

Chapter

Reconciling Control and Discourse Structure

Susan E. Strayer, Peter A. Heeman and Fan Yang

*Department of Computer Science and Engineering, OGI School of Science and Engineering,
Oregon Health & Science University, Beaverton OR, USA*

Abstract: In this paper we consider how control (initiative) is managed in task-oriented dialogues. We propose that control is subordinate to discourse structure. The initiator of a discourse structure segment has control for the entire segment, except occasionally when the non-initiator takes control briefly, then control generally reverts immediately back to the segment initiator, or a new block begins. In analyzing dialogues from the TRAINS corpus we find that inside a segment initiated by one speaker, the other speaker only makes two types of contributions: *collaborative completions*, in which the non-initiator helps the segment initiator achieve their goal of completing an utterance, and short contributions to the discourse segment purpose. The proposal has important implications for dialogue management: a system only needs to model intentional structure, from which control follows.

Key words: discourse structure, mixed initiative, dialogue, subdialogues, acknowledgements

1. INTRODUCTION

The dialogue manager of a spoken language system is responsible for determining what contributions a system can make and when it can make them. The question is, what should the dialogue manager pay attention to in order to accomplish this? Two areas of research have shaped our understanding of what happens in dialogue: research in dialogue structure and in mixed initiative.

Grosz and Sidner (1986) proposed a theory of discourse structure to account for why an utterance was said and what was meant by it. Their theory had three components: linguistic structure, intentional structure and atten-

tional state. *Intentions* are key to accounting for discourse structure, defining discourse coherence, and “providing a coherent conceptualization of the term ‘discourse’ itself.” The *intentional structure* describes the purpose of the discourse as a whole, and the relationship of the purpose of each discourse segment to the main discourse purpose or other discourse segment purposes. All utterances within a segment contribute to the purpose of that segment. This theory, however, does not comment on control (initiative) within the segment. Nor does it specify when and how speakers should start a segment or end the current one. Hence, it underspecifies what speakers can do in dialogue.

Research in control (initiative) works to account for which speaker is driving the conversation at any given point. For example, in a question-answer pair, the speaker asking the question is said to have control (Whittaker and Stenton, 1988; Walker and Whittaker, 1990; Novick and Sutton, 1997). Whittaker and Stenton segmented dialogues at points where control shifts from one speaker to the other. They found that control “did not alternate from speaker to speaker on a turn by turn basis, but that there were long sequences of turns in which control remained with one speaker.” In a mixed initiative system, the dialogue manager needs to track initiative in order to know when the system should add significant content, and when it should let the user take over. However, no theory has offered a good account of why a speaker would want to take control, or keep it once they have it.

In the rest of this paper we first describe previous work in discourse structure and in control and describe our coding of them. Next, we explore the relationship between discourse structure and control. As previous studies have found (Whittaker and Stenton, 1988; Walker and Whittaker, 1990), there is a close correlation between them, but the relationship is not direct. We then explore how control can shift within a subdialogue and find two types of contributions that a speaker can make in a discourse segment: *collaborative completions*, in which the non-initiator helps the segment initiator achieve their goal of completing an utterance¹ (Linell, 1998), and short contributions that add to the discourse segment purpose. We propose that control is subordinate to intentional structure. Additionally, our proposal is better able to account for question-answer pairs and how control returns to the original speaker after an embedded subdialogue. It will have important implications for dialogue management: a system only needs to model intentional structure, from which control follows.

2. DISCOURSE STRUCTURE AND CONTROL ANALYSIS

Our proposal for managing control builds on two main areas of research, discourse structure and control (initiative). We start by discussing the work of Grosz and Sidner (1986), which ties speaker's intentions to linguistic structure, then discuss the work of Whittaker, et al. in initiative. We introduce our coding of eight dialogues totaling 45 minutes from the TRAINS corpus, a corpus of human-human task-oriented dialogues, in which two participants work together to formulate a plan involving the manufacture and transportation of goods (Allen et al., 1995; Heeman and Allen, 1995). In these dialogues, one speaker, the user (u), has a goal to solve, and the other speaker, the system (s), knows the detailed information involved in how long it takes to ship and manufacture goods. We also briefly describe the annotation tool developed at OGI that we are using to code the dialogues.

2.1 Discourse Structure

Discourse structure is used to analyze dialogue from the top down, starting with the purpose of the discourse as a whole, then the purpose of each discourse segment, in order to understand how each utterance fits into the dialogue. The theory of discourse structure developed by Grosz and Sidner (1986) proposes that discourse structure is made up of three components: linguistic structure, intentional structure, and attentional state. Our work focuses on the first two components. The *linguistic structure* is a hierarchical segmentation of the dialogue into *discourse segments*. Segment boundaries are identified by changes in tense and aspect, pause lengths, speech rate, and discourse markers, such as “anyway”, “by the way”, “so”, and “first of all”. The *intentional structure* is a hierarchy of segment purposes. Each discourse segment has a purpose, and the purpose of each segment contributes to the purpose of its parent. Intentional structure is key to understanding what the discourse is about and explains its coherency.

Subdialogue coding: A number of schemes have been proposed for coding discourse structure (Nakatani et al., 1995; Passonneau and Litman, 1997; Flammia, 1998; Traum and Nakatani, 1999; Nakatani and Traum, 1999) ranging from coding flat segmentation of monologues to hierarchical segmentation of dialogues. Our basis for segmenting discourse has drawn from a number of these. We coded dialogue games (Isard and Carletta, 1992), such as question-answer pairs, although we did not code statement-acknowledgement or statement-agreement pairs, which do not have the same obligation to respond as question-answer pairs. Once the dialogue games were coded, we addressed higher level segmentation based on coherence,

utterances that addressed the same topic. In the TRAINS corpus, the primary purpose of the dialogue is to develop a plan, so most of the subdialogues were about the planning process, such as, relate the goal of the plan, construct steps in the plan, modify the plan, summarize the plan and evaluate the plan, as Traum and Hinkelman (1992) and Smith and Gordon (1997) had described.

In deciding if a group of utterances formed a discourse segment, we took into account whether there existed a *shared* discourse segment purpose. If the speaker had not made the purpose of the discourse segment clear, then we did not consider it a discourse segment. For example, in one dialogue, a user asked a series of questions which apparently related to an evolving plan, but, because he never told the system what the goal of the plan was or even the purpose of the questions, we did not consider them part of a discourse segment (TRAINS dialogue d93-23.1). In the other extreme, the speaker stated the goal of the discourse segment explicitly. For example, a user prefaced discussion of the plan relating to moving bananas by saying ‘First, let’s deal with the boxcar full of bananas’, then discussion of how to move oranges with ‘Okay, now let’s deal with the boxcar of oranges’, and finally, discussion of how to move orange juice with ‘Now we still have to deal with our tanker of orange juice’ (dialogue d93-22.2). In another example, the speaker began a stretch that summarized a plan by saying ‘Let’s recap this’ (dialogue d92a-5.2). In all of these cases, a discourse segment was started clearly by an explicit statement of what the goal of the segment was. However, most cases fall somewhere in between. The speaker may or may not explicitly state a goal, but they may signal a new topic with transitional phrases, such as ‘now’ (e.g., ‘Now I want to send them back to Corning’) and ‘and then’ (e.g., ‘And then at the same time we’ll use that other engine’). In the Trains dialogues, the instructions clearly indicate that the user is in charge. Hence, we always coded a discourse segment, initiated by the user, that surrounded the entire dialogue, except for the initial greeting, and the closing.

Figure 1 shows a dialogue excerpt. The left side shows the discourse structure and the segment initiator for each subdialogue. The *segment initiator* gives the first utterance in the segment and establishes its purpose.² The dialogue excerpt is the beginning of a discourse segment with the user as the initiator. It has three subdialogues, two of which were also initiated by the user. The other was initiated by the system, and it has another subdialogue embedded in it, which also was initiated by the system, as the same utterance serves as the first utterance for both segments. Generally, our coding scheme yielded a dialogue structure that is rather flat, with rarely more than 3 levels of embedding.

Segment Initiator	Speaker	Utterance	Control
u	u	u39 uh I need to ship a boxcar of bananas to Corning	u
	u	u40 um	
	u	u41 okay how long would it take to take the engine from Avon to Dansville and also from Avon to Bath?	
	s	u42 uh three hours from Avon to Dansville and four hours from Avon to Bath	
	u	u43 okay	
	u	u44 let's take the engine from Avon to Dansville	
	u	u45 pick up a boxcar	
	u	u46 bring it back to Avon to pick up bananas	
	u	u47 and then deliver it to Corning	
	s	u48 alright	
s	s	u49 would you like to know how long that takes?	s
	u	u50 yes	
	s	u51 uh	
	s	u52 by the shortest route it would take eleven hours	
	u	u53 okay and how long was the trip from Avon to Dansville and back to Avon?	
u	u	u54 three hours each way	u
	s	u55 three hours each way	
	u	u56 okay	
	u	u57 so that won't work	

Figure 1. Example of Discourse Structure and Control Segments (d93-22.2)

Two coders conducted consensus coding for six of the dialogues, by annotating the dialogues independently, then comparing the two annotations side by side and resolving differences. Most of the differences between the two coders were resolved successfully, though there continued to be differences in higher level task-related coding, reflecting the ambiguity of our coding scheme in this area.

Two of the dialogues were coded independently after achieving consensus on speech repairs and utterance boundaries. Intercoder reliability for the two dialogues coded independently showed 43 hits, where discourse segments for both coders were identical in extent and segment initiator³. There were 26 misses, where the first coder coded blocks not included by the second coder, and 39 false positives, where the second coder coded blocks not included by the first coder. These figures yield a recall rate of 62% and a precision rate of 52%.⁴ In examining the missing and false positive blocks, we found that 13 of the misses corresponded very closely to 13 of the false positives, differing by only one utterance in their beginning or end, and always having the same initiator. If we include these as hits, our recall and precision rates increase to 81% and 68% respectively. We further examined

the blocks that were scored as misses and false positives. Similar to parser evaluations (Harrison et al., 1991), we looked at *crossing segments*, where a segment from one coder overlaps with a segment from the other coder, but neither is properly contained in the other. We will refer to such segments as being *inconsistent* with the other coder's segmentation. Obviously, inconsistent segments indicate major disagreement about the structure of the discourse. Only 3 of the 26 misses and 2 of the 39 false positives were inconsistent with the other annotator's segmentation.

2.2 Control

Control (initiative) is held by the speaker who is driving the conversation at any given point in the conversation (Whittaker and Stenton, 1988; Walker and Whittaker, 1990; Novick and Sutton, 1997). It has been used to analyze discourse from the bottom up, starting with utterances. We start with *adjacency pairs* (Schegloff and Sacks, 1973), which consist of a *first part*, uttered by one of the speakers, and a *second part*, uttered by the other. The first part sets up expectations for the second part, and hence the speaker of the first part can be viewed as being in control of the dialogue during both parts of the adjacency pair. Below we give the annotation scheme used by Whittaker, et al. (Whittaker and Stenton, 1988; Walker and Whittaker, 1990) for annotating control based on utterance type.

- **Assertions:** Declarative utterances used to state facts. The speaker has control, except when it is a response to a question.
- **Questions:** Utterances intended to elicit information from others. The speaker has control, except when it follows a question or command.
- **Commands:** Intended to induce actions in others. The speaker has control.
- **Prompts:** Utterances with no propositional content (e.g., “yeah,” “okay”). These utterances do not demonstrate control.

On the right side of Figure 1, we indicate which utterances demonstrate control. For utterances where the speaker does not demonstrate control, control is said to belong to the last speaker that demonstrated it. Hence, when the system uttered “mm-hm” in the second utterance, which does not demonstrate control, control does not change from the user.

Whittaker and Stenton used the control codings as a basis for segmenting dialogues between an expert and a client about diagnosing and repairing software faults. They found that not only did control pass back and forth between the speakers (unlike single-control dialogues), but that control often stayed with a speaker for an average of eight speaker turns. The right side of

Figure 1 also shows the control segments. Here, we see that control swings back and forth between the system and the user several times, leading to three control segments.

Whittaker and Stenton (1988) looked at the correlation of control boundaries to discourse markers, and Walker and Whittaker (1990) looked at anaphoric reference. These are the same kinds of linguistic evidence that Grosz and Sidner (1986) said marks discourse segment boundaries. In fact, Walker and Whittaker claimed that control segments are the discourse segments of Grosz and Sidner's theory, with the speaker with control being the initiator of the segment, who establishes the discourse segment purpose. However, they acknowledged that "there can be topic shifts without change of initiation, change of control without topic shift". When we look at the dialogue excerpt given in Figure 1, we see that the control segmentation identified the second subdialogue, but not the first. Furthermore, control doesn't detect when the embedded subdialogue within the second subdialogue ends, in the same way it fails to detect the end of the first and last subdialogues. Hence, the exact relationship between control and discourse structure has not been explained.

Control coding: We tagged utterances with codes based on the DAMSL coding scheme (Core and Allen, 1997), and coded utterance tags for forward functions (such as, Statement and Info-Request), backward functions (such as Answer, Agreement, Acknowledgement and Stalls). Figure 2 outlines the DAMSL scheme and indicates which utterance tags demonstrate control. We found that in general forward functions mapped to utterances demonstrating control in Whittaker and Stenton's (1988) scheme and backward functions and Stalls mapped to utterances not demonstrating control. A notable exception is the DAMSL Completions, which are classified as a backward looking function, We consider Completions as demonstrating control, because they contribute something new to the discourse which hadn't been explicitly introduced by the other speaker. When applying the Whittaker coding scheme, they typically were coded as Assertions, which demonstrate control.

For annotating utterance tags, we followed the same methodology as for discourse segmentation. The first two authors individually annotated six dialogues; we then compared our annotations and resolved most of our differences. After refining our annotation scheme during consensus coding of the first six dialogues, we coded two dialogues independently. For inter-coder reliability, we are interested whether we agreed as to whether an utterance demonstrated control or not. Using our DAMSL scheme and the mapping given in Figure 2, we found that our agreement on the two dialogues as to which utterances show control was 92%.

TAG	CONTROL	DESCRIPTION
Forward Looking Functions		
Statement	Speaker	Makes a claim about the world
Info-request	Speaker	Questions
Check	Speaker	Solicits confirmation from the other speaker, such as tag questions (Ex: 'right?')
Suggestion	Speaker	Suggests, proposes or promises a future action
Opening	Speaker	Greetings and other conversation openers
Backward Looking Functions		
Agreement	-	Indicates response to a claim or proposal
Answer	-	Complies to an Info-request
Acknowledgements	-	Signals that the previous utterance was understood
Completions	Speaker	Shows understanding by finishing or adding to a clause the other speaker is in the middle of constructing
Other		
Stalls	-	Holds the turn while the speaker forms a thought (Ex: 'um')
Communicative Status		
Abandoned	-	The speaker abandons their utterance and the other speaker takes over
Incomplete	-	An incomplete utterance that the other speaker understands and responds to

Figure 2. Utterance Coding Scheme

2.3 Annotation Tools

DialogueView, the annotation tool we are currently using to code dialogues (Heeman, Yang and Strayer, 2002), allows us to code dialogues at several levels. The word level allows disfluences, abandoned and uninterruptible bits of speech to be coded, and allows utterances to be segmented. The utterance level abstracts out disfluencies and overlapping speech, and allows us to tag utterances. The subdialogue level abstracts out utterances and considers the relations of subdialogues to each other. The tool allows us to view and hear speech at any of these levels, which makes it easier to code more dialogues and establish inter-coder reliability. It also allows us to analyze the relationship among the various layers of a dialogue programmatically, rather than by hand, as we did in our initial coding of the dialogues.

We used ACT, our Annotation Comparison Tool (Yang, Heeman and Strayer, 2002) to facilitate consensus coding of speech repairs, utterance boundaries and utterance tags. We used DialogueView for consensus coding of discourse segment boundaries and to run preliminary statistics comparing control, segment initiator and discourse boundaries and inter-coder reliability.

3. RELATIONSHIP BETWEEN CONTROL AND DISCOURSE STRUCTURE

Walker and Whittaker (1990) suggested that changes in control correspond to changes in discourse structure, but they did not determine the exact relationship between them. In this section we analyze the differences between control segments and discourse structure for eight dialogues from the TRAINS corpus. We find that there is a close relationship, but not a direct one.

3.1 Segment Boundary Comparison

In this section, we compare control boundaries (where control shifts from one speaker to the next) to subdialogue boundaries (where a new subdialogue begins, or where an embedded subdialogue ends and its parent continues) using recall and precision. A control boundary is scored as a hit if there is a corresponding subdialogue boundary. It is scored as a false positive if there is no subdialogue boundary. A subdialogue boundary is scored as a miss if it has no control boundary. For example, in Figure 1, there are two hits (the boundaries after utterances u48 and u52), three misses (the boundaries after utterances u40, u43 and u50), and no false positives. The second column of Table 1, “Control vs Subdialogue”, gives the results.⁵ We see that both recall and precision are very low for control boundaries relative to discourse boundaries. However, comparing control segments to discourse segments is not fair. The misses in Figure 1 should be expected since the initiator of the last subdialogue is the same as the higher level subdialogue.

To show the extent of the unfairness, we contrast changes in segment initiator to discourse segment boundaries in the third column of Table 1, “Segment Initiator vs Subdialogue”. Not surprisingly, we obtained a precision of 100%: by definition, the segment initiator is only set at the beginning of each discourse segment. However, we only obtained a recall rate of 40%.

Table 1. Correlation of Segment Boundaries

	Control vs Subdialogue	Segment Initiator vs Subdialogue	Control vs Segment Initiator
Subdialogue Boundaries	249	249	102
Hits	102	100	91
Misses	147	149	11
False Positives	56	0	71
Recall	41%	40%	89%
Precision	65%	100%	56%

This means only 40% of discourse segment boundaries are initiated by a different speaker. We should not expect these boundaries to have a change in control, since there is no change in segment initiator. A fair comparison should contrast changes in control only to changes in discourse segment initiator. The results of doing this is shown in the fourth column, “Control vs Segment Initiator”. Here we see much better results, but there is far from perfect agreement.

3.2 Shifts Within Discourse Segments

Table 1 indicated that changes in discourse segment initiator do not always match changes in control. To better determine what is happening, we looked at control inside of discourse segments. We asked whether the segment initiator has control for the first utterance in a segment, and whether they kept for the whole subdialogue. For the excerpt in Figure 1, this was the case for all of the subdialogues. The results for all eight dialogues are given in Table 2. Here, we see that the segment initiator always has control for the first utterance of the subdialogue. This is not unexpected, since the speaker needs to contribute something new, otherwise it would not count as the beginning of a new discourse segment. However, control does not always stay with the initiator for the entire segment, as seen in the last row of Table 2.

Table 2. Control Held by Segment Initiator⁶

	Number	(%)
Subdialogues	194	
First utterance	194	(100%)
Whole subdialogue	161	(83%)

4. RECONCILING CONTROL INSIDE DISCOURSE SEGMENTS

The first utterance of each discourse segment shows perfect agreement between the initiator of the segment and speaker with control, as seen in Table 2. But what happens during the course of the segment? In this section we focus on subdialogues where the non-initiator makes a contribution and the initiator finishes the segment.

4.1 Collaborative Completions

In dialogue there are times when two speakers are so synchronized with each other that they say almost the same thing at the same time, and complete each other's utterances and thoughts (Linell, 1998). These *collaborative completions* indicate understanding, help the segment initiator achieve their goal of completing an utterance, and move the conversation forward. Figure 3 gives an example, where the system filled in "by noon" for the user, before the user finished their utterance. Although in the DAMSL coding scheme, Completions are coded as backward-looking functions, we code collaborative completions as demonstrating control, because they add content that wasn't requested by the other participant. However, this control is subordinate to the control of the main segment.

Segment			
Initiator	Control	Speaker	Utterance
u	u	u	okay so we have to take oranges from Corning and bring them to Elmira
		s	right
	u	u	and then back to Bath by . . .
s	s	s	. . . by noon
	u	u	by noon

Figure 3. Collaborative completion (d92a-5.2)
(Marked with dashed lines)

Segment	Initiator	Control	Speaker	Utterance
	u	u	u	and then go to Dansville
	s	s	s	and that's one more hour
			u	yeah
		u	u	we can . . .
	s	s	s	. . . drop off at the . . .
		u	u	drop off that boxcar
		u	u	drop off the boxcar of . . .
	s	s	s	and then take two empty ones
			u	right
		u	u	two empty ones down to to Avon
		u	u	and pick up the the bananas
			s	right

Figure 4. Other contribution (d93-19.5)
(Marked with dashed lines)

4.2 Co-contributions

The rest of the utterances coded with control made by the non-initiator of the segment were more substantial contributions. Here, the speaker added content that contributed to the discourse segment purpose that is not predicted from the initiator's speech. We refer to these as *co-contributions*, and they often occur where the two speakers are closely collaborating and are highly synchronized. In Figure 4, we show a dialogue excerpt in which the two speakers are so closely synchronized that they pick up parts of each others utterances and build on it. Control shifts back and forth between the two speakers, but, in fact, we think this phenomenon of co-contributions is related to the phenomena that Schiffrin (1987) referred to as *shared turns*, which provide a cooperative framework for building some stretches of discourse, rather than one based on control.

4.3 Effect on Control

Table 3 shows what happens to control after a speaker makes a contribution demonstrating control. Here, we are restricting ourselves to what happens in the current discourse segment. From previous theories of control (Walker and Whittaker, 1990; Chu-Carroll and Brown, 1997), we expect control to stay with the last speaker who demonstrates control. However, as Table 3 shows, what happens to control depends on whether the speaker making the contribution is the segment initiator or the other speaker.

Table 3. Changes in Control

	After Contributions by Segment Initiator	After Contributions by Non-initiator		
		Total	Co-contri- butions	Collaborative Completions
Contributions (number)	539	60	49	11
New subdialogue	150	14	13	1
Embedded subdialogue	74	6	3	3
Current speaker continues	276	17	16	1
Switch to other speaker	39	23	17	6

As a baseline, we looked at what happens after the segment initiator makes an utterance that demonstrates control. First, we factor out those cases in which the segment ends or there is an embedded subdialogue before control is next demonstrated. What we find is that in 276 cases, the next speaker who demonstrates control continues to be the segment initiator. In only 39 cases, or 12% of the time, did control change to the non-initiator. This is in fact what control theory predicts, that control tends to stay with the last person in control. We next look at what happens after the non-initiator makes an utterance that demonstrates control. Here, we see that the non-initiator continues with control in 17 cases, and control switches to the initiator in 23 cases, or 58% of the time. This is not what control theory predicts, and stands in strong contrast to the results to what happens after the initiator has control.

We also looked at what happens for the two types of utterances that the non-initiator can make that show control. For collaborative completions, the non-initiator continued control in just 1 of 7 cases. For co-contributions, the non-initiator continues in 17 of 33 cases, for a rate of 52%. The difference between collaborative completions and co-contributions might reflect the level of control that the speaker is demonstrating. In making a collaborative completion, the speaking is taking control in helping the other speaking finish their utterance, rather than in helping them achieve the discourse segment purpose.

4.4 Discussion

Collaborative completions and co-contributions are exceptions to the general rule that control tends to reside with the same speaker. Based on the results of our study, we propose that control is subordinate to the intentional structure of the discourse theory of Grosz and Sidner. Control is held by the segment initiator. The non-initiator can make utterances that contribute to the purpose of the current discourse segment, namely collaborative comple-

tions and co-contributions, but control remains with the segment initiator. Hence, control does not need to be tracked, because it is held by the initiator of the discourse segment.

Our proposal accounts for how either speaker can contribute to the purpose of a discourse segment. However, if the non-initiator contributes, this doesn't mean that he or she has taken over control of the dialogue. Rather, control remains with the initiator. This can also account for embedded subdialogues, which according to the theory of Grosz and Sidner, should contribute to the purpose of their parent subdialogue. Whoever initiates the subdialogue has control of it, but as soon as it finishes, control reverts to the initiator of the embedding segment. For example, in Figure 1, the first subdialogue initiated by the user (u) contributes to the purpose of the parent subdialogue by checking how long an action will take. The next subdialogue initiated by the system (s) contributes to the purpose of the parent by ensuring that the user knows how long the next action will take. Both purposes contribute to the overall goal of moving goods within the time constraints given in the problem. Furthermore, consistent with our proposal, control returns to the segment initiator of the parent subdialogue after the embedded subdialogue ends.

We can extend the above view of subdialogues to better code answers to questions. Although some answers are a single utterance, other answers can take several utterances to answer. In one example from our corpus, in a segment initiated by the user, the user asks the system what the current plan is. The system's answer was a summary of the plan, which took a number of utterances by the system, with the user interspersing acknowledgments. The answer should in fact be coded as a subdialogue, in which the system is the initiator and has control. Of course, this is a special type of subdialogue in which the speaker is given control explicitly by the other speaker to make the answer.

This approach can also handle special dialogue games that speakers innovate between themselves. One example was a command-response game that two speakers seemed to have developed that started by the user saying "I want" and giving a goal, to which the system responded by detailing a set of actions to accomplish the goal and the time to complete them (d93-23.1). When we coded these originally, we coded the system's responses as forward-looking Statements or Suggestions. However, when we coded the discourse segments, it became clear that the system had a standard response to the user's convention that started with "I want". In this case, the control coding is misleading, because it seems like the system is taking more initiative than it has. This accounted for 9 of the 16 utterances where we found control to continue with the non-initiator. It might be more appropriate to view this as a dialogue game with the system's response in a *second-part* discourse

segment, just as we proposed for analyzing answers in the previous paragraph.

Other researchers have struggled with structure in control. Chu-Carroll and Brown (1997) referred to control as *dialogue initiative*, and proposed a second level, *task initiative*, to model who is adding domain actions. In contrast to our proposal, which makes control subordinate to intentional structure, they proposed that dialogue initiative is subordinate to their task initiative. Hence, their model could incorrectly predict who has control after the non-initiator makes a contribution. As was shown in Table 3, after the non-initiator makes an other-contribution within a subdialogue, generally control returns to the segment initiator of the subdialogue, instead of staying with the non-initiator. Chu-Carroll and Brown also used task initiative to model how cooperative a system should be. With novice users, the system would tend to have task initiative and thus make domain actions, but not so with experts. This is similar to Smith and Gordon's (1997) use of four levels of control, which set how much control was given to the system and how much was given to the user in one of four levels. Although a system needs to reason about how helpful it needs to be, it is unclear whether this can be done through a single variable that is tied to dialogue control.

There are several major issues that need to be addressed before discourse segmentation can fully replace control annotation. One is a thorough understanding of the types of contributions a speaker can make in the second part of a dialogue game. Some answers to questions were extended and looked like a series of statements until the discourse segments were blocked out and it became clear that they were subordinate to a request for information from the other speaker. The question is, how much initiative can a speaker show when answering a question? What are the constraints on how much they can contribute within the second part to a dialogue game? Another issue that needs to be addressed is a thorough understanding of what constitutes a discourse segment. Although dialogue games such as question-answer pairs are not problematic, higher level segmentation involving dialogue transactions, such as planning, summarizing and revising a plan, and rhetorical structure that describes the relationship between segments are more difficult to determine. Finally, although this study has looked closely at what sort of utterances can occur within a discourse segment, we have not yet considered what sort of subdialogues can occur within a segment, who can initiate a new subdialogue, or how to transition to the new subdialogue.

5. CONCLUSION

In this paper, we have proposed that control is subordinate to intentional structure in dialogue. We have backed up this claim by examining utterances that demonstrate control made by the non-initiator of the discourse segment. We found that after these utterances, control returns to the segment initiator most cases. The reconciliation of control and discourse segments means that we now understand how control and dialogue level intentions are related and have a clearer picture of how both participants can contribute to discourse intentions.

Based on our results, control in itself does not need to be tracked. Control belongs to the speaker who started the current discourse segment. Therefore, it's not necessary to model control when developing a dialogue manager. A dialogue manager only needs to model intentional structure.

6. FUTURE WORK

In the future we plan to further analyze what happens to control after the non-initiator demonstrates control. For instance, we want to better understand the rare cases where control stays with the non-initiator. We also plan to investigate what happens with embedded subdialogues. Who tends to initiate them? What happens to control after they finish? In addition to more TRAINS dialogues, we will code dialogues from other corpora, such as MapTask (Anderson et al., 1991) and Switchboard (Godfrey et al., 1992). This will help ensure that we do not introduce idiosyncrasies of the TRAINS corpus into our theory.

We will further refine our discourse segment coding scheme in order to improve our intercoder reliability, including better understanding of what a discourse segment is. We plan to explore how rhetorical structure theory may relate to discourse structure in dialogue (Stent, 2000).

Our theory necessitates that we better understand the structure of discourse, how it is built, and the actions and rules that a discourse manager can use to affect the discourse structure. We also need to understand the reasoning process that determines whether a participant will make a co-contribution or start a new subdialogue. Since dialogue is a collaborative effort (Cohen and Levesque, 1994; Clark and Wilkes-Gibbs, 1986), we also need to explore how the participants collaborate on the discourse structure.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge funding from the Intel Research Council. The authors also thank David Traum and members of the Centers for Spoken Language Understanding and Human Computer Communication at OGI for helpful discussion and comments.

NOTES

- 1 In the DAMSL coding scheme, these utterances are tagged as *Completions*, a type of utterance that signals understanding (Allen and Core, 1997).
- 2 The segment initiator corresponds to the *initiating conversational participant* (ICP) of Grosz and Sidner's theory. The non-initiator corresponds to the *other conversational participant* (OCP).
- 3 If two segments only differed on whether they included utterances tagged as Acknowledgements or Stalls, we counted them as hits.
- 4 $\text{Recall} = \text{Hits} / (\text{Hits} + \text{Misses})$
 $\text{Precision} = \text{Hits} / (\text{Hits} + \text{False Positives})$
- 5 The results reported here are consistent with the results of our preliminary study (Strayer and Heeman, 2001), which had analyzed half the data that serves as the basis for this report.
- 6 We no longer distinguish between task and clarification subdialogues, as we did in our preliminary study (Strayer and Heeman, 2001). We are in the process of developing a more complete taxonomy of discourse segment types, including dialogue games, transactions and asides, but have not finalized it yet.
- 7 The ellipses (...) after the user's utterance 'and then back to Bath by ...' indicate the utterance was tagged as an Incomplete. The ellipses before the system's utterance '... by noon' indicates the utterance was tagged as a Complete.

REFERENCES

- J. Allen, L. Schubert, G. Ferguson, P. Heeman, C. Hwang, T. Kato, M. Light, N. Martin, B. Miller, M. Poesio, and D. Traum. 1995. The Trains project: A case study in building a conversational planning agent. *Journal of Experimental and Theoretical AI*, 7:7-48.
- A. Anderson, M. Bader, E. Bard, E. Boyle, G. Doherty, S. Garrod, S. Isard, J. Kowtko, J. McAllister, J. Miller, C. Sotillo, H. Thompson, and R. Weinert. 1991. The HCRC map task corpus. *Language and Speech*, 34(4):351-366.
- J. Chu-Carroll and M. Brown. 1997. Tracking initiative in collaborative dialogue interaction. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*.
- H. Clark and D. Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22:1-39.

- P. Cohen and H. Levesque. 1994. Preliminaries to a collaborative model. *Speech Communication*, 15(3-4): 265-274, December.
- M. Core and J. Allen. 1997. Coding dialogs with the DAMSL annotation scheme. In *Working notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*.
- G. Flammia. 1998. Discourse segmentation of spoken dialogue: an empirical approach. Doctoral dissertation, Department of Electrical and Computer Science, Massachusetts Institute of Technology.
- J. Godfrey, E. Holliman, and J. McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of the International Conference on Audio, Speech and Signal Processing (ICASSP)*, pages 517-520.
- B. Grosz and C. Sidner. 1986. Attention, intentions and the structure of discourse. *Computational Linguistics*, 12(3): 175-204.
- P. Harrison, S. Abney, E. Black, D. Flickinger, C. Gdaniec, R. Grishman, D. Hindle, B. Ingria, M. Marcus, B. Santorini, and T. Strzalkowski. 1991. Evaluating syntax performance of parser/grammars of English. In *Proceedings of the Workshop on Evaluating Natural Language Processing Systems*, 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, CA, pages 71-77.
- P. Heeman and J. Allen. 1995. The Trains spoken dialog corpus. CD-ROM, Linguistics Data Consortium, April.
- P. Heeman, F. Yang and S. Strayer. 2002. DialogueView: A Dialogue Annotation Tool. In *Proceeding of 3rd SIGDial workshop on discourse and dialogue*.
- A. Isaard and J. Carletta. 1995. Transaction and action coding in the MapTask Corpus Research Paper HCRC/RP-65
- P. Linell. 1998. Approaching dialogue: Talk, interaction and contexts in dialogical perspectives. John Benjamins Publishing Company, Amsterdam.
- C. Nakatani, B. Grosz, D. Ahn, and J. Hirschberg. 1995. Instructions for annotating discourse. Technical Report 21-95, Center for Research in Computing Technology, Harvard University, Cambridge MA, September.
- C. Nakatani and D. Traum. 1999 Coding discourse structure in dialogue (version 1.0) Technical Report UMIACS-TR-00-03, University of Maryland
- D. Novick and S. Sutton. 1997. What is mixed-initiative interaction? Papers from the 1997 AAAI Spring Symposium on Computational Models for Mixed Initiative Interaction.
- R. Passonneau and D. Litman. 1997. Discourse segmentation by human and automated means. *Computational Linguistics*, 103-139.
- E. Schegloff and H. Sacks. 1973. Opening up closings. *Semiotica*, 7:289-327.
- D. Schiffrin. 1987. *Discourse Markers*. Cambridge University Press, New York.
- R. Smith and S. Gordon. 1997. Effects of variable initiative on linguistic behavior in human-computer spoken natural language dialogue. *Computational Linguistics*, 23(1):141-168.
- A. Stent. 2000 Rhetorical structure in dialog, in *Proceedings of the 2nd International Natural Language Generation Conference (INLG 2000)*, June 2000. Student paper.
- S. Strayer and P. Heeman. 2001. Reconciling Initiative and Discourse Structure. In *Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue*, pages 153-161.
- D. Traum and E. Hinkelman. 1992. Conversation acts in task-oriented spoken dialogue. *Computational Intelligence*, 8(3):575-599. Special Issue on Non-literal language.
- D. Traum and C. Nakatani. 1999. A two-level approach to coding dialogue for discourse structure: Activities of the 1998 working group on higher-level structures. In *Proceedings of the ACL'99 Workshop Towards Standards and Tools for Discourse Tagging*, pages 101-108, June.

- M. Walker and S. Whittaker. 1990. Mixed initiative in dialogue: An investigation into discourse segmentation. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 70-78.
- S. Whittaker and P. Stenton. 1988. Cues and Control in Expert Client Dialogues. In *Proceedings of the 26th Annual Meeting of the Association for Computational Linguistics*. pages 123-130.
- F. Yang, P. Heeman and S. Strayer 2002 ACT: A graphical dialogue comparison tool. In *Proceedings of the 7th International Conference on Spoken Language Processing*