conversation-act :type grounding
                 :name acknowledge
                 :id system-conversation-act2
                 :semantics nil
                 :syntax s
                 :text 'mmhm'
                 :fulfils {discourse-oblig2 discourse-oblig1}
                 :who system)

```

These first two steps in the planning of a contribution happen for nearly every contribution following user input. If the time required to reach this point is greater than the permitted between-turn gap specified in the Generation Manager, then the contribution structuring process will have to choose something to say (this does not happen very often). In this case, the two acts ready for production in the contribution state are a grounding act and a turn-taking act, and the contribution structuring process has a rule saying that grounding acts subsume turn-taking acts, so the system will probably say “Mmhm”. However, there is another content structuring rule that says that about 10% of the time a *take-turn* act should be explicitly realized even if there is another act in the contribution state, so there is a chance the *take-turn* act will be explicitly produced as well as the grounding act.

Notice that we have already seen uses of the subsumption principle and ordering constraints, even though the only conversation acts seen so far are turn-taking and grounding acts.

By now the Behavioral Agent has been informed by the Interpretation Manager that it has a new problem solving goal of identifying the locations of ambulances (we will label this goal *user-problem-solving1*). What happens next depends on whether the Behavioral Agent finds these locations.

If the Behavioral Agent finds the locations of the ambulances identified in the user’s problem-solving goal, it will send the Generation Manager a communicative intention to inform the user of this information. The communicative intention may look like this:

```
(goal :type problem-solving-intention
      :id system-problem-solving1
      :communicate {(at-loc ambulance-1 strong-memorial-hospital)
                   (at-loc ambulance-2 strong-memorial-hospital)}
      :who system
      :why (complete user-problem-solving1))
```

communicative intention has a marker for the relationship of this intention to the system's and/or user's problem solving goals. In this case, the user initiated the problem solving goal of identifying ambulance locations. If the system succeeds in communicating these locations to the user and the user acknowledges having received the information, the system will have completed the user's problem solving goal.

With the addition of intention `system-problem-solving1` to the contribution state, the contribution planning process is run again. Now the contribution state includes discourse obligations to take the turn and to respond to the user's question, and the problem solving-related communicative intention of providing the locations of ambulances in answer to the user's question. One of the example contribution planning rules shown earlier specifies how to combine these pieces of information into an answer for the user. That rule requires that both speech and displays be used for output.

The content structuring algorithm is smart enough to combine the two pieces of content in `system-problem-solving1` into an aggregate location description:

```
(at-loc
  (description (class set) (type ambulance)
               (refers-to {ambulance-1 ambulance-2}))
  (description (type hospital)
               (refers-to strong-memorial-hospital)))
```

A new conversation act for the answer is added to the contribution state, and this semantic form, the *assert* conversation act, and the top-level syntactic label `s` are sent to the Text and Display Generators.

One of the modality-specific surface generators will reply before the other. As a result, `system-conversation-act3` will be updated to include the reply from that surface generator, but no output will be produced until both surface generators reply. If all of the processing described above takes place "quickly enough", which is generally the case, the entire system contribution may consist of the utterance "Ambulance 1 and ambulance 2 are at Strong", plus the two display actions of highlighting ambulances 1 and 2. In this case, three conversation acts will have been generated, but only one explicitly produced. If the processing takes longer, the take-turn act, the grounding act or both may also be explicitly produced.

There are many times when the Behavioral Agent may decide to communicate with the user. For example, it may become aware of a new situation in the world

or an update to a current one. It may decide to notify the user of a new goal or a new planning action. It may also decide to inform the user if an action in the plan is finished or cannot be completed. The Behavioral Agent's decisions about the goals it wants to pursue and the things it wants to communicate are not necessarily the same as the user's, particularly if the system knows something the user does not. Even in this example, where the system has strong obligations to respond to the user's question, the Behavioral Agent may decide to pursue some other goal.

Assume that the Behavioral Agent has become aware that there are a new water main break and power line down, and wants to inform the user about them. It may send the following communicative intention to the Generation Manager:

```
(goal :type problem-solving-intention
      :id system-problem-solving2
      :communicate {(new-situation
                    (at-loc (object (type water-main)
                                   (status broken))
                           culver-road)
                    (at-loc (object (type electric-line)
                                   (status broken))
                           route-383))})
      :who system
      :why (initiate system-problem-solving2))
```

Notice that this communicative intention arises from a new system problem solving goal, and is not performed in order to continue or complete a user's goal. In other words, the system is now taking the initiative.

The Behavioral Agent may select this communicative intention instead of or in addition to the earlier intention `system-problem-solving1`. If the former, the Generation Manager will have to find an alternative way to respond to the question. When it sees intention `system-problem-solving2`, the Generation Manager assumes that no answer will be forthcoming and plans to produce a *hold* act to respond to the user's question. The final system output, using the *hold* act for the question and describing the new situation using a *generalization:specific* relation, is likely to be something like "Just a second. I have a new situation. There is broken water main at Culver road. There is also a broken electric line at route 383.", plus appropriate display actions.

If the Behavioral Agent produces both communicative intentions, the Generation Manager will have to decide which speech-act level conversation act to produce first. Before producing conversation acts for new problem solving-related communicative intentions, the system tries to produce conversation acts for all communicative intentions that are in some way responsive, so the final spoken output will probably be "Ambulance 1 and ambulance 2 are at Strong. I have a new situation. There is broken water main at Culver road. There is also a broken

electric line at route 383.” This contribution can even be produced if the user acknowledges the answer to her question in the middle, since the acknowledgment does not require any modification to the system’s problem solving.

After each conversation act is produced, the Generation Manager sends discourse and/or problem solving-related updates to the rest of the TRIPS system.

This example illustrates some of the generation processing in our system. It also highlights the range of possible system behaviors. Often, the system is primarily responding to user input, and so the conversation acts that are considered include turn-taking and grounding acts as well as adjacency pairs and whatever speech acts are part of them. At other times, the Behavioral Agent may pursue some goal of its own, informing the user of a new situation or describing a new action or plan. In these cases, the conversation acts involved may also include rhetorical relations, such as *generalization:specific*, *summary* and *motivation*. Finally, there are various ways for the dialogue to break down (e.g. the system can’t hear the user, the system can’t understand the user, or the system can’t do what the user asks). In these cases, the generation system may have to use a different set of relations, such as *solutionhood*. For example, it may say, “Um I can’t hear you. Could you speak up?”

4. Evaluation

In this section we discuss an evaluation we have performed on the generation system described in this paper, for the purpose of determining if our system does indeed model human conversational behavior.

4.1. Evaluation methodology

There are several different types of evaluation performed on dialogue systems or their component parts [Minker, 1998, Antoine et al., 2000]. Researchers may conduct a language-based evaluation, evaluating the types of language handled, the amount of language handled, or the accuracy of the language modeling in a system. Alternatively, they may conduct a task-based evaluation, evaluating how and to what extent the system helps the user fulfill a designated task. Measures used in task-based evaluations include the amount of time it takes to complete the task, the optimality of the solution, and the number of user turns [Sikorski and Allen, 1996, Stent and Allen, 1997, Walker et al., 1998].

There have also been task- and language-based evaluations of generation systems. For example, Hirasawa *et. al.* evaluated different approaches to system backchannels [Hirasawa et al., 1999]. Litman *et. al.* compared two different response strategies in an information-retrieval dialogue system [Litman et al., 1998]. Carenini has performed a task-based evaluation of a system to generate evaluative arguments [Carenini, 2000]. Jokinen has proposed three metrics for evaluating a system’s communicative ability: cooperativeness, robustness and coherence [Jokinen, 1996].

Some researchers have used human judges to evaluate the output of generation systems. Cawsey used human judges to evaluate the explanations generated by her system for coherence and instructiveness [Cawsey, 1993]. Lester and Porter used human judges to score automatically-produced explanations and explanations produced by humans, and then compared the scores of the two sets of documents [Lester and Porter, 1997].

The central issue of our work has been to enable a dialogue system to produce more “natural”, human-like contributions. String- or tree-based metrics based on pairing system output with desired system output (e.g. [Bangalore et al., 2000]) cannot be used here, because there is no one “right” output for any input goals (although there are many “wrong” outputs). Such a metric would give the impression that our system is performing badly, when in fact flexibility and variability in the output are positive outcomes.

A task-based evaluation would allow us to address the question of whether more natural system contributions help users solve tasks, or whether users preferred such contributions. However, it also would not answer the question of whether the system contributions are more “natural”.

The evaluation we have chosen to perform, then, is a language-based evaluation of the generation components of TRIPS when separated from the rest of the system. Since the best judges of “naturalness” are humans who are themselves experts in the type of discourse being evaluated, we chose to use human judges to evaluate our system output. We asked two linguists and one teacher of English as a foreign language to evaluate a set of dialogue transcripts for us.

We selected three dialogues from the Monroe corpus that had not been previously annotated. For each of these three dialogues, we replaced one participant’s contributions with contributions produced by the system that included roughly the same content. Each evaluator received six transcripts: the transcripts of the three modified dialogues, and transcripts of three unmodified dialogues. Each was also given a map of the domain, a list of the tasks, and some information about the original data collection. We asked each judge to evaluate the three modified dialogues, using the other three as reference material. No evaluator was told that sometimes the dialogue involved a computer.

The two linguists, a graduate student and a post-doctoral fellow, were given a check-list of items to note in each dialogue. However, one did not use it, instead marking in the transcripts things that seemed odd to her and explaining why. The teacher of English as a foreign language was simply asked to evaluate the level of expertise in English of the dialogue participants, or at least to indicate whether they were native or non-native speakers.

This evaluation is far from perfect. First, the evaluators were unable to participate in dialogues themselves, and so interact with the system directly (this particularly affects the judgments of the teacher of ESL). They were also unable to hear the dialogues (the prosody of a text-to-speech system would have given the game away). However, each evaluator had experience working with spoken

dialogue and reading dialogue transcripts, and we did supply limited information about pauses and non-verbal sounds.

Second, this evaluation covers only those aspects of the system's contributions produced using language. The multi-modal aspects are left un-evaluated.

Third, having to produce something approximately equivalent to the original human contribution limited some aspects of the system's performance. For example, if a new situation comes up in the world the Behavioral Agent can tell the Generation Manager all about the situation at once, and the Generation Manager can then construct a multi-utterance description. This was not always possible in this evaluation, because the original human participant organized the information differently.

Fourth, using human judges introduces an element of subjectivity into the evaluation (this is perhaps unavoidable given the goal we have set ourselves).

Nonetheless, this type of evaluation does address the central question of our work. Given the circumstances and the nature of generation itself, this is perhaps the optimal solution.

4.2. Construction of evaluation dialogues

As stated previously, three dialogues were selected for use in the evaluation. They were not selected at random. Length was a concern. Also, it seemed important to use dialogues involving different speakers from those who participated in the 8 annotated dialogues.

TRIPS uses messages to communicate between different components. At the time this evaluation was conducted, we did not have access to robust, large-coverage understanding components. When we conducted this evaluation we used the output from the TRIPS interpretation and problem solving components when they produced any output. When they did not produce output, we constructed the messages that would come from those components and fed them into TRIPS' central message-passing hub. These include messages indicating the syntax and semantics of user input, as well as problem solving-related communicative intentions. We were able to construct these messages because the semantic representation and abstract problem solving language used in the system are clearly defined in the shared system ontology. This evaluation is therefore a "best measure" of how the entire TRIPS dialogue system would perform in this domain.

We did not simulate the performance of any of the components of the generation system.

Because some input messages came slowly and one-by-one instead of in an asynchronous and overlapping fashion as would happen if the entire TRIPS system were actually running, the time-specific aspects of our generator were somewhat out of order. In particular, the system produced many more utterances explicitly performing turn-taking than it would in an actual interaction. We dealt with this partly by increasing the amount of time the system could wait before

speaking, and partly by filtering out by hand some of the extra turn-taking utterances.

If it seemed that a dialogue system would never form the intention to produce a particular utterance, that utterance would be left out. This happened in four situations. TRIPS would never produce an utterance that was self-talk, unintelligible or abandoned, as in the following (abandoned and self-talk):

Example 1

i believe we still have

TRIPS would also never initiate a discussion of the absence of a situation:

Example 2

but we have no reports of injuries or other problems at those areas

TRIPS would never discuss things requiring general world knowledge not in the domain:

Example 3

<laugh> pittsford alright i think i think the people in pittsford will win right they're paying more taxes but why are we fixing their roads i mean oh this is monroe county people that's right

Finally, some utterances simply would not be produced by TRIPS because it already automatically does the action being discussed or proposed:

Example 4

so i don't know what they are expecting of us but i <breath> i th- i don't really think we need to calculate how much time each bus is going to take

The randomness built into all stages of the generation process leads to variation in the output. Our method of dealing with this was to run each message through the system once and use the first output, unless we noticed a clear error in the output string or the performance of the system (e.g. bits of the semantic form in the string, a system crash). Errors in syntax, misleading referential expressions, information missing from the output and similar errors were not corrected. Suboptimal output is obvious from a simple read through the transcripts.

The dialogue extracts at the beginning of this paper were from a human-human dialogue. The extracts in figures 10 and 11 are the same sub-dialogues, except that one participant's contributions have been replaced by system-generated ones. These extracts come from the dialogues shown to our evaluators.

utt218 u i will summarize the plan
 utt219 u + so an + electric crew will take highway three ninety
 and highway five ninety from r g and e to the intersection
 of east avenue and highway three ninety
 utt220 s + okay +
 utt221 u + it will + take one point five hours
 utt222 s + yes +
 utt223 u and another electric crew will take highway three ninety
 and three eighty three to the intersection of the river
 and jefferson avenue
 utt224 u it will take one hour to move the sick people
 utt225 s um <SIL> one correction is i think <SIL> that
 <SIL> the uh you have the times reversed <SIL> oh
 no <SIL> you're right

Figure 10: First dialogue extract from evaluation dialogues

utt81 u and then <breath> sent <SIL> an ambulance to
 <SIL> <breath> from rochester <SIL> general
 utt82 s mm-hm
 utt83 s so there will be two walking-wounded people at culver
 road + still +
 utt84 u + right +
 utt85 s and also we will have three walking-wounded people at
 east main street still
 utt86 u that's right
 utt87 s police units can move to + there +
 utt88 u + um +
 utt89 what about <SIL> the uh <lipsmack> <SIL> the
 fire hydrant
 utt90 s police and the water crew could go to the fire + hydrant +
 utt91 u + okay +

Figure 11: Second dialogue extract from evaluation dialogues

4.3. Evaluation results

Teachers of English as a second language typically categorize the expertise of people they evaluate based on a set of tests, including written and oral examinations. There are established categorizations for oral proficiency (e.g. [Breiner-Sanders et al., 1999]), but all require that the examiner have the opportunity to converse with the test-taker, ask a range of questions that evaluate fluency in social and other interactions, and hear the responses.

The teacher of English as a second language, operating under the severe constraints imposed on her (she could not talk to the dialogue participants herself

or even hear the dialogues), gave evaluations of six speakers, including some for whom she had only one dialogue. She correctly identified the one non-native English speaker among the human participants; she incorrectly identified one other human participant as a non-native speaker of English and was unsure about a third. She identified the computer system as a non-native speaker of English. She estimated the level of fluency of all three “non-native” participants (two human, one computer) as high intermediate. The participant she was unsure about she rated advanced. The human who was really a non-native speaker of English received the most certain evaluation, and the most comments about behaviors indicating lack of fluency.

The behaviors noted by this evaluator as evidence of the lack of fluency of our system include: inconsistent use of interrogatives in questions, lack of contractions, and lack of variation in the use of adverbs indicating time progression (this same lack of variation appears in the human contributions replaced by the system, however). This evaluator also noted system behaviors indicating expertise in language use, among them the correct use of tense and the use of parenthetical phrases. Behaviors noted for the human speakers identified as non-native include: repetitions, repeated use of fillers, awkward sentence structure, indications of lack of knowledge about road construction, use of overly general or overly informal language (e.g. “road stuff”), and changes of topic in the middle of utterances.

Neither of our linguists identified the system as the most error-prone, uninformative or unnatural dialogue participant.

We received a variety of comments on the dialogues from our linguist evaluators, and have classified them into comments about speech acts, comments about grounding and turn-taking acts, comments about referential descriptions, and comments about discourse cues that can signal argumentation acts. We further classified the comments as positive, negative or neutral and broke them down by participant (human or system). Sometimes more than one comment was made for a particular utterance; we counted these as two comments. Sometimes a comment made for one utterance mentioned that the evaluator had observed the phenomenon in other places but was not going to mark every indication; we counted these as one comment.

In table 8 we give an approximation of the frequencies of positive and negative comments from our linguists, broken down by class of comment and type of speaker (system or human). In every category, the computer’s contributions received fewer negative comments than the humans’. However, this should not necessarily be taken at face value. For one thing, the fact that the system produced fewer incoherent utterances is not necessarily evidence of “naturalness” (although incoherence is probably not a type of naturalness one would want to duplicate). Also, the fact that the system had fewer inappropriate or ineffective referential descriptions in these simulated dialogues does not mean that many would not arise in actual human-computer dialogues, when the generation components rely solely on the Discourse Context for information about reference.

Speaker	Positive	Neutral	Negative
Speech acts			
human		5%	11%
computer		2%	2%
Turn-taking/grounding			
human		1%	7%
computer			6%
Descriptions			
human		1%	21%
computer			11%
Discourse cues			
human			6%
computer			2%
Other			
human		5%	13%
computer	1%	3%	3%

Table 8: Types of comments for different speakers

In the rest of this section, we discuss some of the specific comments we received. In each example, the utterance that was commented on is marked with ‘*’. Also, in these examples, if the speaker is ‘S’ then the utterance was constructed by the system.

4.3.1. Comments about speech acts

We received several types of comments about utterances that performed speech acts. Some utterances were marked syntactically incomplete, some were marked as being unclear, and some were marked incoherent (this last category does not include abandoned utterances).

Most of the utterances that were labeled ‘syntactically incomplete’ serve their communicative purpose, are clear in context, and seem perfectly natural. An example is:

Example 5

S 6 and there are seventy people at the end of highway two sixty one
7 the end of two sixty one is at the northwest of rochester
B 8 two sixty one okay
9* how many

Utterances with unclear semantics, on the other hand, are clearly awkward:

Example 6

S 56* so it takes half a hour for a helicopter to go in monroe county

Utterances marked as incoherent are either non-responsive or make no sense:

Example 7

- S 32 so + we need to + take the people there
 B 33 + <SIL> it's +
 34* except <SIL> except the guy who needs to <SIL> okay

4.3.2. Comments about acknowledgments

The comments about grounding and turn-taking acts fell into two categories: comments about ineffective acknowledgments, and comments about redundant acknowledgments. Ineffective acknowledgments are acknowledgments that seem not to fit the context, e.g.:

Example 8

- S 41 and also there is a bus at irondequoit police station
 B 42* yes

Some repeated acknowledgments were marked as redundant:

Example 9

- S 84 and they will return to irondequoit police station
 B 85* okay <SIL> yes

One evaluator also commented on other grounding and turn-taking acts without marking them as redundant. For example, she commented that in the following utterance “You can sort of hear them thinking out loud”:

Example 10

- S 65 um <SIL> okay

4.3.3. Comments about descriptions

We received quite a few comments about descriptions that were unclear (had no unique referent), incorrect, inefficient, or syntactically or semantically incomplete. Examples include the following; in each case, the description commented on is italicized:

Example 11: Unclear referent

- S 146* how long does it take for the road crew to fix *something*

Example 12: Semantically incomplete description

- A 183* the uh <SIL> then we'll send out <SIL> the <SIL> digger <SIL> the road crew and a second electrical <SIL> crew to monroe and highland <SIL> at *the second*

Example 13: Syntactically incomplete description

- A 27* also downed power lines at monroe and highland avenue <SIL> and route sixty five and route two fifty three <SIL> which is <SIL> one of the locations for + *water main break* +

n

4.3.4. *Comments about discourse cues*

One of our linguists made comments about the use of discourse cues that can signal argumentation acts. Two related to the use of “then”; the human participants in two of these dialogues sometimes used “then” to connect two utterances proposing or describing different actions, even when the actions were not in sequence. She also commented on a habit of the third human participant, who used “so we have” to mark summaries or restatements of locations.

This same evaluator disliked the system’s use of “and also” as synonymous with “and” (the evaluator thought the “also” was redundant). These comments are particularly helpful, because the appropriate use of discourse cues has an enormous effect on discourse coherence [Marcu, 1997].

5. Conclusions and future work

In this paper, we have explored ways to improve the naturalness of automatically generated spoken dialogue contributions for conversational agents that participate in complex, task-oriented dialogues. Our goal has been to explicitly track, in one framework, not only speech acts and higher-level discourse structure, but also the parts of the interaction that serve to maintain the participants’ collaboration, such as turn-taking and grounding behaviors.

We chose to adopt conversation acts theory [Traum and Hinkelman, 1992] as our model for this work. We implemented the key ideas of this theory in the context of generation components for the TRIPS system that plan individual conversation acts, perform surface generation, and organize conversation acts into coherent dialogue contributions.

We have conducted a language-based evaluation of our generation components, which use a model for dialogue based on conversation acts theory. We asked expert human judges to examine dialogues from the Monroe corpus that had been modified by substituting computer-produced contributions for the contributions of one participant. Our judges considered the system’s contributions to be at least as coherent, informative and robust as those of the human participants. Our system passes the “adequacy” test proposed by Jokinen [Jokinen, 1996].

Conversation acts theory works well as a model for producing grounding and speech acts. A key idea of the theory, that certain types of behavior can be subsumed by others, has proven extremely powerful in increasing the naturalness of system contributions. Also, since conversation acts theory helps clarify which intentions come from different levels of dialogue behavior, using it as a model for generation has enabled the production of dialogue components that are very flexible.

However, our examination of the other two levels of conversation acts theory has led us to identify some difficulties. In particular, turn-taking behavior does not fit well within an act-based framework. True, in order to perform an acknowledgment or produce a speech act it is necessary to take the turn. Also, there are

utterances (or parts of utterances) that cannot be explained as anything other than turn-taking. However, verbal turn-taking behavior is only a small part of the whole phenomenon, which also involves prosody, gaze, gesture and timing.

As a result of this work, we have identified several areas for future research. The first is a more in-depth study of multi-modal generation. We are interested in exploring how conversation acts theory needs to be extended to cover generation in other modalities, and in identifying aspects of multi-modal generation that may not fit well in an act-based framework.

We are also excited about the possibility of conducting a task-based evaluation of TRIPS that explores the effects of this new approach to generation. We plan to allow two modes of system operation: one in which the system produces only speech acts; and one in which it also models turn-taking, grounding and argumentation. Regardless of the mode, the system will act as an equal participant in the dialogue, taking initiative where appropriate. Each subject will interact with the system in both modes. Some subjects will have simple tasks, and others complex tasks such as those used for the Monroe corpus. We can then explore whether modeling more human-like conversation leads to faster or better solutions in complex and simple tasks.

References

- J. Allen, D. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent. An architecture for a generic dialogue shell. *Natural Language Engineering special issue on Best Practice in Spoken Language Dialogue Systems Engineering*, 6 (3), December 2000.
- J. Allen, D. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent. Towards conversational human-computer interaction. *AI Magazine*, 2001a. To appear.
- J. Allen and M. Core. Draft of DAMSL: Dialog Act Markup in Several Layers, 1997. available at: <http://www.cs.rochester.edu/research/cisd/resources/damsl/>.
- J. Allen, G. Ferguson, and A. Stent. An architecture for more realistic conversational systems. In *Proceedings of IUI'01*, Santa Fe, NM, 2001b.
- J. Allen and D. Litman. Discourse processing and commonsense plans. In P. Cohen, J. Morgan, and M. Pollack, editors, *Intentions in Communication*. MIT Press, 1990.
- E. Andre and T. Rist. Coping with temporal constraints in multimedia presentation planning. In *Proceedings of AAAI-96*, 1996.

- J. Antoine, J. Siroux, J. Caelen, J. Villaneau, J. Goulian, and M. Ahafhaf. Obtaining predictive results with an objective evaluation of spoken dialogue systems: Experiments with the DCR assessment paradigm. In *Proceedings of LREC'00*, pages 713–720, 2000.
- S. Bangalore, O. Rambow, and S. Whittaker. Evaluation metrics for generation. In *Proceedings of INLG'00*, 2000.
- K. Breiner-Sanders, P. Lowe, J. Miles, and E. Swender. ACTFL proficiency guidelines – speaking. *Foreign Language Annals*, 33(1), 1999.
- E. Brill. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP'92*, pages 152–155, Trento, IT, 1992.
- R. Brooks. Intelligence without representation. *Artificial Intelligence*, 47, 1991.
- M. Bull. *The timing and coordination of turn-taking*. PhD thesis, University of Edinburgh, 1998.
- G. Carenini. A task-based framework to evaluate evaluative arguments. In *Proceedings of INLG'00*, pages 9–16, 2000.
- J. Cassell, T. Bickmore, M. Billingham, L. Campbell, K. Chang, H. Vilhjalms-son, and H. Yan. Embodiment in conversational interfaces: Rea. In *Proceedings of CHI'99*, pages 520–527, Pittsburgh, PA, 1999.
- J. Cassell, M. Stone, and H. Yan. Coordination and context-dependence in the generation of embodied conversation. In *Proceedings of INLG'00*, pages 171–178, 2000.
- J. Cassell, O. Torres, and S. Prevost. Turn taking vs. discourse structure: How best to model multimodal conversation. In Y. Wilks, editor, *Machine Conversations*. Kluwer, 1998.
- A. Cawsey. Planning interactive explanations. *International Journal of Man-Machine Studies*, 38:169–199, 1993.
- J. Chu-Carroll and S. Carberry. Generating information-sharing subdialogues in expert-user consultation. In *Proceedings of IJCAI'95*, pages 1243–1250, 1995.
- J. Chu-Carroll and S. Carberry. Collaborative response generation in planning dialogues. *Computational Linguistics*, 24(3):355–400, 1998.
- H. Clark and E. Schaefer. Contributing to discourse. *Cognitive Science*, 13(2): 259–294, 1989.
- B. Grau and A. Vilnat. Cooperation in dialogue and discourse structure. In *Working Notes of the IJCAI 1997 Workshop on Collaboration, Cooperation and Conflict in Dialogue Systems*, 1997.

- B. Grosz and C. Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.
- J. Hirasawa, M. Nakano, T. Kawabata, and K. Aikawa. Effects of system barge-in responses on user impressions. In *Proceedings of Eurospeech '99*, 1999.
- E. Hovy. Automated discourse generation using discourse structure relations. *Artificial Intelligence*, 63(1-2):341–385, 1993.
- K. Jokinen. Adequacy and evaluation. In *Proceedings of the ECAI-96 Workshop, Gaps and Bridges: New Directions in Planning and NLG*, pages 105–107, 1996.
- A. Joshi and Y. Schabes. Tree-adjointing grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 69–124. Springer, 1997.
- D. Jurafsky, R. Bates, N. Coccaro, R. Martin, M. Meteer, K. Ries, E. Shriberg, A. Stolcke, P. Taylor, and C. Van Ess-Dykema. Switchboard discourse language modeling project report. Technical Report Research Note 30, Center for Speech and Language Processing, Johns Hopkins University, Baltimore, MD, 1998.
- J. Lester and B. Porter. Developing and empirically evaluating robust explanation generators: The KNIGHT experiments. *Computational Linguistics*, 23(1):65–101, 1997.
- D. Litman, S. Pan, and M. Walker. Evaluating response strategies in a web-based spoken dialogue agent. In *Proceedings of COLING-ACL '98*, pages 780–786, Montreal, Canada, 1998.
- K. Lochbaum. A collaborative planning model of intentional structure. *Computational Linguistics*, 24(4):525–572, December 1998.
- W. Mann and S. Thompson. Rhetorical structure theory: A theory of text organization. Technical Report RS-87-190, USC, Information Sciences Institute, 1987.
- D. Marcu. *The Rhetorical Parsing, Summarization and Generation of Natural Language Texts*. PhD thesis, University of Toronto, 1997.
- M. Maybury. Planning multimedia explanations using communicative acts. In M. Maybury, editor, *Intelligent Multimedia Interfaces*, pages 59–74. MIT Press, Menlo Park, CA, 1993.
- K. McKeown. *Text Generation: Using Discourse Strategies and Focus Constraints to Generate Natural Language Text*. Cambridge University Press, Cambridge, 1985.

- W. Minker. Evaluation methodologies for interactive speech systems. In *LREC'98*, pages 199–206, Granada, May 1998.
- J. Moore and C. Paris. Exploiting user feedback to compensate for the unreliability of user models. *UMUAI*, 2(4):331–365, 1992.
- D. Novick, B. Hansen, and K. Ward. Coordinating turn-taking with gaze. In *Proceedings of ICSLP'96*, 1996.
- D. O'Connell, S. Kowal, and E. Kaltenbacher. Turn-taking: a critical analysis of the research tradition. *Journal of Psycholinguistic Research*, 19:345–373, 1990.
- B. Oreström. *Turn-taking in English Conversation*. CWK Gleerup, 1983. Lund Studies in English: Number 66.
- M. Poesio and D. Traum. Conversational actions and discourse situations. *Computational Intelligence*, 13(3):309–347, 1997.
- R. Power. The organisation of purposeful dialogues. *Linguistics*, 17:107–152, 1979.
- E. Reiter. Has a consensus NL generation architecture appeared, and is it psycholinguistically plausible? In *Proceedings of the Seventh International Workshop on Natural Language Generation*, pages 163–170, 1994.
- N. Reithinger. The performance of an incremental generation component for multi-modal dialog contributions. In *Aspects of Automated Natural Language Generation*, pages 263–276. Springer-Verlag, Berlin, Germany, 1992.
- C. Rich and C. Sidner. COLLAGEN: A collaboration manager for software interface agents. Technical Report TR-97-21a, Mitsubishi Electric Research Laboratory (MERL), 1998.
- H. Sacks, E. Schegloff, and G. Jefferson. A simplest systematics for the organization of turn-taking in conversation. *Language*, 50(4):696–735, 1974.
- E. Schegloff. Recycled turn beginnings: A precise repair mechanism in conversation's turn-taking organisation. In G. Button and J. Lee, editors, *Talk and Social Organization*. Multilingual Matters Ltd., 1987.
- J. R. Searle. *Speech Acts*. Cambridge University Press, Cambridge, 1969.
- T. Sikorski and J. Allen. A task-based evaluation of the TRAINS-95 dialogue system. In *Proceedings of the ECAI Workshop on Dialogue Processing in Spoken Language Systems*, August 1996.
- A. Stent. Content planning in continuous-speech spoken dialog systems. In *Proceedings of the KI'99 workshop "May I Speak Freely?"*, 1999.

- A. Stent. The Monroe Corpus. Technical Report TR728 and TN99-2, University of Rochester Computer Science Department, 2000a.
- A. Stent. Rhetorical structure in dialog. In *Proceedings of INLG'00*, 2000b. Student paper.
- A. Stent. *Dialogue Systems as Conversational Partners: Applying conversation acts theory to natural language generation for task-oriented mixed-initiative spoken dialogue*. PhD thesis, Computer Science Dept., University of Rochester, 2001.
- A. Stent and J. Allen. TRAINS-96 system evaluation. Technical Report TN97-1, University of Rochester Computer Science Department, 1997.
- A. Stent, J. Dowding, J. Gawron, E. Owen Bratt, and R. Moore. The CommandTalk spoken dialogue system. In *Proceedings of ACL'99*, 1999.
- M. Stone and C. Doran. Sentence planning as description using tree adjoining grammar. In *Proceedings of ACL'97*, pages 198–205, 1997.
- D. Traum and E. Hinkelman. Conversation acts in task-oriented spoken dialogue. *Computational Intelligence*, 8(3):575–599, 1992.
- David Traum and James Allen. Discourse obligations in dialogue processing. In *Proceedings of ACL'94*, pages 1–8, 1994.
- M. Walker, D. Litman, C. Kamm, and A. Abella. Evaluating spoken dialogue agents with PARADISE: Two case studies. *Computer Speech and Language*, 12(3), 1998.
- XTAG Research Group. A lexicalized tree adjoining grammar for English. Technical Report 95–03, The Institute for Research in Cognitive Science, University of Pennsylvania, 1995.