

Directing the Flow of Conversation in Task-Oriented Dialogue

Fan Yang

B.S., Sun Yat-sen University, 1998

M.Eng., The Chinese Academy of Sciences, 2001

A dissertation submitted to the faculty of the
OGI School of Science & Engineering
Oregon Health & Science University
in partial fulfillment of the
requirements for the degree
Doctor of Philosophy
in
Computer Science and Electronic Engineering

March 2008

The dissertation “Directing the Flow of Conversation in Task-Oriented Dialogue” by Fan Yang has been examined and approved by the following Examination Committee:

Peter A. Heeman
Research Associate Professor
Thesis Research Adviser

Brian Roark
Assistant Professor

Esther Klabbers
Assistant Scientist

David Traum
Research Assistant Professor
Institute of Creative Technology
University of Southern California

Acknowledgements

To begin with, I would like to thank my advisor, Peter A. Heeman, for his support and encouragement during my stay at Oregon Health & Science University. Peter not only advised me through this long PhD study by supplying ideas on research and giving comments on papers, but also at the same time set up an example of how to be an excellent advisor.

I would also like to thank the other members of my thesis committee: David Traum, Brian Roark, and Esther Klabbers, for their effort of reading the thesis and giving a lot of valuable comments to revise this thesis.

I wish to thank the CSLU members. I especially wish to thank Kristy Hollingshead, Xiaochuan Niu, and Susan E. Strayer, for their collaboration and helpful discussion during my PhD study.

I wish to thank Gokhan Tur and Elizabeth Shriberg for providing me the opportunity to work on meeting understanding for the CALO project at SRI, where I learned a lot on applying machine learning techniques to research on human language. I also wish to thank Junlan Feng and Giuseppe Di Fabbrizio for providing me the opportunity of the summer internship at AT&T. I enjoyed the research work at AT&T a lot.

This thesis is based upon work supported by funding from the Intel Research Council and NSF under grant IIS-0326496. The opinions and conclusions are those of the author and not necessarily endorsed by the sponsors.

Finally, I wish to thank the people who are dearest to me. I wish to thank my parents and my sister who have always been there for me. I wish to thank Huana Zheng, who married me in the summer of 2007, for her love and confidence in me.

Contents

Acknowledgements	iii
Abstract	xi
1 Introduction	1
1.1 System- and Mixed-Initiative Dialogue Systems	1
1.2 Thesis Statement	2
1.3 Approach	4
1.4 Terminology	6
1.5 Outline of the Thesis	7
2 Related Research	9
2.1 Modeling Turn-Taking	10
2.1.1 Weak-Turn Model	10
2.1.2 Strong-Turn Model	12
2.2 Modeling Discourse Structure	14
2.2.1 Theories of Discourse Structure	14
2.2.2 Cues for Identifying Discourse Structure	16
2.3 Modeling Initiative and Control	17
2.3.1 Modeling Dialogue Initiative and Control	18
2.3.2 Modeling Task Initiative and Control	19
2.3.3 Mixed-Initiative in Human-Computer Dialogue	20
2.4 Modeling Multi-Tasking in Dialogue Systems	21
2.5 Computer Simulation in Dialogue	23
2.6 Chapter Summery	26
3 DialogueView: An Annotation Tool	27
3.1 Introduction	27
3.2 Existing Tools	29
3.2.1 Audio View Tools	29
3.2.2 Utterance-by-Utterance View Tools	30

3.2.3	Tools with Multiple Views	31
3.2.4	Toolkits	31
3.3	The Basics of DialogueView	32
3.3.1	WordView	33
3.3.2	Segmenting Word Sequences into Utterances	33
3.3.3	UtteranceView	34
3.3.4	Annotating Utterances	36
3.3.5	Grouping Utterances into Blocks	38
3.3.6	Annotating Blocks	40
3.4	Further Abstractions	40
3.4.1	Overlapping Speech	41
3.4.2	Incomplete Utterances, Abandoned Utterances, and Completions . .	43
3.4.3	Speech Repairs	45
3.4.4	Closing Blocks	48
3.4.5	BlockView	50
3.5	User Studies on Abstraction of Dialogue	50
3.5.1	Evaluation on Play Cleanly	51
3.5.2	Evaluation on Clean Display	51
3.6	Comparing Annotations	53
3.6.1	Comparing Speech Repairs	54
3.6.2	Comparing Utterance Segmentations	55
3.6.3	Comparing Utterance Tags	55
3.6.4	Comparing Blocks	56
3.7	Extensibility and Implementation	57
3.8	Chapter Summary	58
4	Corpora and Annotations	60
4.1	The Trains Corpus	60
4.2	The MTD Corpus	61
4.2.1	Tasks	62
4.2.2	Corpus Collection	63
4.2.3	Transcriptions and Annotations	65
5	Managing Complex User Goals	69
5.1	Introduction	69
5.2	Restricted Initiative Model	70
5.3	Evidence from Human-Human Dialogue	71
5.3.1	Frequency of Initiative	71

5.3.2	After Non-initiator Shows Initiative	72
5.3.3	Initiative Patterns	73
5.3.4	Discussion	75
5.4	Evidence from Computer Simulation	76
5.4.1	Computer Simulation: Domain	78
5.4.2	Initiative Strategies	80
5.4.3	Computer Simulation Results	82
5.4.4	Discussion	84
5.5	Chapter Summary	85
6	Managing Multiple Tasks	87
6.1	Introduction	87
6.2	Where to Switch	88
6.2.1	Urgency Level and Place of Switching	88
6.2.2	Waiting Time and Place of Switching	89
6.2.3	Discussion	90
6.3	How to Switch	90
6.3.1	Task Switching and Discourse Markers	91
6.3.2	Task Switching and Prosody	91
6.3.3	Signaling Place of Switching	93
6.3.4	Discussion	94
6.4	Chapter Summery	95
7	Managing Initiative Conflicts	96
7.1	Introduction	96
7.2	Identifying Initiative Conflicts	97
7.3	Avoiding Initiative Conflicts	99
7.3.1	Offset of Initiative Conflicts	99
7.3.2	Urgency Level and Initiative Conflicts	101
7.4	Resolving Initiative Conflicts	102
7.4.1	Features during the Overlap	103
7.4.2	Features from Context	107
7.5	Repairing Speech after Initiative Conflicts	108
7.6	Discussion	109
7.7	Chapter Summery	111

8 Conclusion	112
8.1 Human Conventions on Directing the Conversation Flow	112
8.2 Application to Dialogue Systems	114
Biographical Note	125

List of Tables

3.1	Statistics for the two dialogues used in testing	52
5.1	Frequency of initiative	72
5.2	After a speaker shows initiative	73
5.3	Patterns of initiative	74
6.1	Urgency level and place of switching	89
6.2	Waiting time and place of switching	90
6.3	Average prosodic values for each player	92
6.4	Pitch (Hz) and place of switching	94

List of Figures

1.1	An example for system-initiative dialogue	3
1.2	An example dialogue with ConQuest	3
1.3	An example for initiative and control	7
3.1	Utterance segmentation in WordView	33
3.2	Utterance segmentation in UtteranceView	35
3.3	UtteranceView with utterance tags displayed	36
3.4	Sample Utterance Annotation Panel	38
3.5	Grouping utterances into blocks in UtteranceView	39
3.6	Misleading utterance segmentation	42
3.7	Altered start time for “okay” utterance	42
3.8	Abandoned Utterances	44
3.9	Speech Repair Example	45
3.10	Effect of overlap, abandoned, incomplete, and completion tags	47
3.11	Dialogue excerpt with three blocks closed	49
3.12	Dialogue excerpt with all of the blocks closed	49
3.13	BlockView	50
3.14	Accuracy results: black is for the full style and white for abstracted	53
3.15	Duration results: black is for the full style and white for abstracted	53
3.16	Interface of ACT	54
3.17	Block comparison in the BlockDiffView	56
4.1	The game display for players	64
4.2	Events displayed in the WordView of DialogueView	65
4.3	An excerpt of an MTD dialogue	68
5.1	An example for initiative and control	70
5.2	Example of an adjacency pair and delegated control (d92a-5.2)	76
5.3	Solution quality	83
5.4	Communication effort	84
5.5	Solution quality on each plan	84

7.1	An example of an initiative conflict	97
7.2	Preceding-pause (A-B), offset (B-C), and overlap (C-D) of initiative conflict	99
7.3	CDF plot for offset of initiative conflicts	100
7.4	Long offset: ambiguous intonation	101
7.5	Long offset: other	101
7.6	CDF plot for duration of overlaps and length of forward utterances	103
7.7	Negotiating initiative	111

Abstract

Directing the Flow of Conversation in Task-Oriented Dialogue

Fan Yang

Supervising Professor: Peter A. Heeman

We envision that next-generation spoken dialogue systems will be supporting a complex user goal and multiple parallel tasks, which requires the system and the user to jointly direct the flow of conversation. The problem, however, is that there lacks an effective model of directing the flow of conversation. This thesis research aims to develop such a model for next-generation spoken dialogue systems. We started with conventions actually used in human-human dialogue, which are natural for users to follow and probably also efficient in problem-solving. An annotation framework, DialogueView, was established to allow for the investigation of complex interaction in dialogue. A series of empirical studies on two corpora, the Trains and the MTD, were then conducted to understand people's initiative behavior of directing the conversation flow. We first examined people's initiative behavior in decomposing a complex goal into sub-goals and achieving each of them in the Trains domain. We found that initiative is subservient to discourse goal. We next examined people's initiative behavior in switching the conversation to a more urgent task. We found that conversants strive to switch tasks at a less disruptive place; but where they cannot, they exert additional effort to signal the task switching, such as increasing pitch. We finally examined people's behavior on initiative conflicts, where both conversants try to direct the conversation at the same time. We found that conversants try

to avoid initiative conflicts; but when initiative conflicts occur, they are efficiently resolved with simple linguistic devices such as volume. Computer simulation experiments were also conducted to better understand the underlying benefits of using the human conventions. Our findings on human-human dialogues have important implication for building next-generation spoken dialogue systems by (1) guiding the system when to show initiative, and when to let the user show initiative; (2) guiding the system when and how to switch to a more urgent task, and to understand the user's switch; and (3) guiding the system how to resolve and repair initiative conflicts.

Chapter 1

Introduction

1.1 System- and Mixed-Initiative Dialogue Systems

Researchers and engineers are developing spoken dialogue systems that communicate with human users via speech. Spoken dialogue systems extend the traditional human-computer interaction modality of keyboard and mouse. Such systems have promising applications in automatic call centers, in hand-busy and eye-busy situations, and for blind people.

The first generation of spoken dialogue systems have mainly been *system-initiative*, in which the system asks questions and the user answers. In these applications, the system directs the flow of conversation; the user is not allowed to volunteer information, but just follows the lead of the system. Figure 1.1 shows an example system-initiative dialogue with ELVIS (Email Voice Interactive System) copied from Walker et al. (1997a). In this example, the system asked a sequence of questions to acquire the user's request, and the user passively gave short answers as parameters of the request. These systems have been successful in simple domains, such as banking assistance, flight status checking, and weather forecast inquiry. In fact, the usefulness of such systems depends on how structured the task is. If the task can be formulated into a form-filling process, or be modeled as a decision tree, system-initiative dialogue is a good choice, especially when the state of the art technology on speech recognition is far from perfect. System-initiative dialogue reduces the perplexity of the user's response, simplifies syntactic parsing, semantic interpretation and dialogue management, and thus makes the conversation more robust.

As the complexity of the task increases, system-initiative dialogue will not be sufficient, as it will hinder the user's ability to work with the system to accomplish the task. Hence,

researchers have been attempting to go beyond the realm of system-initiative dialogue to *mixed-initiative* dialogue. The ConQuest system (Bohus et al., 2007) providing conference information such as program schedule, is a deployed mixed-initiative spoken dialogue system. Figure 1.2 shows an example dialogue with ConQuest copied from Bohus et al. (2007). The ConQuest system is an improvement over system-initiative as the user is allowed to take control to specify requests, which leads to more efficient conversations. The initiative strategy, however, is simple. The user is assigned control, and shows initiative of making requests (e.g. u2 and u12); if the system fails to understand (e.g. u15) or is unable to fulfil the request (e.g. u13), the system will take over the control to guide the user in completing the task. This mixed-initiative strategy, however, is still in the framework of form-filling dialogue. If the user's request involves complex domain reasoning and planning, or even requires the collaboration of both the system and the user to carry it out, this rigid strategy will not be sufficient.

1.2 Thesis Statement

We envision that next-generation spoken dialogue systems will be able to:

manage a complex user goal: The user's goal can be quite complex. The system needs to work with the user to decompose it into sub-goals, and then work on each sub-goal. Both the system and the user should be able to direct the conversation flow, i.e. to show initiative. On one hand, the system should allow the user to actively participate in the reasoning procedure; for example, to ask a question or to make a proposal. On the other hand, the system also needs to actively contribute to a dialogue, efficiently and effectively helping the user to accomplish a task. Thus the system should know when to take the lead and when to let the user take the lead in directing the conversation flow.

manage multiple tasks: Next generation spoken dialogue systems should be able to work with the user on multiple tasks at the same time. For instance, a driver in a car might use a dialogue system to check the daily schedule, while occasionally asking for upcoming traffic conditions and receiving navigation instructions. When

u1 system: Hi, Elvis here. You have 5 new and 0 unread messages in your in box.
 Say Repeat to repeat this message, or say Read, Summarize, or Im done here.
 u2 user: Read.
 u3 system: Select by Content or Position?
 u4 user: Content.
 u5 system: Select by Sender or Subject?
 u6 user: Sender.
 u7 system: Which Sender?
 u8 user: Kim.
 u9 system: The message from Kim is about Meeting Tomorrow.
 The meeting tomorrow is at 10:30 in 2D-516.

Figure 1.1: An example for system-initiative dialogue

u1 system: Welcome to ConQuest [...] I can give you information about papers, sessions, special events and announcements at Inter-Speech 2006. You can also cast a vote for the best paper award. How can I help you?
 u2 user: are there any papers on spoken dialog systems today?
 u3 system: There are 23 papers on spoken dialog systems today. These papers are being presented in 3 sessions: Dialog and Discourse, Language Modeling for Spoken Dialog Systems and Spoken Dialog Systems I. To browse this list of papers, please tell me the author name, the paper title, the time or the venue youre looking for. If youre not interested in any of these papers say quit.
 u4 user: which ones are in the afternoon?
 u5 system: There will be 11 papers on spoken dialog systems this afternoon.
 Would you like to hear information about all these papers?
 u6 user: yes.
 u7 system: The list of papers on spoken dialog systems this afternoon is: Paper 1 - "Dynamic Extension of a Grammar-based Dialogue System: Constructing an All-Recipes Knowing Robot", Paper 2 [...] Would you like more information about any of these?
 u8 user: tell me more about the third one.
 u9 system: Paper 3 is titled "System- versus user-initiative dialog strategy for driver information systems", and was presented yesterday at 2 pm in the poster session in Allegheny Ballroom Foyer. This paper was written by Chantal Ackermann, and Marion Libossek. Would you like more information about any of these?
 u10 user: no when is the banquet
 u11 system: The Banquet will be held Wednesday from 7pm to 9:30 pm in the Heintz History Center. The Heinz History Center is located on Smallman street between 12th and 13th streets. The History Center is a museum that relates and interprets 250 years of history in the Western Pennsylvania area. Is there anything else I can do for you?
 u12 user: yes I'd like to cast a vote
 u13 system: Please tell me the title of the paper youd like to vote for
 u14 user: I don't really know the title but I think it had Diane as one of the main authors
 u15 system: [non-understanding] Sorry, I didnt catch that. Please dial the first six letters from the title of the paper you would like to vote for.
 u16 user: I dont know if it's a paper by Diane Litman
 u17 system: There are 3 papers by Diane Litman. Paper 1 "Discourse Structure and Speech Recognition Problems", Paper 2 [...], Paper 3 [...] Would you like to vote for any of these?
 u18 user: Discourse Structure and Speech Recognition Problems
 u19 system: Okay, Ive recorded your vote for the paper entitled "Discourse Structure and Speech Recognition Problems "What else can I do for you?

Figure 1.2: An example dialogue with ConQuest

working with the user on multiple tasks, the system should know when to switch the conversation flow to another task, how to signal the switch, and how to understand a user's switch.

manage initiative conflicts: *Initiative conflicts*, where both the system and the user try to direct the conversation flow in different directions at the same time, may arise in mixed-initiative and multi-tasking dialogues. Current spoken dialogue systems either always ignore the user's speech or always stop speaking whenever they detect that the user is speaking. However, with more advanced computer intelligence, the system will play a more important role in human-computer interaction. In a complex domain where the system and user are collaborating on a complex goal or on multiple tasks, the system, just like the user, might have the need to override the other conversant to correct a misconception, or to convey some ideas. Thus the system should know under what situations initiative conflicts occur, what mechanisms are used to resolve them, and how to repair any impairment to the conversation that the conflicts cause.

The problem, however, is that there exists no effective model to guide the system to jointly direct the flow of conversation with the user on a complex goal, on multiple tasks, and on initiative conflicts. The research of this thesis aims to develop a mode of directing the conversation flow for next-generation spoken dialogue systems. This model will provide the basis for natural mixed-initiative and multi-tasking human-computer dialogue by (1) guiding the system when to show initiative, and when to let the user show initiative; (2) guiding the system when and how to switch to another task, and to understand the user's switch; and (3) guiding the system how to resolve and repair initiative conflicts.

1.3 Approach

There are several approaches to develop a model for spoken dialogue systems. For example, one can start by conjecturing a model, implement it, and then fine-tune it according to feedback from users. Instead of creating a model based on conjecture, we feel that a more reasonable approach is to start with conventions actually used in human-human dialogue.

In human-human dialogue, people organize words into utterances. Each utterance usually serves as a speech act (Searle, 1969), such as stating a fact, or answering a question. People take turns to speak, communicating meaningful utterances (Sacks, Schegloff, and Jefferson, 1974). Turn-taking is a basic mechanism of conversation: when one talks, the other listens. It resolves the limited-resource problem that the communication channel can be used for information transmission by only one conversant at a time.

Turn-taking, however, does not address the problem of how people direct the conversation flow towards the dialogue goal. According to Robert Hargrove (1995): *a dialogue is a conversation where there is a free flow of meaning in a group and diverse views and perspectives are encouraged*. There is no pre-set direction for a conversation. At any time, each conversant might have his or her own (partial) solution to achieve the (sub-)goal, and each solution might lead the conversation flow in a different direction. There are conventions of directing of conversation flow towards the dialogue goal in human-human conversations.

An analogy of directing the conversation flow of dialogue is operating a dual-control airplane by two pilots. Each conversant is a pilot who can operate the plane. There must be a set of mediation rules in determining who operates and when, otherwise if both pilots operate the plane at the same time in different directions, unexpected consequences might happen. Likewise, researchers have been interested in how conversants collaborate in directing the flow of conversation to avoid the dialogue falling apart.

A good model for spoken dialogue systems should be natural for users to follow and be efficient in accomplishing a task at hand. Human conventions are a good starting point for developing a model for spoken dialogue systems. First, a model based on human conventions is natural as the conversational style of talking to the system is similar to human-human dialogue; users can employ their daily conversational skill to talk with the system without extra training. Second, it should also be efficient as human languages have evolved over generations: efficiency, such as minimizing collaborative effort (Clark and Wilkes-Gibbs, 1986), is one if not all of the driving forces for this era-long evolution.

In this thesis research, we examined how people direct the flow of conversation. A series of empirical studies on human-human dialogue were conducted to understand the

conventions people use in negotiating who directs the conversation and when. Computer simulation is also used to further investigate the human conventions.

1.4 Terminology

In the literature on directing conversation flow, two terms, *initiative* and *control*, have been used by different authors or sometimes interchangeably by the same authors. These terms are used at two separate levels. The first is at the utterance level. When a conversant makes a certain type of utterance, the conversant is said to be showing control or initiative. The second is at a more global level, and is something that the conversant has that lives on beyond individual utterances. We treat these two phenomena as separate, and use *initiative* to refer to the first, and *control* to the second.

Initiative is the behavior of volunteering new information without being explicitly requested to, or of asking a question to establish mutual belief, or of proposing a new domain goal to achieve. Initiative is short-lived. Initiative begins and ends with the utterance. In the dialogue excerpt in Figure 1.3, the user shows initiative in utterance u1, u3, u5 and u6; the system shows initiative in u7; but nobody shows initiative in utterance u2, u4, and u8.

Initiative is closely related to turn-taking: a conversant has to possess the turn in order to show initiative. However, a conversant who has the turn might not show initiative. For example, an answer takes a turn, but does not show initiative.

Control is the authority to direct the conversation flow towards the achievement of the dialogue (sub-)goal. Control is usually negotiated and mutually agreed by both conversants. In contrast to initiative, control has lasting effect beyond an utterance. Control is exhibited by initiative: the conversant who has control tends to show initiative. In Figure 1.3, from u1 to u6, the user is the only conversant who shows initiative in proposing a solution; thus the user has control in this stretch of dialogue. Similarly, the system has control on u7 and u8.

u1	user:	then how about if I take an engine along with a boxcar over to Corning
u2	system:	mm-hm
u3	user:	add a tanker on
u4	system:	mm-hm
u5	user:	fill the boxcar with oranges and bring it back to Elmira to have it turned into juice
u6	user:	and drop off the boxcar fill the tanker with orange juice and deliver it to Avon
u7	system:	that will take twelve hours
u8	user:	ok

Figure 1.3: An example for initiative and control

1.5 Outline of the Thesis

This thesis research aims to understand how people direct the conversation flow on managing a complex goal, on managing multiple tasks, and on managing initiative conflicts. Chapter 2 gives an overview of some of the previous research work on which this thesis work is built. It provides a more detailed description of turn-taking, discourse structure, control and initiative. We argue that existing literature did not fully capture human’s initiative behavior of solving a complex goal, switching between tasks, or resolving initiative conflicts.

To facilitate research on modeling human-human dialogue, we have built the tool, DialogueView, to graphically display a dialogue in multiple views, with each view presenting an abstraction of the dialogue. DialogueView can be used to annotate speech repairs, utterance boundaries, utterance tags, and hierarchical discourse structure. DialogueView allows for the investigation of complex interactions within dialogue, including task switching and initiative conflicts. Chapter 3 describes DialogueView.

The tool DialogueView was used to annotate the Trains corpus (Heeman and Allen, 1995b) and the MTD corpus (Heeman et al., 2005), both of which are human-human task-oriented dialogues that we examined to understand human’s conventional behavior on directing the conversation flow. The Trains corpus was collected at University of Rochester. It contains a collection of conversations in which conversants solve a task that requires complex domain reasoning. To understand how people manage multiple tasks in conversation, we have also taken the effort to collect the MTD corpus, in which conversants finish two types of tasks via conversation: an on-going task that takes a long time to finish

and a momentary task that just takes several turns to complete but has a time constraint. We describe the collection and annotations of these two corpora in Chapter 4.

In Chapter 5, we describe an empirical study on human behavior of directing the conversation flow to solve a complex goal. Strayer et al. (2003) proposed the restricted initiative model that the person who starts a discourse sub-goal leads the conversation flow to achieve the sub-goal. We provided strong evidence from the Trains corpus to support this model. Computer simulation experiments were also conducted to investigate the explanatory account of this model.

In Chapter 6, we describe an empirical study on human behavior of directing the conversation flow with multiple tasks. We examined the discourse context where conversants switch from an ongoing task to a momentary task. We found that conversants strive to switch at a less disruptive place. We also examined the cues that conversants use to signal task switching. We found that task switching is correlated with certain discourse markers and higher pitch. Moreover, when conversants switch tasks at a more disruptive place, they would use even higher pitch. We thus establish pitch as the primary prosodic signal of task switching.

In Chapter 7, we describe an empirical study on human behavior of initiative conflicts, where both conversants speak at the same time to drive the conversation flow towards different directions. We examined how initiative conflicts occur, how they are resolved, and how people repair the impaired speech due to the conflicts. We found that conversants try to avoid initiative conflicts, but when initiative conflicts occur due to an urgency reason or an unintentional collision when both conversants happen to start speaking almost at the same time, they are efficiently resolved. We established volume as the primary mechanism for winning initiative conflicts.

We summarize this thesis in Chapter 8.

Chapter 2

Related Research

In this chapter, we first review research on turn-taking in Section 2.1, the basic mechanism of interaction in conversation. In human-human dialogue, conversants take turns to speak: when one talks, the other listens. Overwhelmingly, one conversant talks at a time, although there might be times when both conversants are talking simultaneously, causing *overlapping speech* that includes back-channels, fights for the turn, and others. Turn-taking in conversation is strongly related to initiative. Conversants have to possess the turn in order to show initiative. When conversants are fighting for initiative, they are also fighting for the turn to speak. Thus the mechanisms of turn-taking might share some similarity with initiative. On the other hand, turn-taking is different from initiative; for example, an answer takes a turn, but does not show initiative.

Research on turn-taking describes how speakership is transferred between conversants. Turn-taking, however, does not capture how people solve a complex goal. We then review research on discourse structure in Section 2.2, which focuses on a higher organization of dialogue – how people decompose a complex goal into a set of sub-goals. We also review research on signalling discourse structure as it inspires the hypothesis that people use the similar cues to signal task switching in multi-tasking dialogue.

Discourse structure theory, however, does not capture how people direct the conversation flow to achieve the sub-goals. We next review models of initiative in Section 2.3 – how conversants collaborate to direct a conversation. As we will point out, current initiative models were vague about the relation between initiative and discourse goals, and did not address the problem of initiative conflicts.

As we envision that next-generation dialogue systems need to support multiple tasks,

we review research on multi-tasking dialogue in Section 2.4. Current multi-tasking dialogue systems turned out to freely switch between tasks without much signaling, which could confuse the user which task is currently active in the system.

A minor part of this thesis research is to investigate the use of computer simulation experiments for understanding human initiative behavior. We thus review research of using computer simulation to help understanding human-human conversations in Section 2.5.

2.1 Modeling Turn-Taking

Turn-taking in conversation has been discussed extensively in the literature. Researchers have different accounts of what constitutes a speaking turn, particularly on whether an acknowledgement constitutes a turn. We refer to the account that an acknowledgement constitutes a turn as a *weak-turn model* and the opposite as a *strong-turn model*.

2.1.1 Weak-Turn Model

Sacks, Schegloff and Jefferson (1974) viewed any stretch of speech between switching speakership as a turn. An acknowledgement, a collaborative completion, or a clarification question can constitute a speaking turn. Sacks, Schegloff and Jefferson proposed a set of rules for turn-taking (SSJ model): at each possible turn transition point, the current speaker can select somebody else to speak; otherwise, the hearer can self-select to speak; otherwise, the speaker can self-select to speak.

Timing is critical in the SSJ model. The hearer should decide when the speaker is going to release the turn, and so can self-select to speak; and the speaker should decide when the hearer does not want to speak, and so can self-select to speak. If either goes too early, there will be overlapping speech; and if either waits too long, there will be a long pause. Sacks (2004) claimed that conversants try to minimize the pause at turn switches, while at the same time try to avoid overlapping speech. Jefferson (1973) illustrated that hearers can precisely predict the end of an utterance to minimize the pause, and Schegloff (1987; 2000) examined overlapping speech in details.

Schegloff (1987) described a special situation where conversants negotiate for turns

by purposefully introducing overlapping speech. Schegloff pointed out that “there are also techniques available to current speakers to try to extend their talk past a possible transition point, into a whole new unit, which could itself get some considerable extension. In one such technique, a current speaker approaching a possible completion point of a turn-constructional unit, speeds up the talk and runs the intonation contour and phrasing across the completion point, getting into a new sentence before slowing down or taking a breath.” If the current speaker kept exercising the technique to extend her¹ turn, she could block the hearer from having the turn and could talk forever. Schegloff (1987) proposed that the hearer has a counter technique to fight for his turn. The hearer can start talking in parallel with the speaker, causing overlapping speech. If the speaker stops talking, the hearer restarts from the turn beginning. Schegloff called this technique *recycling turn beginnings*. However, Schegloff did not examine how often conversants would exercise this technique, or under what circumstances conversants would exercise this technique to fight for the turn.

Schegloff (2000) also investigated how fights for the turn are resolved. He found that (a) fights for the turn are often accompanied with sudden acoustic alterations, such as louder volume, higher pitch, and faster or slower speaking rate; (b) the vast majority of fights for the turn are resolved very quickly; and (c) fights for the turn are resolved through an interactive procedure, e.g. syllable by syllable negotiation, using devices such as volume, pitch, and speaking rate. However, his analysis only consisted of a few examples; no statistical evidence was given. It is thus unclear whether his conclusions represent human conventions of initiative conflict, or are occasional behavior that would only occur under special circumstances.

Schegloff pointed out that the hearer’s restarting the turn beginnings after fighting for the turn is critical because the turning beginnings are overlapped and thus might be impaired. Turn beginnings, as argued by Schegloff, are important, because (a) they identify the relation between the current turn to the previous turns; and (b) they identify what the speaker is going to talk about. Restarting the turn beginning thus better guarantees

¹Throughout this thesis, when there is no ambiguity, we use the female pronoun “she” and “her” to refer to the speaker, and the male pronoun “he” and “him” for the hearer.

that the intended information is delivered. However, it is unclear how prevalent speakers restarting turn beginnings after fights for the speaking turn.

Following the SSJ model, Traum and Hinkelman (1992) developed an annotation scheme for turn-taking in dialogue. The annotation scheme includes four turn-taking acts: take-turn, keep-turn, release-turn, and assign-turn. At each possible turn completion point, any instance of the hearer starting to talk is viewed as a take-turn attempt; and any instance of the speaker continuing to talk is viewed as a keep-turn action. Release-turn actions are usually signaled by intonation. Assign-turn actions are a sub-class of release-turn in which the speaker explicitly allocates the turn to the hearer, for example, by asking a question. In Traum and Hinkelman’s annotation scheme, acknowledgements are viewed as a take-turn followed immediately by a release-turn.

2.1.2 Strong-Turn Model

Duncan defined back-channels to include acknowledgements, collaborative completions, clarification questions, and brief statements, and excluded them from speaking turns (Duncan, 1972; Duncan, 1974; Duncan and Niederehe, 1974). Back-channels were viewed as the hearer actively avoiding the speaking turn, keeping it squarely with the current speaker. However, Duncan also stated that “on an intuitive basis, some of these longer back-channels appeared to take on the quality of a turn”, such as clarification questions or brief statements. Duncan’s strong-turn model distinguishes itself from weak-turn model in that it is closer to initiative. In Duncan’s model, when a conversant takes the turn to speak, the conversant is also showing initiative, except when the speaking turn is assigned, such as answering a question.

Duncan (1972) identified six types of *turn-releasing cues* that the speaker signals to the hearer of an opportunity to take over the turn. The first cue is an intonational pattern of either a rising or falling pitch followed by an intermediate pitch at the end of an intonational phrase. The second cue is a drawl on the final syllable or on the stressed syllable of an intonational phrase. The third cue is gesture: the termination of any hand gesture used during a speaking turn or the relaxation of a tensed hand position (e.g. a fist). The fourth cue is cue words, such as “or something” or “you know”. The fifth cue

is a drop of pitch or loudness with cue words. The sixth cue is syntax: the completion of a grammatical clause. Duncan found that the probability of the hearer taking the turn increases as a positive linear function of the number of cues conjointly displayed by the speaker. However, Duncan also found that some gestures seem to negate the turn-releasing cues. These gesture cues are (a) one or both of the speaker's hands moving generally away from the body, or (b) a tensed hand position, such as a fist. When these cues appear, the hearer rarely takes the turn, no matter how many turn-releasing cues are displayed by the speaker.

Duncan and Niederehe (1974) examined cues that the hearer uses to signal the taking of the speaking turn. Compared to back-channels, turn beginnings have a significantly higher correlation with four types of cues: shifting the head away from pointing directly toward the speaker, audible inhalation, initiation of a gesture, and higher loudness.

Duncan (1974) identified two types of cues that the speaker uses to signal to the hearer an opportunity to make a back-channel, which he called *speaker within-turn cues*: (a) completion of a grammatical clause, and (b) turning the head toward the hearer. The display of either one of the cues is sufficient to constitute the signal. The nature of the relationship between the speaker within-turn cues and the hearer's back-channel is identical to the relationship between the speaker turn-releasing cues and the hearer taking the turn.

Duncan (1974) also found that when the speaker resumes talking after the hearer's back-channel (or a silence response from the hearer), the speaker tends to turn the head away from the hearer. Duncan called this the *speaker continuation cue*.

A model based on these signals was created to account for conversants' turn-taking behavior (Duncan, 1974). This model, represented as a flow chart starting from one of the conversants (say A) having the turn, determines who will have the turn next, depending on what signals are displayed. For example, if A displays a within-turn signal, and B gives a back-channel, and if A then displays a continuation signal, then A still keeps the turn to speak; if A does not display a within-turn signal, but displays a turn-releasing signal (without the negating gesture signal), and if B then displays a turn-taking signal, then B has the turn. This model, however, has one logical deficit. Because the within-turn signal

and the turn-releasing signal share the same cue of completion of a grammatical clause, if A is releasing the turn at the completion of a grammatical clause, the flow chart will never go to the branch of A displaying the turn-releasing signal, but goes to the branch of A displaying the within-turn signal. Thus the next action will always be B making a back-channel, and never be B claiming the turn.

According to Duncan’s flow chart model, miscues are the cause of fights for the turn: for example, the hearer mis-recognizes the speaker’s cue to keep the turn, or the speaker incorrectly signals the release of turn. Duncan and Nederehe (1974) also proposed a predictive algorithm of how people resolve fights for the turn. For each conversant, Duncan and Nederehe calculated a value as the number of cues of taking the turn less the number of turn-releasing cues displayed. The conversant who has the higher value was predicted to win the fight. In an evaluation of 19 cases of fights for turn, 18 were correctly predicted by this model; and the last one was a tie, which yielded no prediction. The problem with this model is that, especially for the turn-releasing cues, it is unclear whether these cues are a cause or an effect of the conversants resolving the fights for turn. For example, after the fight has been resolved, the yielder might then produce the turn-releasing cues. A model using cues that are the side-effects of taking the turn or releasing the turn is less useful in guiding a dialogue system negotiating with its user.

2.2 Modeling Discourse Structure

Beyond single-utterance and turn-taking, conversants organize individual utterances into a coherent conversation to achieve the dialogue goal. Several theories on discourse structure have been proposed, capturing the organization of written text, monologue, and dialogue.

2.2.1 Theories of Discourse Structure

Researchers have proposed a lot of theories to model discourse structure. Grosz et al. (1995) proposed the centering theory to account for the coherence of local discourse

(inside a discourse segment). Mann and Thompson (1987) proposed the rhetorical structure theory (RST) in which all rhetorical coherence relations between text spans are categorized into 23 relation types on two levels: the informational level and the intentional level. Stent (2000) developed an annotation scheme to apply RST to dialogue. Hobbs (1985) proposed a theory that relates text interpretation to belief systems. Polanyi (1988) proposed the linguistic discourse model (LDM) in which each clause is attached to the discourse tree from top to down and from left to right; Rose et al. (1995) extended the LDM by allowing a new utterance to be attached to any open branch; Reichman-Adar (1984) adopted augmented transition networks (ATN) to create a context space for each clause and interpret the clause in this context space; Kamp and Reyle (1993) proposed the discourse representation theory (DRT), which provides a semantic framework for expressing the meaning of a sentence as an operation of updating contexts. To combine discourse structure with domain planning, Litman and Allen (1987) proposed a two-level model: dialogue planning as a lower level and domain planning as a higher level. Ramshaw (1991) extended Litman and Allen’s model by adding a problem-solving level in between. None of these models, however, emphasized enough on the conversants’ intentions.

Grosz and Sidner (1986) proposed a theory of discourse structure to account for why an utterance was said and what was meant by it. The theory has three components: linguistic structure, intentional structure and attentional state. Utterances, as the basic units of discourse, are grouped into embedded discourse segments according to the speaker’s intention. Discourse segment boundaries are identified by linguistic cues, such as discourse markers (e.g. now, anyway, and well), and prosodic cues (Grosz and Hirschberg, 1992). Each discourse segment has a discourse segment purpose (DSP), which is the recognizable intention of the segment initiator. The other dialogue participants recognize this DSP and contribute together with the initiator to the DSP. Relations between DSPs can be either dominance or satisfaction-precedence. DSP1 dominates DSP2 if DSP2 contributes to part of DSP1; DSP1 satisfaction-precedes DSP2 if DSP1 must be achieved before DSP2. The DSPs and their relations constitute the intentional structure. Attentional state is an abstraction of the focus of the dialogue participants. Each discourse segment also corresponds to an attentional state. The attentional state contains the objects, the properties,

and the relations that are most salient at the current point, which are what the dialogue participants are talking about. The attentional state also contains the current DSP, which is why the dialogue participants are talking about the other components in the attentional state. The structure of the set of attentional states is modeled as a stack. A push operation occurs when a discourse segment begins, and this attentional state is popped out when the corresponding discourse segment ends. All the items in the stack are in the current focus with the degree of salience decreasing with the depth.

Lockbaum (1998) tied Grosz and Sidner’s discourse structure theory with domain planning. Each discourse segment is associated with a sub-plan that contributes to the overall domain goal. The DSP is the initiator’s intention that both conversants jointly form the sub-plan and commit to its success. The relations between sub-plans provide the basis for determining the corresponding relations between DSPs.

Grosz and Sidner’s discourse structure theory, or the extended model by Lockbaum, however, did not comment on the initiative behavior of conversants. Hence, they underspecified how conversants collaborate in directing a dialogue to achieve the (sub-)goals.

2.2.2 Cues for Identifying Discourse Structure

Although there is little research on how conversants signal task switching in multi-tasking dialogues, there is a tremendous amount of research on signalling the organization of a discourse. Several types of cues have been identified, including discourse markers, prosodic information, and co-reference. These cues are a good starting point from which to study how people signal task switching in multi-tasking dialogue.

Discourse Markers

The first type of cues are discourse markers (Moser and Moore, 1995; Schiffrin, 1987; Grosz and Sidner, 1986; Passonneau and Litman, 1997; Bangerter and Clark, 2003; Byron and Heeman, 1997). Discourse markers can be used to signal the start of a new discourse segment and its relation to the previous segments. For example, *now* might signal moving on to the next topic, while *well* might signal a negative or unexpected response.

Prosody

The second type of cues are prosody. In read speech, Grosz and Hirschberg (1992) studied broadcast news and found that pause length is the most important factor that indicates a new discourse segment. Ayers (1992) found that pitch range appears to correlate more closely with hierarchical topic structure in read speech than in spontaneous speech. In spontaneous monologue, Butterworth (1972) found that the beginning of a discourse segment exhibited a slower speaking rate; Swerts (1995), and Passonneau and Litman (1997) found that pause length correlates with discourse segment boundaries; Hirschberg and Nakatani (1996) found that the beginning of a discourse segment correlates with higher pitch. In human-human dialogue, similar behavior was observed: the pitch value tends to be higher for a new discourse segment (Nakajima and Allen, 1993). In human-computer dialogue, Swerts and Ostendorf (1995) found that the first utterance of a discourse segment correlates with longer duration and longer preceding pause.

Co-reference

Walker and Whittaker (1990) suggested that co-reference is also a cue for identifying discourse structure: for example, 3rd person pronouns, such as *it* and *he*, are usually used to refer to objects in the same discourse segment (the most salient attentional state); while deictic pronouns, such as *this* and *that*, can be used across discourse boundary.

The research on signalling discourse structure has been mainly focused on single-tasking dialogue, i.e. signalling the finish of a topic and moving-on to the next one. Little is known about what cues conversants use to signal the switch of tasks in multi-tasking dialogue.

2.3 Modeling Initiative and Control

Researchers have been investigating how people manage initiative and control in conversation. Models of initiative and control aim to account for which conversant is driving the conversation flow at any given point.

2.3.1 Modeling Dialogue Initiative and Control

Whittaker and Stenton (1988) proposed rules for tracking initiative² based on utterance types; for example, statements, proposals, and questions show initiative, while answers and acknowledgements do not. Applying these rules, they found that “[initiative] did not alternate from speaker to speaker on a turn by turn basis, but that there were long sequences of turns in which [initiative] remained with one speaker.” They also found that the person who initiates a topic shows initiative in more turns in discussing the topic, which suggested that the switch of initiative is related to discourse structure. According to these observations, Whittaker and Stenton segmented dialogues at points where initiative shifts from one conversant to the other and proposed a three-level model of discourse structure: initiative segments at the lowest level, topic segments in the middle, and global organization at the top. Walker and Whittaker (1990) extended this work by finding anaphora evidence showing that negotiated shifts of initiative imply shifts of attentional states, i.e. initiative segments are consistent with attentional states. However, initiative segments are only an approximation of discourse segments. There are cases where initiative shifts in a discourse segment, and cases where a conversant shows all initiative in several consecutive discourse segments.

Cohen et al. (1998) proposed that the person who initiates a process has control over the whole process. A process is “a finite set of turns in an interaction.” They also argued that initiative is not a binary choice, but that control can be presented in variable degrees of strength. The non-initiator of a process can also show initiative, but this initiative is not as strong as that of the first utterance of the process. However, the definition of process by Cohen et al. is vague, obscuring the meaning of control. Furthermore, they did not build a model to measure the strength of initiative.

Strayer et al. (2003) also examined how initiative and control are managed in task-oriented dialogues. Strayer et al. hypothesized that control is subordinate to discourse structure. They annotated the Trains dialogues (Heeman and Allen, 1995b) with discourse structure following Grosz and Sidner (1986)’s theory. They found that the initiator of

²Whittaker and Stenton used the term *control*, which corresponds to our definition of *initiative* in Section 1.4.

a discourse segment shows more initiative inside the segment. The non-initiator only makes two types of contributions occasionally: collaborative completions, in which the non-initiator helps the initiator to complete an utterance, and short contributions to the discourse segment purpose. Thus they concluded that the initiator of a discourse structure segment has control for the entire segment. However, the evidence found by Strayer et al. was not sufficient for supporting this conclusion. For example, they did not examine whether the initiator and the non-initiator alternate in showing initiative inside a discourse segment, which is an indication that the initiator does not have control over the entire segment.

Furthermore, all models of initiative and control above allow the possibility that multiple conversants will want to show initiative at the same time, causing initiative conflicts, particularly because a conversant only has incomplete information about the world and the other conversant. However, none of the models addressed initiative conflicts.

2.3.2 Modeling Task Initiative and Control

Chu-Carroll and Brown (1998) further distinguished two types of initiative: dialogue initiative and task initiative. A person is said to have the dialogue initiative if the person “takes the conversational lead in order to establish mutual beliefs between the agents,” for example, stating knowledge about the world. A person is said to have the task initiative if the person “is directing how the agents’ task should be accomplished,” for example, proposing actions to perform. Task initiative implies dialogue initiative. Chu-Carroll and Brown also identified eight types of cues of initiative shifts, such as explicit request, no new information, and invalid or ambiguous information. They then employed these cues to build an evidential model to predict which conversant is showing dialogue initiative and task initiative next.

Guinn (1998) argued that there is no need to separate dialogue initiative from task initiative because the dialogue goal is a sub-goal of the overall task, thus “dialogue initiative should always pass immediately to the participant who is able to handle the current task.” Guinn proposed that the person who is able to coordinate the current subtask is the person who leads the dialogue. Initiative switches between conversants as goals get

pushed and popped from the problem-solving stack. Allen (1999) holds the same viewpoint. Each agent constantly monitors the current task and evaluates whether it should take the initiative in the interaction, basing this decision on many factors, including: the agent’s capability to effectively coordinate the current subtask, the other agent’s request to help, and the other agent’s capability to coordinate the interaction at the present time.

In Guinn’s capability model, conflicts of task initiative might occur when conversants overestimate their own capability or underestimate the other’s. Guinn (1998) proposed a negotiation model to resolve these conflicts. Conversants negotiate by informing each other of advantages of their solution and disadvantages of the other’s. By comparing each other’s plan, the conversant whose plan has the higher probability of success takes initiative. Guinn’s research on conflicts of task initiative, however, has little bearing on conflicts of dialogue initiative. For dialogue initiative, very often, one of the conversants just gives up the attempt very quickly, without giving a justification. As stated by Haller and Fossum (1999): “[dialogue initiative] conflicts are often simple clashes that result from both participants trying to take the initiative at the same time. Such conflicts do not necessarily require complex negotiation to resolve. Often, unwritten rules based on factors like social roles, personal assertiveness, and the current locus of control play a part in determining who will give away.” However, Haller and Fossum did not further investigate how conversants efficiently resolve conflicts of dialogue initiative.

2.3.3 Mixed-Initiative in Human-Computer Dialogue

Hagen (1999) took a simple viewpoint of mixed-initiative in dialogue systems. Hagen claimed that taking the initiative is equal to taking the turn. A dialogue system is mixed-initiative if the user is allowed to over-answer questions. Hagen used a dialogue grammar to represent a dialogue, similar to a parsing tree. The conversant at each node has control over the sub-tree. Hagen’s viewpoint on initiative and control seems to only apply to limited-domain human-computer interaction.

Smith (1993) studied mixed-initiative in dialogue systems. Smith implemented four different initiative strategies with differing amounts of control by the system: directive, suggestive, declarative, and passive. In the strategy of directive, the system has complete

control and the user is only allowed to answer questions. In suggestive, the system has control, but the user is allowed to initiate a sub-goal, for example, to ask a clarification question. In declarative, the user has dialogue control, but the system can provide relevant facts as a response to the user's statements. In passive, the user has complete control. Smith found that novice users would like to follow the lead of the system, but when they get experienced, expert users prefer more control.

Allen (1999) proposed four levels of mixed-initiative in dialogue systems. The first level is single-initiative, in which only one conversational party (the system or the user) is allowed to show initiative, and the other makes acknowledgements or answers questions. The second level is sub-dialogue mixed-initiative: it is similar to single-initiative in that control is pre-assigned to a conversational party, but the other party is allowed to initiate a sub-dialogue, such as asking a clarification question. The third level is fixed-subtask mixed-initiative: for each sub-task, either the system or the user is pre-assigned control over it. The highest level is negotiated mixed-initiative. Each party evaluates a list of factors, such as each other's capability over the current sub-goal, and then negotiates who controls the dialogue. It is generally agreed that negotiated mixed-initiative is the future trend for dialogue systems.

2.4 Modeling Multi-Tasking in Dialogue Systems

Current computer technology allows the user to perform multiple tasks at the same time, for example, to browse the web while installing and configuring software. From time to time, the software task needs the user's attention switched between tasks, for example, to click a button or to fill in some information. McFarlane (1999) proposed four strategies of task switching: immediate, negotiated, mediated, and scheduled. The first strategy, immediate, is that whenever an interruption request arrives, the active task will be switched to the interruption task immediately. The negotiated strategy is that the user is notified of an interruption request (e.g. by a flash) and it is the user's decision when to switch to the interruption task. The mediated strategy is to use an intelligent software agent to help the user make the decision when to switch to the interruption task. The scheduled

strategy is to switch tasks at a fixed time interval, for example, every 50 seconds.

Researchers are also building next-generation spoken dialogue systems that support multi-tasking (Lemon and Gruenstein, 2004; Kun, Miller, and Lenharth, 2004). Kun et al. (2004) developed a system called Project54, which allowed a user to interact with multiple devices in a police cruiser using speech. The architecture of Project54 allowed for handling multiple tasks overlapped in time. For example, when pulling over a vehicle, an officer can first issue a spoken command to turn on the lights and siren, then issue spoken commands to initiate a data query, go back to interacting with the lights and siren (perhaps to change the pattern after the vehicle has been pulled over), and finally receive the spoken results of the data query. While the current implementation of Project54 assumed that the officer initiates the task switching (e.g. the one about lights and the one about records), the system could initiate task switching too. However, Project54 did not provide infrastructure for signaling to the officer a system-initiated switch. Any such signaling would have to be hand-coded by developers.

Lemon et al. (2004) also explored multi-tasking in a dialogue system. They built a multi-tasking dialogue system for a human operator to direct a robotic helicopter on executing multiple tasks, such as searching for a car and flying to a tower. The system kept an ordered set of active dialogue tasks, and interpreted the user utterance in terms of the most active task for which the utterance made sense. Conversely, during the system's turn of speaking, it could produce an utterance for any of the dialogue tasks. Thus the system did not take into account the user's cost of task switching. The system switched to a new task as soon as it arose, instead of at an opportune place to minimize the user's effort. Moreover, the system did not signal when it switched between tasks. As with the approach of Kun et al. (2004) to multiple devices, it is unclear whether an actual user will be able to understand such conversations. The user might become confused about which task the system is on.

In all, current existing multi-tasking systems freely switch among different tasks as soon as a new task arises, and without much signaling. Thus the user might be confused about which task the system is talking about. Multi-tasking dialogues, even in the best of circumstances, will be difficult for users, as users need to remember the details of each

task and be aware of task switches. More studies on human-human multi-tasking dialogue are needed to better understand the human conventions, which will provide guidelines for building more user-friendly dialogue systems.

2.5 Computer Simulation in Dialogue

To better apply human conventions to spoken dialogue systems, we not only need to understand what people do in conversation (a descriptive account), but also why people are doing so (an explanatory account). Analyzing human-human dialogues, however, only reveals a descriptive account of human conventions. It does not reveal the reasons that underlie the human conventions (Walker, 1996). Computer simulation has been a promising tool for looking inside human conventions (Power, 1979; Carletta, 1992; Walker, 1996; Guinn, 1998). Human languages have evolved over generations. There must be some benefits for people to adopt certain conventions, such as to minimize collaborative effort (Clark and Wilkes-Gibbs, 1986). Given appropriate models of communication, reasoning, and solution quality, computer simulation can help researchers understand human conventions at a deeper level by justifying the adoption of them.

Power (1979) is probably the first to use computer simulation to study task-oriented dialogues. In Power's simulation experiment, the two agents that talked to each other were two processes of an identical program, although they might have different capabilities or different beliefs. For example, one was able to execute some actions, and one was able to observe the world. One of the agents, say A, had a domain goal to achieve, such as to enter a room. A employed a top-down planner to decompose the domain goal into sub-goals. For a sub-goal, say G, that A was not able to achieve by itself, A conducted a conversational procedure to involve the other agent, say B. G became a joint-goal if A succeeded in making B commit to it. On the other hand, when achieving G, B could also start a conversational procedure to get A's help. Each agent used a stack to represent the hierarchical structure of sub-goals. Dialogue context thus was represented as the goal stack plus the planning tree. Power's simulation experiment reflected the claim that dialogue is a bi-product of domain planning (Cohen and Levesque, 1994; Cohen and Perrault, 1979).

It gave a formal account of interpreting the intention of an utterance in the context of the goal stack.

Carletta (1992) simulated the behavior of *planning to fail*. Analysis of the MapTask corpus (Carletta et al., 1997) showed that, when achieving a goal, sometimes people convey information as specific as possible in the initial utterance to avoid failure; while sometimes people include some abstract information in the initial utterance, and rely on the other conversant to initiate a repair process to clarify what is not clear. The latter was referred to as *planning to fail* because the conversant knows that it has a high risk of failure and probably a repair process is needed. Carletta found that planning to fail is a frequent dialogue phenomenon, and argued that people adopt this strategy for the purpose of minimizing collaborative effort (Clark and Wilkes-Gibbs, 1986) because the repair process is relatively inexpensive. Carletta simulated the behavior of planning to fail for the purpose of better understanding how to implement this strategy in a dialogue system.

Walker (1996) studied *information redundant units* with computer simulation. Conversants often re-state information that can be easily derived from the context. Walker called these utterances *information redundant units*. Walker set up the Design World domain, in which two software agents communicated to arrange furniture in two rooms with certain constraints. Each furniture item was given some points. Walker also adopted a limited memory model for the software agents so that agents might “forget” things, which better simulated human’s memory and also quantified the calculation of memory effort. Walker calculated communication effort as the length of a dialogue, and calculated reasoning effort as the steps in the reasoning. Agents’ performance was calculated as the points of furniture less the sum of communication effort, memory effort, and reasoning effort. Walker showed that information redundant units, under certain situations (for example, when agents had a limited memory), led to higher performance. Walker concluded that information redundant units conform to the theory of least collaborative effort.

Guinn (1998) proposed the capability model and negotiation model for task initiative, as described in Section 2.3. Guinn then set up the Murder Mystery domain, in which two software agents exchanged information and reasoned about the murderer from twenty-two

suspects. Each agent had some knowledge about the suspects, such as whether the suspect had a motif or access to weapons. The murderer was the one who met all conditions of murder. Agents discussed each suspect to evaluate whether the conditions were met. Using computer simulation, Guinn showed that the capability model and the negotiation model resulted in a (locally) optimized order of suspects being discussed, i.e., a suspect with higher probability of being the murderer would be discussed earlier, thus less suspects were discussed to find out the murderer. Guinn concluded that the models lead to more efficient problem solving.

These simulated dialogues, by Power, by Carletta, by Walker, and by Guinn, were claimed to be mixed-initiative, i.e. each software agent was allowed to show initiative. However, the initiative strategies adopted in the simulation are by no means natural to humans. For example, in Guinn's simulation experiment, embedded question-answer pairs are the only communication protocol allowed, thus the person who brings up a goal always abandons initiative right after. Moreover, software agents need to ask a question to explicitly give up initiative, while in human-human dialogues people can implicitly give up initiative by just keeping silence. Such implementations might distort what mixed-initiative is, because from a human observer's point of view, the coherence of the simulated dialogues is not guaranteed.

Furthermore, Power, Carletta, Walker, and Guinn, only gave some illustrating examples of the human conversational phenomena that they simulated. None of them gave a full statistics of such phenomena in human-human dialogue. It is thus unclear whether people really execute such conventions in their conversations, and to what extent. Computer simulation results should be taken as secondary evidence to support the observations from human-human conversations. We are far from truly understanding human memorizing and human reasoning from the perspective of cognitive science. The state of the art in artificial intelligence technology also does not allow complicated human-style reasoning. Assumptions that might not truly reflect human behavior are made in computer simulation to simplify the problem, which might lead to biased results. Without statistical analysis of human-human dialogues, computer simulation alone is not sufficient.

2.6 Chapter Summery

People take turns to speak. Turn-taking is a basic mechanism of interaction in conversation, but turn-taking does not account for how people solve a complex goal. Grosz and Sidner’s discourse structure theory described how people decompose a complex goal into a set of sub-goals, but failed to account for how people direct the conversation flow to achieve the sub-goals. Current initiative theories, on the other hand, focused on how people direct the conversation flow but were vague on the relation between initiative and discourse goals. This thesis research thus aims to fill in the gap by tying initiative with discourse goals: how people collaborate in directing the conversation flow to solve a complex goal.

Little is known about how people manage multiple tasks in their conversations, although some initial effort was made to develop multi-tasking computer dialogue systems. These systems, however, freely switched between tasks as soon as a new task arrives without considering whether the current discourse context is opportune for task switching, which might confuse the user or burden the user with higher cognitive effort. Moreover, these systems switched between tasks without much signalling. Previous research found that people use discourse markers and prosody to signal topic shifts, which suggest that people might be signalling task switching as well. This thesis aims to understand how people direct the conversation flow with multiple tasks, including the discourse context where task switching happens and the cues that conversants use to signal task switching.

Mixed-initiative and multi-tasking dialogues open up the possibility of initiative conflicts, where both conversants speak at the same time, trying to drive the conversation towards different directions. Current research of fighting for the speaking turn provided some insights of initiative conflicts, but lacked statistical evidence. Moreover, research of turn-taking conflicts did not provide clear distinction between cues that are the cause of resolution and those that are the effect of resolution. Existing initiative models did not address initiative conflicts either. This thesis also aims to understand how initiative conflicts happen, and if they happen, how they are resolved and the conversation is repaired afterwards.

Chapter 3

DialogueView: An Annotation Tool

In order to run empirical studies on human conventions of initiative and control, we need to annotate and analyze human-human dialogues. This chapter describes DialogueView, a tool for annotating dialogues with utterance boundaries, speech repairs, utterance tags, and hierarchical discourse structure.¹

3.1 Introduction

There is growing interest in collecting and annotating corpora of language use. Annotated corpora are useful for formulating and verifying theories of language interaction, and for building statistical models to allow a computer to naturally interact with people.

There are a number of different annotation tasks, such as transcribing what words were said, marking POS tags, disfluencies, utterance boundaries, speech acts, co-references, and hierarchical discourse segments. A number of tools have been developed for these annotation tasks. Depending on the annotation task, the tools take different input, and present the input in different ways. For example, for word transcription, they give users a view of the audio signal with all annotations time-aligned to it. For annotating speech acts, co-reference, and hierarchical discourse segments, the tools tend to take a list of utterances as input, and give users a view that shows the interaction utterance by utterance, like a movie script. These two views let users focus on different aspects of the data, such as the exact timing of audio events, or on an abstracted level, where the user can see a larger

¹Content of this chapter was previously published in (Heeman, Yang, and Strayer, 2002), (Yang, Strayer, and Heeman, 2002), and (Yang et al., 2008).

portion of the dialogue and how an utterance fits into the context.

Annotating with a single view, however, is inadequate. First, an annotation task might require more than one view of the data, as a single view abstracts away some of the details that the other view will show. For example, annotating discourse segments needs a larger perception of the interaction, as afforded by an utterance-by-utterance view. However, to determine whether an utterance, such as “okay”, belongs to the end of a segment or signals a new one, it is helpful for the user to see the exact timing of the “okay”; if the “okay” is distant from the previous utterance, then this is evidence (along with intonation) that the “okay” begins the next segment. Thus seeing multiple views can be helpful for an annotation task.

The second reason why a single view is inadequate is that users need the ability to do multiple tasks at the same time. Although best practice calls for doing each annotation task separately (Bernsen, Dybkær, and Kolodnytsky, 2003), this best practice needs to take into account the interactions between tasks. It might only be when doing one annotation task that one realizes a mistake in another. For example, one could first segment a dialogue into utterances, then annotate disfluencies, and then tag the utterances with speech acts. The problem is that the tasks of annotating disfluencies and utterance boundaries are highly intertwined, and it is only by considering both tasks that one can make sense of each (Heeman and Allen, 1999). A second example is with utterance segmentation and speech act tags. What constitutes an utterance often depends on what can be annotated with the speech act tags (Allen and Core, 1997). Thus in doing one annotation task, users might need to modify annotations done previously. These annotation tasks might require different views.

In order for users to have multiple views, they could launch multiple tools. This is problematic for the following two reasons. First, when users reposition the part of data that one tool is showing, they have to manually reposition the second tool as well, which is cumbersome. Second, when users change an annotation in one tool, this change will not propagate into the second tool. Moreover, if the changed annotation is part of the input to the second tool, users might have to redo all of the annotations they had done with the second tool. Hence, a single tool that shows the interaction in multiple views is needed.

A single tool with multiple views does not need to show all of the information in each view; rather, each view can use a subset of the information. We refer to the data used in a view as an *abstraction*. The abstraction can even include arbitrary functions on the data. A simple example is an abstraction for an audio-level view, which might not include discourse blocks or all utterance tags. In contrast, the abstraction for an utterance-by-utterance view uses the utterance segmentations from the audio-level view to display the words segmented by utterance units, abstracting away the exact timing of the words. Such an abstraction might also exclude some words, or even some utterances, depending on the annotation task. With abstractions, the information in each view can be chosen so that it complements what is shown in the other views.

The annotation tool, DialogueView, consists of three views: WordView, UtteranceView, and BlockView. These three views present different abstractions, which help users better understand what is happening in the dialogue. WordView shows the words time-aligned with the audio signal. UtteranceView shows the dialogue as a sequence of utterances. It abstracts away from the exact timing of the words and can even skip words, based on WordView annotations, that do not impact the progression of the dialogue. BlockView shows the dialogue as a hierarchy of discourse blocks, and abstracts away from the exact utterances that were said. Annotations are done at the view that is most appropriate for what is being annotated. The tool allows users to easily navigate among the three views and it automatically updates all views when changes are made in any one view.

3.2 Existing Tools

There are many existing tools for annotating dialogue. We categorize them by how they present a dialogue to users.

3.2.1 Audio View Tools

The first category of tools displays the waveform of the audio file and allows users to make annotations time-aligned to the audio signal. This category includes *Emu* (Cassidy and Harrington, 2001), *Speech Viewer* (Sutton et al., 1998), *Praat* (Boersma and Weenink,

2005), and *WaveSurfer* (Sjölander and Beskow, 2000). These tools often allow multiple annotation tiers, which can be used for different annotation tasks. For instance, the ToBI annotation scheme (Pitrelli, Beckman, and Hirschberg, 1994) uses one tier for the word transcription, a second for intonation tags, a third for break indices, and a fourth for miscellaneous information, including disfluencies. These tools often have powerful signal analysis packages for displaying such phenomena as the spectrogram, pitch contour, and voicing. Some tools display a video as well, such as *Anvil* (Kipp, 2001).

These tools are intended for tasks like word and phoneme transcription, acoustic event annotation (such as intonational phrasing and word accents), and utterance segmentation. However, these tools are not good for coding higher level information, such as speech acts or hierarchical discourse structure, because these tasks require that users understand a much larger context. For these tasks, often there is no need to see the exact timing of each word or to see the abandoned speech (which does not contribute to the dialogue); instead, users want to see how utterances are segmented, what they are intended to convey, and how they are related to each other. Thus users need a view that suppresses information that is not directly relevant, such as the exact timing of words, and emphasizes what is important, such as utterance segmentation.

3.2.2 Utterance-by-Utterance View Tools

The second category of tools displays a dialogue as a sequence of utterances and formats the utterances line-by-line, like a movie script. These tools are used for annotating the higher-level structure of dialogue. *DAT* allows users to annotate utterances or groups of utterances with the *DAMSL* scheme (Allen and Core, 1997; Core and Allen, 1997). *MMAX* allows for multiple annotation tasks, such as speech acts and co-reference (Müller and Strube, 2003). *N.b.* allows users to annotate discourse structure as a hierarchical grouping of utterances (Flammia, 1998). All three tools take as input a dialogue as a sequence of speakers' utterances. This simplification can make it difficult for users to truly understand what is happening in the dialogue, especially when there are disfluencies, abandoned utterances, and overlapping speech.

3.2.3 Tools with Multiple Views

Transcriber gives users two views of a dialogue (Barras et al., 2001). It takes as input an audio file of a dialogue. The user can mark utterance boundaries and transcribe what was said in each utterance. A second view, which organizes the dialogue as a sequence of speakers' utterances, allows users to annotate them with speech acts and to group them into topics.

Transcriber, however, lacks support for annotating some common dialogue phenomena. First, *Transcriber* is weak in handling overlapping speech. If there is overlapping speech, users must treat the overlap as a special utterance, spoken by both speakers. For some typical cases of overlap (see Section 3.4.1), this creates a convoluted interpretation of the dialogue. Second, *Transcriber* does not provide direct facilities for annotating disfluencies. Third, although *Transcriber* offers flat (topic) segmentation, hierarchical segmentation is not supported.

Transcriber has some basic abstraction mechanisms. The utterance-by-utterance view abstracts away the detailed timing of words for coding speech acts and discourse blocks. However, more powerful abstractions, such as abstracting away disfluencies, abandoned speech, and overlapping utterances, are not provided.

3.2.4 Toolkits

In addition to existing tools, there are several toolkits that allow researchers to build tools for a wide range of annotation tasks. *AGTK* (Bird et al., 2001) is a toolkit based on a knowledge representation scheme called *Annotation Graphs* (Bird et al., 2000). The toolkit provides functions for reading and writing annotations, displaying a waveform, transcribing a segment of speech, and annotating segments with multiple sets of labels. These modules can be used with a programming language (C++, Tcl/Tk, or Python) to build an annotation tool.

A second toolkit is the *MATE Workbench* (McKelvie et al., 2001). Users specify their annotation task using the MATE style sheet language, which is an XML specification of (1) the data that users will annotate, (2) how to annotate the data, and (3) how to display

the annotated data. This toolkit can be used for creating tools that let users perform multiple annotation tasks, including part-of-speech tags, co-reference, and utterance tags.

A third toolkit is the *NITE XML Toolkit*, which is intended for annotating multi-modal data (Carletta et al., 2003). This toolkit provides two ways of building an annotation tool. First, similar to AGTK, the toolkit provides modules that can be used with the Java programming language to build an annotation tool. Second, similar to MATE, the toolkit can take as input a style sheet to create an annotation tool.

These toolkits suffer the following problems. First, they do not provide built-in support for a number of annotation tasks, such as annotating disfluencies, overlapping speech, and hierarchical discourse structure. Second, the toolkits do not advocate nor provide built-in abstraction mechanisms useful for understanding and annotating dialogue. No support is provided for abstracting out disfluencies, abandoned speech, and overlapping utterances in an utterance-by-utterance view; nor is support provided for displaying annotations in a time-aligned audio view. Although the toolkits are flexible enough that users can program almost any functionality that is needed, it is unclear how much effort it would take to implement the functionalities listed above in order to build a tool as powerful as DialogueView.

3.3 The Basics of DialogueView

In this section, we describe the basic features of DialogueView, which encapsulates the features provided by other tools such as Speech Viewer, DAT, N.b., and Transcriber. DialogueView provides the user with the following key features:

- Annotation can be done at the view that is most appropriate
- Each view abstracts information away from other views
- Changes in one view propagate to others
- Views can be synchronized
- Users can see the dialogue

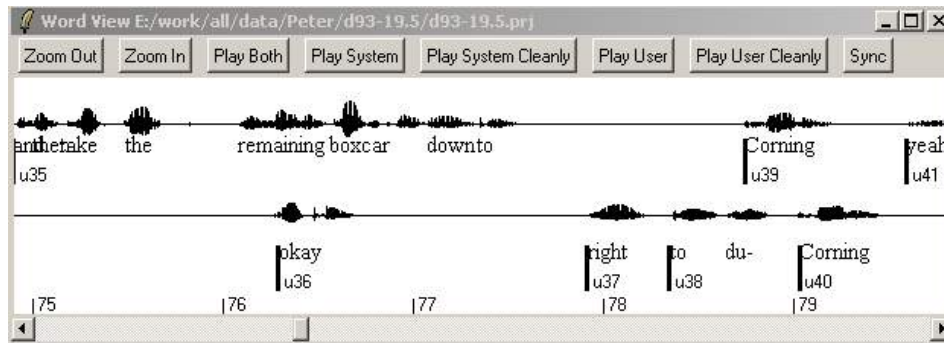


Figure 3.1: Utterance segmentation in WordView

- Users can hear the dialogue

3.3.1 WordView

The first view in DialogueView is *WordView*, which takes as input two audio files, the words said by each speaker together with their start and stop times (in XML format), and shows the words time-aligned with the audio signal. Figure 3.1 shows WordView with an excerpt of a dialogue from the Trains corpus (Heeman and Allen, 1995b). This view is ideal for seeing the exact timing of speech, especially overlapping speech.

WordView gives users the ability to select a region of the dialogue and to play it. Users can play each conversant channel individually (*Play System* and *Play User*) or both combined (*Play Both*). This capability is especially useful for overlapping speech, where users might want to listen to what each conversant said individually, as well as to hear the overlap between the conversants. The *Play System Cleanly* and *Play User Cleanly* buttons provide functionality to play the speech with disfluencies removed, which will be addressed in Section 3.4.3.

3.3.2 Segmenting Word Sequences into Utterances

One function of WordView is to allow users to break up the speech of each conversant into sections, which we refer to as *utterances*. Figure 3.1 shows a dialogue excerpt segmented into utterances. A vertical bar indicates the start time of the utterance. The end time is not explicitly shown; it is simply the end time of the last word in the utterance. In the

example, the top conversant’s utterances were “and then take the remaining boxcar down to”, “Corning”, and “yeah”, and the bottom conversant’s utterances were “okay”, “right”, “to du-”, and “Corning”. Users can easily move, insert or delete utterance boundaries. WordView ensures that boundaries never fall in the middle of a word.

As can be seen in the example, utterances of one conversant are allowed to overlap with the utterances of another conversant. This allows us to easily annotate back-channels, which often overlap with the other conversant’s utterances, as illustrated by the bottom conversant’s “okay” at 76.3 seconds. This example would be cumbersome for the tool Transcriber (Barras et al., 2001) to handle.

In our own annotation scheme, we have defined an *utterance* as a semantically meaningful contribution that is not affected by subsequent speech from the other conversant (Heeman and Allen, 1995a). We felt that the back-channel of the bottom conversant during the top conversant’s utterance of “and then take the remaining boxcar down to” did not change the top conversant’s actions. However, we felt that the bottom conversant’s attempted completion of the top conversant’s utterance did have an effect on the top conversant, which is why the top conversant’s speech was broken into two utterances.

3.3.3 UtteranceView

The annotations in WordView are utilized in building the next view, *UtteranceView*. This view shows the utterances of two conversants as if it were a script for a movie. To derive a single ordering of the utterances of the two conversants, we use the start time of each utterance as annotated in WordView. We refer to this process as *linearizing* the dialogue (Heeman and Allen, 1995a). The order of the utterances should show how the conversants are sequentially adding to the dialogue, and is our motivation for defining utterances as being small enough so that they are not affected by subsequent speech of the other conversant. Figure 3.2 shows UtteranceView with the dialogue excerpt from Figure 3.1.

UtteranceView abstracts away from the detailed timing information of the words that were said. Instead, it focuses on the sequence of utterances. This view is ideal for annotations that require a larger context of what is happening in the dialogue, such as speech act annotation. Of course, if users want to see the exact timing of the words in the utterances,

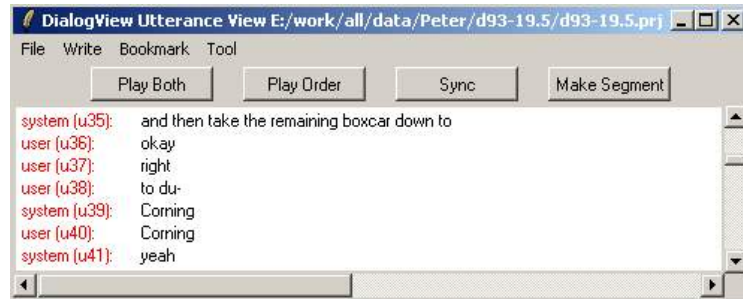


Figure 3.2: Utterance segmentation in UtteranceView

they can examine WordView, as it is displayed alongside UtteranceView. Although both views have scroll bars that allow the user to independently position them anywhere in the dialogue, both have a navigation button (*Sync*) that repositions the other view to the same place. Furthermore, changes made in WordView are immediately propagated into UtteranceView, and hence users will immediately see the impact of their annotations. For example, if a user were to delete the utterance boundary between “to du-” and “Corning” in WordView in Figure 3.1, then “to du- Corning” would be shown as a single utterance in UtteranceView.

UtteranceView also includes playback buttons. To playback, the user first highlights a sequence of utterances with the mouse. The user can then press *Play Both* to hear the audio starting from the start time of the first highlighted utterance to the end time of the last highlighted utterance. This is similar to *Play Both* in WordView. The second playback, *Play Order*, is slightly different. It takes the linearization into account and dynamically builds an audio file in which each utterance in turn is concatenated together, and a 0.5 second pause is inserted between each utterance. This gives the user an idealized rendition of the utterances, with overlapping speech separated. In the example from Figure 3.2, utterances **u38** through to **u40** would be each played sequentially, rather than with overlapping between **u38** and **u39**, and **u39** and **u40**. By comparing the *Play Both* and *Play Order* playbacks, users can aurally check if their linearization of the dialogue is correct.

UtteranceView also has a print facility. This produces a postscript file with the script-like rendition of the dialogue as it is displayed in UtteranceView.

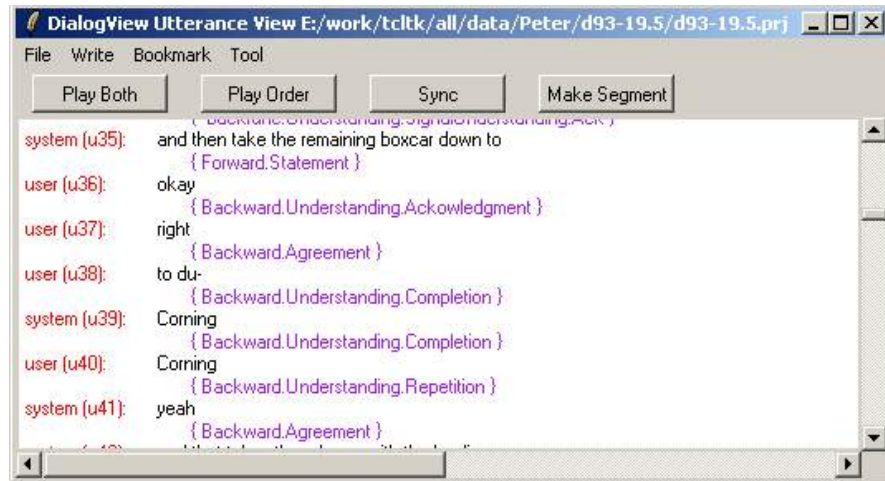


Figure 3.3: UtteranceView with utterance tags displayed

3.3.4 Annotating Utterances

In either UtteranceView or WordView, users can annotate utterances with various tags. Although the notion of segmenting word sequences into utterances is built into the tool, what the utterances can be annotated with is not. The annotation scheme is specified in a configuration file. This makes the tool useful for users who want to apply different annotation schemes, and makes it easy to experiment while developing a scheme. Utterances can be annotated in either WordView or UtteranceView. WordView is more suitable for tags that depend on the exact timing of the words, or a very local context, such as whether an utterance is abandoned or incomplete (see Section 3.4.2). UtteranceView is more suitable for tags that relate the utterance to other utterances in the dialogue, such as whether an utterance is an answer, a statement, a question, or an acknowledgement. Here, the exact timing of the words is not important, but the larger context is, which UtteranceView shows. Figure 3.3 shows UtteranceView with the utterance tags displayed underneath each utterance. Whether an annotation tag can be used in WordView or UtteranceView (or both) is specified in the configuration file. Which view a tag is used in does not affect how it is stored in the annotation files.

To illustrate how the configuration file works, we present a simplified version of the DAMSL annotation scheme (Allen and Core, 1997; Core and Allen, 1997). Utterances can

be annotated in `UtteranceView` with tags of forward function, backward function, and a comment. Forward functions include statement, information request, and suggestion. Backward functions include agreement, answer, and understanding, which can be either acknowledgement, repetition, or completion. The following shows the specification of these tags in the configuration file, which is simply a text file that can be edited in any text editor.

```
UttrTags@UttrView = anyof Forward Backward Comment
UttrTags.Forward  = anyof Statement InfoRequest Suggestion Other
UttrTags.Forwad.Other.<type> = string
UttrTags.Backward = anyof Agreement Answer Understanding Other
UttrTags.Backward.Other.<type> = string
UttrTags.BackWard.Understanding = oneof Acknowledgement Repetition Completion
UttrTags.Comment.<type> = string
```

The first line specifies what utterance tags should be selected in `UtteranceView`, namely *Forward*, *Backward* and *Comment*. The keyword *anyof* specifies that any number of the items on the righthand side can be annotated, while the keyword *oneof* (as in line 6) specifies that only one member of the righthand side can be selected. Tokens on the righthand side can be further decomposed, as we see for *Forward* in line 2. Tokens are assumed to be of type boolean unless otherwise specified; see, for example, *UttrTags.Forward.Other* in line 3. `DialogueView` will use the configuration file to create the annotation panel shown in Figure 3.4, which will pop up if the user double-clicks on an utterance in `UtteranceView`.

An added feature of `DialogueView` is that it allows users to quickly locate particular words or utterance tags in a dialogue, just as one can search for words in a text file. Both `WordView` and `UtteranceView` support this search feature. This feature can be used to find interesting utterances or patterns. For example, one might be interested in analyzing discourse markers, such as “like” or “well”; users could search for the word they are interested in and quickly view every instance of it in the dialogue, along with the context surrounding the word. Likewise, if they are interested in all utterances tagged as completions, they can easily find these as well.

Figure 3.4: Sample Utterance Annotation Panel

3.3.5 Grouping Utterances into Blocks

In UtteranceView, users can annotate hierarchical groupings of utterances. We call each grouping a *block*, and blocks can have other blocks embedded inside of them.

This blocking function is similar to what N.b. provides (Flammia, 1998), but rather than showing the structure with indentation and color, we use boxes to show the extent of each block. Figure 3.5 shows a dialogue excerpt with discourse blocks annotated. To create a segment, users highlight a sequence of utterances and then press the *Make Segment* button. Users can change the boundary of a block by simply dragging either the top or bottom edge of the box. The tool ensures that block edges are never moved in such a way as to cause two blocks to be interleaved, where one is not entirely contained in another.

DialogueView supports hierarchical blocks because many linguistic theories of discourse argue that conversation is organized as a hierarchical tree-like structure (e.g. (Reichman-Adar, 1984; Grosz and Sidner, 1986; Polanyi, 1988; Mann and Thompson, 1987; Carberry and Lambert, 1999)). These theories propose that conversants can continue talking about a topic, introduce a new subtopic, or finish with the current one. A few researchers, however, have proposed that conversants can interleave topics, as when comparing two different options (e.g. (Ramshaw, 1991; Rose et al., 1995; Hagen, 1999)); however, DialogueView currently does not support such structures.

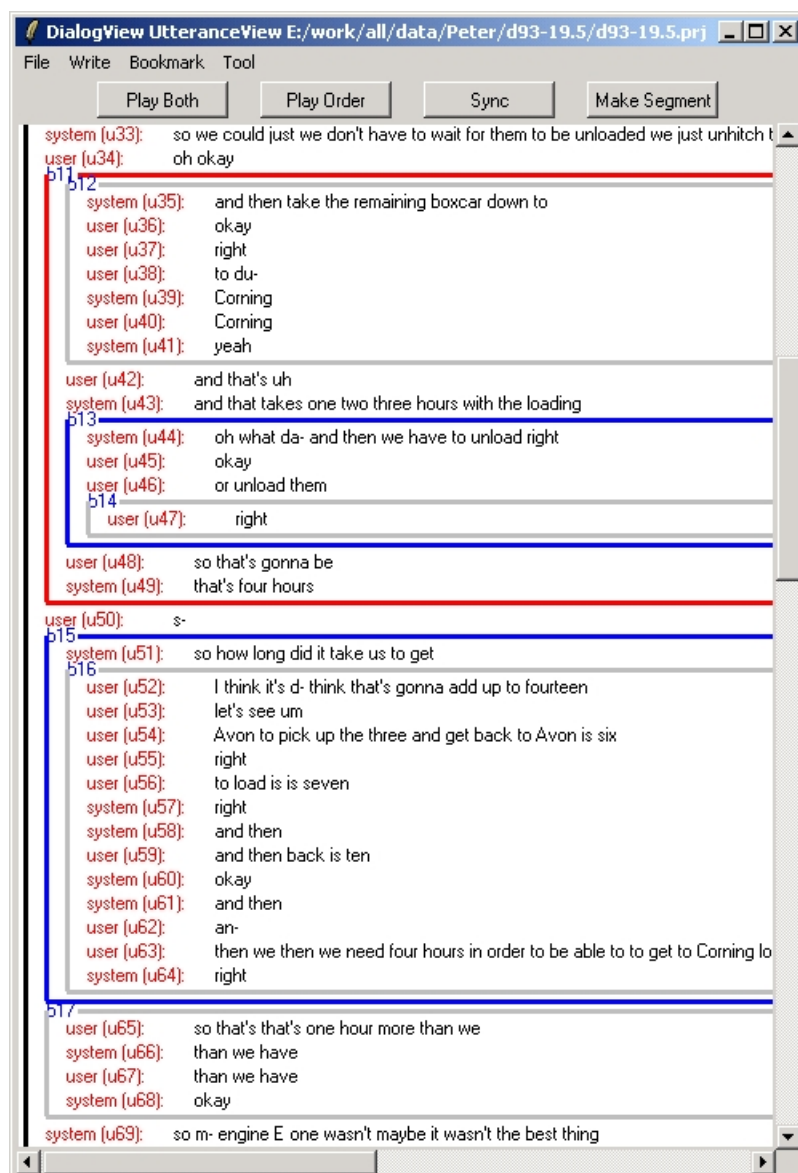


Figure 3.5: Grouping utterances into blocks in UtteranceView

3.3.6 Annotating Blocks

Just as utterances can be tagged, so can discourse blocks. Many researchers of discourse theory have proposed that there are different types of discourse blocks. For instance, Carletta *et al.* (1997) distinguish between dialogue games and transaction blocks; Traum and Hinkelman (1992) classify blocks by their purpose, such as *construct plan*, *summarize*, *elaborate*, and *clarify*. In Grosz and Sidner’s theory (1986), discourse blocks also have a purpose associated with them. In DialogueView, users can define the block tags as appropriate for their own annotation scheme in the configuration file. In fact, in our annotation scheme, we annotate all of the aspects discussed above. A simplified version of our scheme is shown in the following.

```
BlockTags      = anyof Type Task Purpose
BlockTags.Type = oneof Transaction DialogueGame SecondPart Grounding
BlockTags.Task = oneof Greeting SpecifyGoal ConstructPlan Summarize Verify
BlockTags.Type.Transaction.<color>   = red
BlockTags.Type.DialogueGame.<color>  = blue
BlockTags.Type.SecondPart.<color>    = grey
BlockTags.Type.Grounding.<color>     = grey
BlockTags.Purpose.<type>               = string
```

Note that we use color to distinguish different block types; for example, red for transaction blocks and blue for dialogue games.

3.4 Further Abstractions

In Section 3.3.3, we discussed how UtteranceView abstracts out the detailed timing information from WordView. UtteranceView allows users to focus on a larger context, which is often necessary for annotating utterance tags and discourse structure. In this section, we present additional mechanisms for abstracting away details, which we feel will help users better understand what is happening in a dialogue, and thus help them better annotate it for utterance tags and discourse structure. We illustrate these abstractions on the dialogue excerpt from Figure 3.5. The abstractions of Section 3.4.1, 3.4.2 and 3.4.3 result in the UtteranceView shown in Figure 3.10. Sections 3.4.4 and 3.4.5 offer further abstractions

yet. Note that all these abstractions are optional to users.

3.4.1 Overlapping Speech

There are cases where both conversants talk at the same time. Capturing this overlap properly is critical for understanding what is happening in the dialogue. Hence, it is imperative that users can properly annotate overlap and that the tool properly displays the overlap. We classify overlap into three types, and provide a mechanism in `DialogueView` for effectively dealing with each type.

The first type of overlap is where the beginning of one utterance overlaps with the ending of the previous utterance. In Figure 3.1, the second instance of “Corning” (by the bottom speaker) overlapped slightly with the first instance of “Corning” (by the top speaker). Such overlaps are very common in dialogue, as conversants are very good at anticipating the end of the other’s turn (Beach, 1991). These overlaps are not problematic, as `UtteranceView` will display them in the proper order.

The second type of overlap is where the overlapping utterance is responding to what the other speaker is currently saying, but only the first part of it. For instance, in Figure 3.1, the bottom speaker said “okay” during the middle of an utterance, perhaps to tell the top speaker that everything was understood so far. However, in the `UtteranceView` of Figure 3.5, it may appear that the “okay” of **u36** is acknowledging the entire utterance, rather than just a part of it. Hence, users could be misled as to what is happening. In order to remove this source of confusion, we have an utterance tag for this type of overlap, which we annotate in `WordView`. The following shows the specification of the overlap tag in the configuration file.

```
UttrTags@WordView = any Overlap
UttrTags.Overlap.WordView.<color>          = blue
UttrTags.Overlap.UtteranceView.<prefix>    = {  +}
UttrTags.Overlap.UtteranceView.<postfix> = +
```

In order that users can easily see what has been tagged as overlap, we do the following: in `WordView`, we color the utterance segmentation marker in blue; in `UtteranceView`,

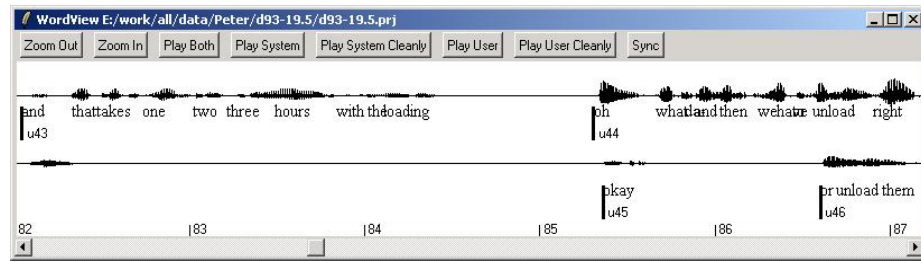


Figure 3.6: Misleading utterance segmentation

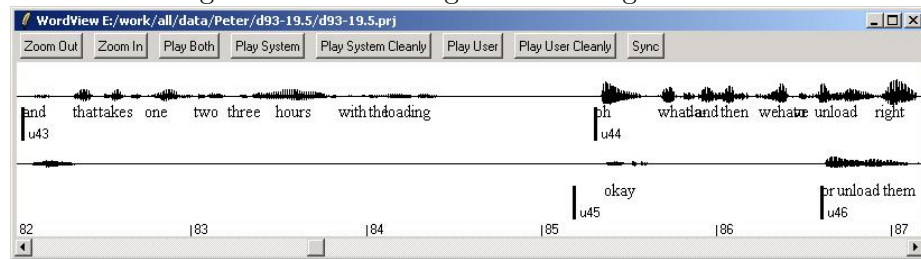


Figure 3.7: Altered start time for “okay” utterance

we indent the utterance and put ‘+’ on either side (similar to the Chiles scheme, cf. MacWhinney 2000).

The third type of overlap occurs where one conversant is responding to something that the other said, but responds after the other has resumed speaking. This is illustrated in Figure 3.6, where the top conversant finished utterance **u43** and then paused for over a second. The bottom conversant then acknowledged the utterance with the “okay” of **u45**, but this happened a fraction of a second after the top conversant started speaking again with **u44**. As shown in Figure 3.5, the linearization of the dialogue has the “okay” of **u45** following **u44** as if it were acknowledging that utterance. One could tag **u45** with the overlap tag. However, the fact that **u45** overlaps with **u44** is not critical in understanding what is occurring in the dialogue, as long as **u45** is linearized before **u44**. This can be done in WordView by moving the start marker of **u45** so that it is before the start of **u44**, as shown in Figure 3.7.² After the start marker has been moved, UtteranceView is automatically reformatted as shown in Figure 3.10.

²The start marker can be moved provided that the speaker was silent in the time interval preceding the point when the other speaker started talking.

In summary, overlapping speech can be handled in three ways. The overlap can be ignored if it is not critical for understanding what is going on in the dialogue; the utterance can be explicitly tagged as overlap; or the start time of the utterance can be changed so that the overlap does not need to be tagged.

For the audio playback of *Play Order*, if there is overlap that has not been explicitly tagged as such, the utterances are simply concatenated together according to the start time, as explained in section 3.3.3. If the overlap is explicitly marked, we keep the overlap in the audio file. Users can thus aurally check if their treatment of the overlap is correct.

3.4.2 Incomplete Utterances, Abandoned Utterances, and Completions

In the previous section, we discussed how overlapping utterances can be annotated so that UtteranceView succinctly captures what is happening. In this section, we discuss three more utterance tags that can be used to better format UtteranceView, and thus help abstract away from the details in WordView. These tags are for abandoned utterances, incomplete utterances, and completions.

Abandoned: The conversants abandoned what they were saying and the speech had no impact on the rest of the dialogue. This mainly happens when one conversant loses the bid to speak next. Figure 3.5 shows a number of abandoned utterances, namely **u42**, **u46**, **u48**, **u50**, **u58**, and **u62**. Figure 3.8 shows two of these abandoned utterances: **u46** and **u48**. In **u46**, the bottom conversant tried to help the top conversant finish the utterance, but abandoned the attempt, which could be heard by a decreasing volume in the speech. In **u48**, the bottom conversant started to speak, but then the top conversant cut in, and the bottom speaker gave up the turn.

Incomplete: The conversant did not make a complete utterance, either intonationally, syntactically, or semantically. Unlike the previous category, incomplete utterances do impact the dialogue behavior. Figure 3.1 has two incomplete utterances: **u35** and **u38**. In both instances, the other conversant completes the utterance, which is a strong indicator that the utterance should not be tagged as abandoned.

Completions: The conversant attempted to complete the utterance of the other conversant. This is similar to the *co-operative completions* of (Linell, 1998). Completions are

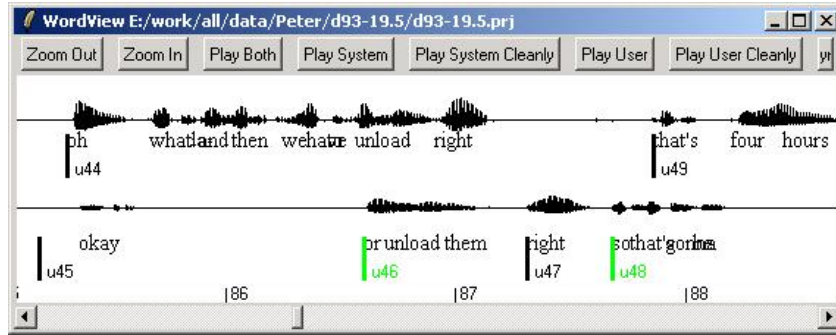


Figure 3.8: Abandoned Utterances

often viewed as evidence that an utterance was understood (Schober and Clark, 1989), which is why we included them as a subtype of the *Understanding* tag (see Section 3.3.4). Figure 3.1 gives two examples of completions: **u38** and **u39**.

Abandoned and incomplete utterances are tagged in WordView and completions in UtteranceView. However, regardless of where they are tagged, we want to use these tags to better format UtteranceView. Incomplete utterances are displayed with a trailing ellipsis marker “...” and completions with a preceding one. Figure 3.10 shows UtteranceView with these markings, which makes it much easier to see what is going on in the dialogue. As for abandoned utterances, these are not displayed in UtteranceView, as they do not impact the subsequent development of the dialogue. Hence, users do not need to see them in annotating utterance tags in UtteranceView and segmenting the discourse structure. Since abandoned utterances are not displayed in UtteranceView, they are also skipped in the audio playback of *Play Order*. The specification for these tags is given in the following:

```

UttrTags@WordView = anyof Overlap Abandoned Incomplete
UttrTags.Incomplete.UtteranceView.<postfix> = { ...}
UttrTags.Abandoned.WordView.<color>          = green
UttrTags.Abandoned.<display@UttrView>        = no
UttrTags.Completion.<prefix@UttrView>        = {... }

```

3.4.3 Speech Repairs

DialogueView has a special facility for annotating speech repairs. A speech repair is the disfluency where a speaker goes back and repeats or changes something that was just said (Heeman and Allen, 1999). The following gives an example of a repair. The *reparandum*

why don't we put um take two boxcars
 $\underbrace{\hspace{1.5cm}}$ $\uparrow$ $\underbrace{\hspace{1.5cm}}$ $\underbrace{\hspace{1.5cm}}$
 $\underbrace{\hspace{1.5cm}}$ *ip* $\underbrace{\hspace{1.5cm}}$ *et* $\underbrace{\hspace{1.5cm}}$ *alteration*

is the speech that is being replaced, and the *interruption point* (ip) is the end of the reparable. This can be followed by an optional *editing term* (et), such as “um”, “uh”, or “let’s see”, which helps marking the repair. The piece of speech that replaces the reparable is called the *alteration*.

As the annotation of speech repairs requires careful listening to the speech, and speech repairs have strong interactions with utterance boundaries, their annotation is best done in WordView. To annotate a repair, the user highlights a sequence of words and then tags the sequence as a reparable or an editing term. The user also chooses the type of repair. Figure 3.9 shows how speech repairs are displayed in WordView. The words in the reparable are underlined and displayed in a different color. The words in the editing term are displayed in a similar manner.

The speech repair scheme is specified in the configuration file. The following specifies the scheme used by Heeman and Allen (1999). The first line says that there are three types of repairs: cancel repair (*can*), modify repair (*mod*), and abridge repair (*abr*). The next three lines specify the constituents of each type of repair, in terms of *Reparandum*

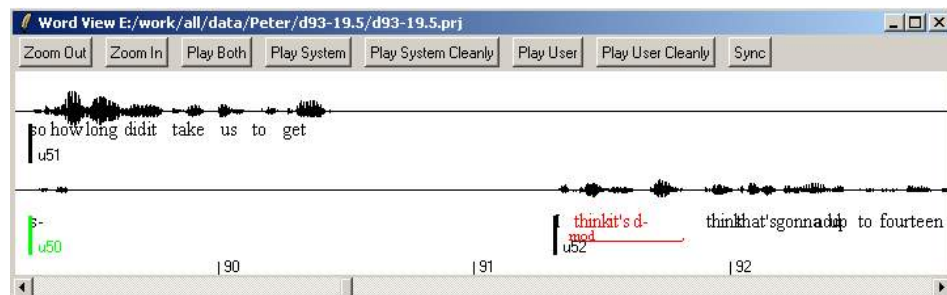


Figure 3.9: Speech Repair Example

and *EditingTerm*, which are reserved words.

```
SpeechRepair      = oneof can mod abr
SpeechRepair.can = consistsof Reparandum EditingTerm
SpeechRepair.mod = consistsof Reparandum EditingTerm
SpeechRepair.abr = consistsof EditingTerm
```

After a speech repair is annotated in *WordView*, there is little need to show the repaired words in *UtteranceView*. Hence, we show only the cleaned up speech in which we have removed the reparandum and editing terms of speech repairs. This is what accounts for the difference between Figures 3.5 and 3.10 in utterances **u33**, **u43**, **u44**, **u52**, **u54**, **u63**, **u65**, and **u69**. As can be seen, the cleaned up utterances are easier to understand than the original ones. Note that after a user annotates a speech repair in *WordView*, *UtteranceView* is immediately updated, which lets the user visually verify their annotations.

DialogueView also allows users to aurally verify their speech repair annotations. *WordView* has two playback buttons, *Play System Cleanly* and *Play User Cleanly*, that play the selected region of speech but with the annotated reparanda and editing terms skipped over. We have found this useful in deciding whether a speech repair is correctly annotated. If one has annotated the repair correctly, the edited speech will sound fairly natural. In addition to aurally separating each utterance, the *Play Order* button in *UtteranceView* also skips over reparanda and editing terms.

There are a few cases in which one might want to see the speech repairs. First, speech repairs sometimes indicate lack of certainty on the part of the speaker, and one might want to see this conveyed in *UtteranceView*. Second, the alteration of a speech repair might have an anaphoric reference to an object introduced in the reparandum. For instance, in utterance **u69** in Figure 3.5, the system said, “so engine E one wasn’t maybe it wasn’t the best thing.” When this is cleaned up in Figure 3.10, it becomes “maybe it wasn’t the best thing.” The referent of “it” is no longer shown, which could mislead the user. Although this rarely happens, when it does, a user can still see exactly what happened by looking at *WordView*.

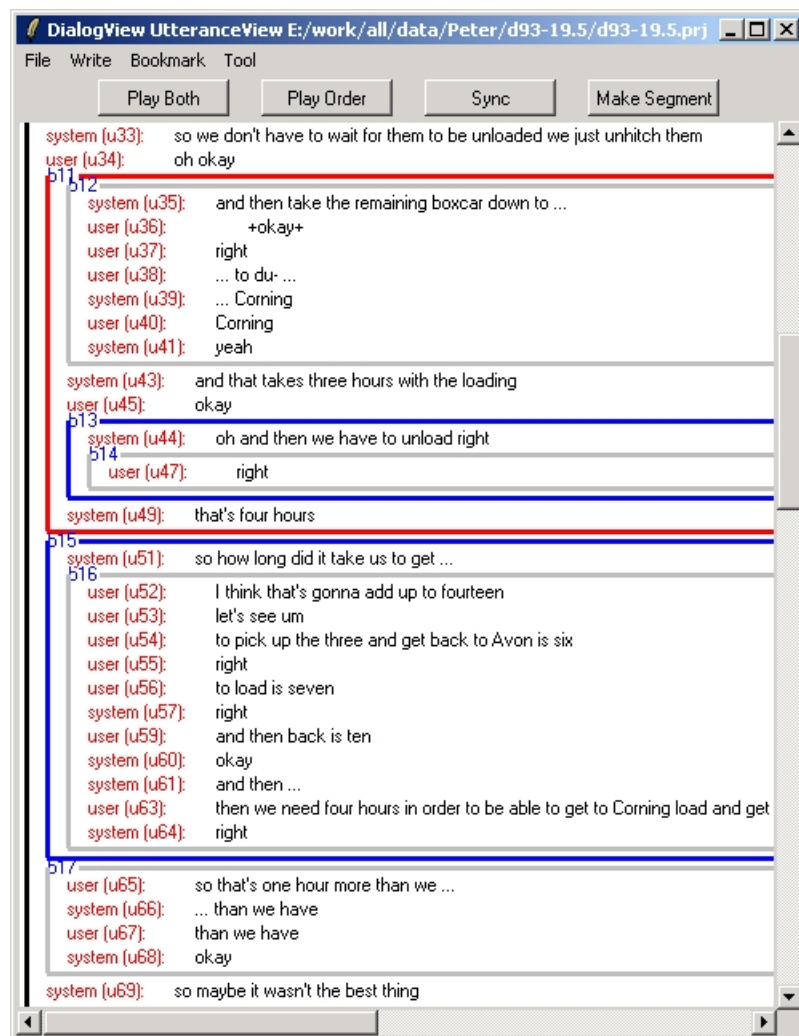


Figure 3.10: Effect of overlap, abandoned, incomplete, and completion tags

3.4.4 Closing Blocks

Even with removing abandoned utterances and cleaning up speech repairs, users can still be overwhelmed by the amount of detail in UtteranceView. This makes it difficult to understand the higher-level structure of the dialogue, which is needed for annotating discourse blocks. Consider utterances **u35** to **u41** in Figure 3.10 where the two conversants are collaborating in building an utterance. In our annotation scheme, we call this a *grounding* block (cf. Clark and Shaeffer 1989; Traum and Nakatani 1999). After being annotated as a grounding block, it would be convenient if the block was replaced by a single utterance that captures the essence of what was said in the block: “and then take the remaining boxcar down to Corning,” as if spoken by the system. Utterances **u65** to **u68** also comprise a grounding block and can be replaced by “so that’s one hour more than we have,” as if spoken by the user. Now consider utterances **u44** and **u47**, which form a dialogue game. Here, the system asked, “oh and then we have to unload, right?” and the user responded with, “right.” Here again, it would be convenient to replace this with a single utterance: “oh and then we have to unload,” as if spoken by the system. Figure 3.11 shows the dialogue excerpt from Figure 3.10 but with the three example blocks *closed* (shown as blocks **b12**, **b13**, and **b17**).

As can be seen, closing the three annotated blocks greatly reduces the amount of detail that users see, which should help them find higher-level discourse blocks. Thus, for doing discourse segmentation, users can start by finding smaller discourse blocks and then work their way up to higher-level ones. Figure 3.12 shows the dialogue excerpt with blocks **b11**, **b15** and **b17** closed, which reduces the excerpt to only 6 lines. Note that for block **b11**, we took some liberties in composing the summary. Fewer lines and less text make it easier for users to keep track of what is going on in the dialogue.

To support the capability of closing blocks, each block is associated with a *summary*, which users need to fill in. When a block is closed, it is replaced by its summary, which is displayed as if it were said by the conversant who initiated the block. To re-open the block, the user can simply double-click on it.

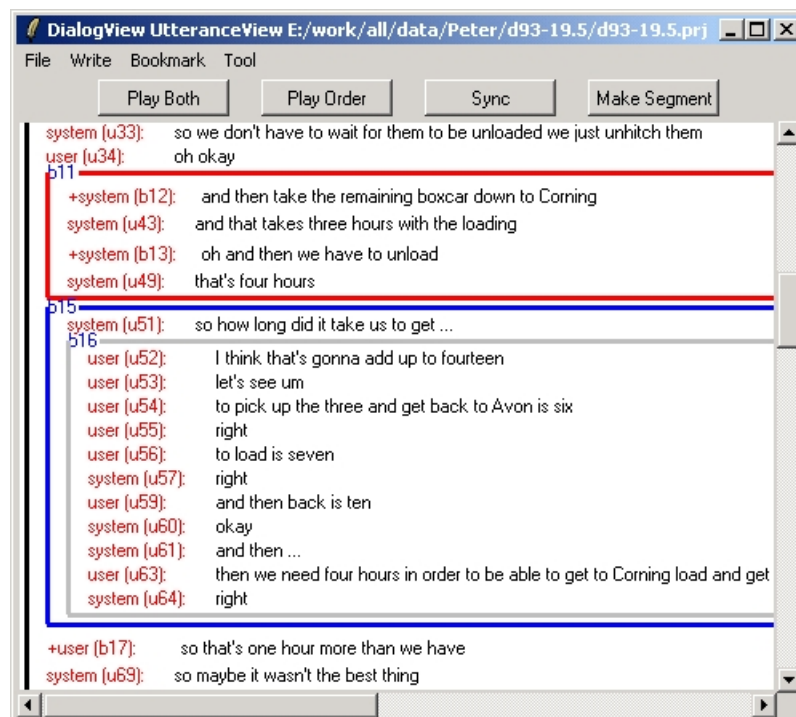


Figure 3.11: Dialogue excerpt with three blocks closed

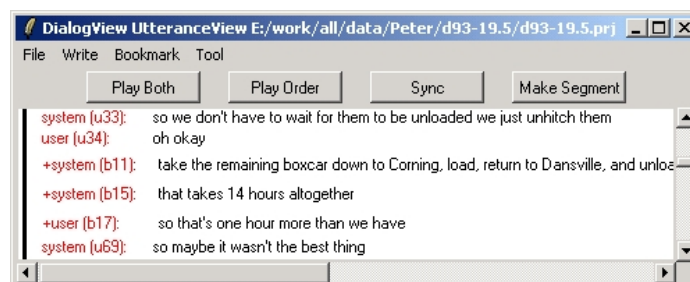


Figure 3.12: Dialogue excerpt with all of the blocks closed



Figure 3.13: BlockView

3.4.5 BlockView

In addition to WordView and UtteranceView, we are experimenting with a third view, which we call *BlockView*. Figure 3.13 gives an example of this view. The bottom half of the figure corresponds to the dialogue excerpt that we have been using in our examples. As can be seen, this view shows the hierarchical structure of the discourse by displaying the summary for each block, indented appropriately. BlockView gives a very concise view of the dialogue. It is also convenient for navigating in the dialogue. By highlighting a line and then pressing *Sync*, the user can see the corresponding part of the dialogue in UtteranceView and WordView.

3.5 User Studies on Abstraction of Dialogue

We have run some user studies on evaluating the usefulness of abstraction.

3.5.1 Evaluation on Play Cleanly

In the WordView, the *Play Cleanly* buttons play back the audio with speech repairs removed. We hypothesized that such functionality should help users annotate speech repairs. We conducted a human-subject experiment to investigate the usefulness those buttons (Yang, Heeman, and Strayer, 2003). Subjects were divided into two groups: the experimental group had access to the *Play Cleanly* buttons and the control group did not. After a short training period, both groups annotated repairs using WordView for six dialogue excerpts. The results showed that the experimental group did better in identifying the scope of repairs, giving a relative improvement of 29% in accuracy, although the performance in detecting repairs was not improved.

3.5.2 Evaluation on Clean Display

Sections 3.4.1, 3.4.2 and 3.4.3 introduced advanced features of DialogueView that affect how a dialogue is displayed in UtteranceView by abstracting away lower-level details, such as speech repairs and abandoned speech. To test the usefulness of these abstractions, we conducted a user study (Yang, Heeman, and Strayer, 2004).

Two dialogues from the Trains corpus, d93-19.5 (Sample1) and d93-17.2 (Sample2), were chosen as the experimental materials. We annotated the speech repairs, overlaps, utterance boundaries, and abandoned utterances for both dialogues. The frequencies of these phenomena are given in Table 3.1. Another person, who has a degree in English as a Second Language Education and was not previously involved with this project, was hired to construct eight comprehension questions for each dialogue. This person was given access to the dialogue via DialogueView, but in which nothing was abstracted out in UtteranceView (the full display style). He was also given the Trains maps so that he had complete and accurate domain information.

Subjects were asked to answer the comprehension questions by reading the dialogue transcripts in UtteranceView using different display styles: the full style and the abstracted style. Ten subjects, all of them graduate students, were randomly divided into two groups. Both groups worked on Sample1 first, and then Sample2. Subjects in Group A used the

Table 3.1: Statistics for the two dialogues used in testing

	Sample1	Sample2
	d93-19.5	d93-17.2
Duration (sec)	260	252
Total utterances	160	138
Total words	825	670
Speech repairs	73	37
Reparandum & editing term words	159	87
Abandoned utterances	18	10
Words in abandoned utterances	38	40

abstracted style for the first dialogue and the full style for the second, while subjects in Group B used the two styles in the opposite order. We set up the experiment this way to control for training effect, inter-personal differences, and dialogue differences. For each subject, we measured the *accuracy gap* and *duration gap* between the two dialogues:

$$\text{accuracy gap} = \text{correct answers in Sample1} - \text{correct answers in Sample2}$$

$$\text{duration gap} = \text{time to finish Sample1} - \text{time to finish Sample2}$$

The results for accuracy are given in Figure 3.14. One can see that all the subjects had more difficulty with the questions for Sample2 than for Sample1 (dialogue differences). One can also see that subjects 3 and 5 had more difficulty with the questions than the other subjects (inter-personal differences). By looking at the accuracy gap, these differences plus the training effect are removed. On average, Group A had an accuracy gap of 1.2, while Group B had an accuracy gap of 1.4. However, the difference between the two groups is not statistically significant (Wilcoxon ranksum, two-tail, $p = 0.59$), suggesting that accuracy is not affected by abstraction.

The results for duration are given in Figure 3.15. Here, Group A had an average duration gap of 5 seconds with a standard deviation of 126; while Group B had an average duration gap of 181 seconds with a standard deviation of 62. The difference between the two groups is statistically significant ($t(6) = 2.80$, $p = 0.03$). Thus the abstracted style should allow annotators to more quickly understand a dialogue, enabling them to more quickly annotate aspects that require complete comprehension of a dialogue, such as utterance tags and discourse structure.

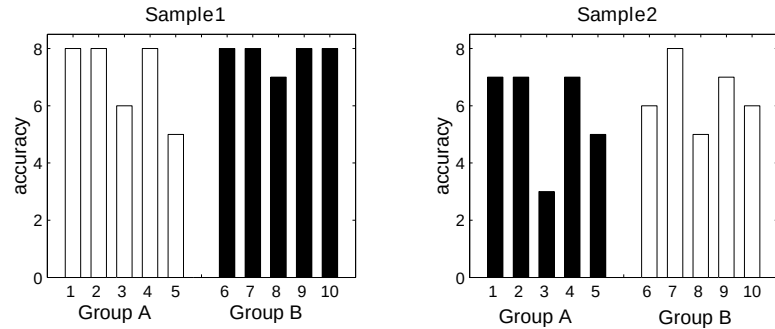


Figure 3.14: Accuracy results: black is for the full style and white for abstracted

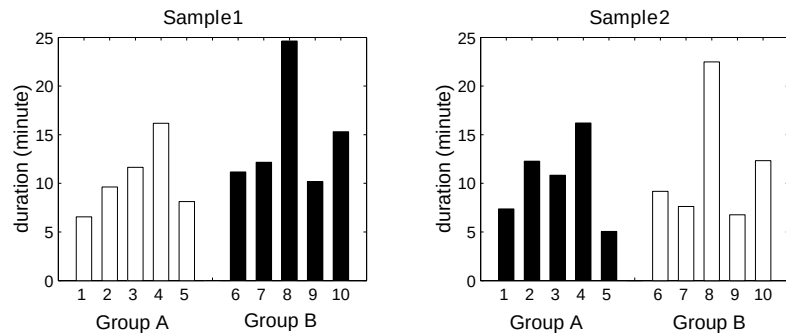


Figure 3.15: Duration results: black is for the full style and white for abstracted

3.6 Comparing Annotations

It is sometimes useful for several annotators to compare their annotations of a dialogue. First, it is useful for consensus annotation, where differences in annotations are compared and resolved in order to reduce the number of coding errors. Second, it is useful for training novice annotators, where novices compare their annotations against an expert's. Third, it is useful for refining an annotation scheme, where several experts compare their annotations to determine if there are ambiguities in their scheme.

We have built an annotation comparison tool (*ACT*) that displays the annotations from different users so that users can visually compare the differences in annotations.

Figure 3.16 shows the interface for *ACT*. The main view for comparing annotations is a variation of *UtteranceView*. Each user's annotation is shown with blank lines inserted to keep identical utterances from different annotations aligned. A single scrollbar on the right controls views of all annotations simultaneously. Although we just show two users'

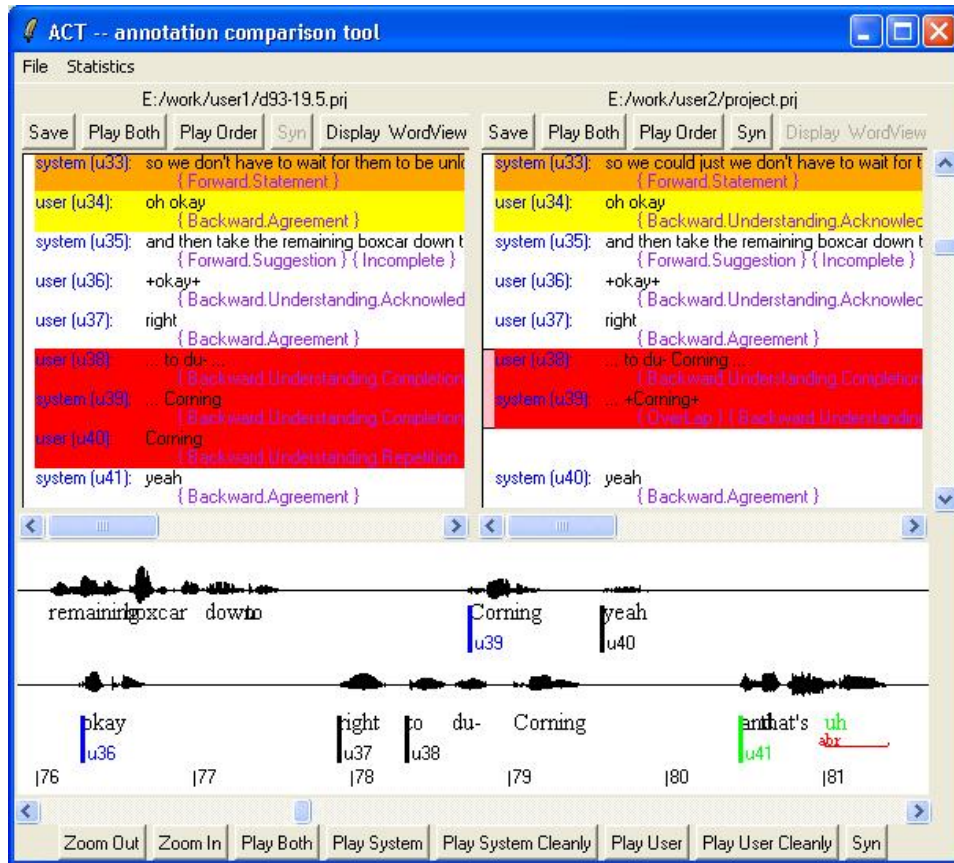


Figure 3.16: Interface of ACT

annotations here, there is no limit to how many can be compared at the same time. ACT also includes WordView. Due to limited screen space, only one user's WordView is shown at a time. The button *Display WordView* in each UtteranceView sets whose WordView annotation is shown.

3.6.1 Comparing Speech Repairs

ACT has several different modes. In one mode, it only highlights differences in the speech repair annotations. Utterances with different speech repairs will be highlighted in orange in each UtteranceView. This is shown in Figure 3.16 for utterance **u33**. To see whether the difference is in the type of the repair or in the extent of the repair, the users will need to look at their exact annotations in WordView.

3.6.2 Comparing Utterance Segmentations

Users can also have ACT highlight differences in utterance segmentation. Differences are shown in red in each UtteranceView. Rather than focusing on individual utterances, our strategy is to highlight *regions* of utterances that may have been affected by a single disagreement in the annotations. To illustrate, consider the two utterance segmentations in Figure 3.16, which correspond to the dialogue excerpt in Figure 3.1. For simplicity, we refer to utterances in the first user’s annotation with a superscript ‘A’, and the second user’s with a ‘B’. Clearly, utterance **u38^A** is different than **u38^B** because it contains different words. Utterance **u40^A** has no corresponding utterance in the second user’s annotation, so it is clearly different. We also view utterances **u39^A** and **u39^B** as being different. This is because the preceding context of the utterances are different. The preceding context of **u39^A** does not include “Corning”, while the preceding context of **u39^B** does. ACT marks utterances as different if they either do not have the same content or their preceding context is different. Hence, the regions of utterances **u38^A** to **u40^A** and **u38^B** to **u39^B** are marked as different. Our definition of what constitutes different utterances results in fewer regions being marked than when utterances are marked individually.

3.6.3 Comparing Utterance Tags

ACT can also highlight differences in utterance tags. Utterance **u34** is highlighted in yellow as a tag difference in Figure 3.16; the first user annotated it as an agreement and the second as an acknowledgement. Note that differences in whether utterances are annotated as overlap, incomplete, or abandoned also count as utterance tag differences.

When comparing utterance tags, users might want to focus on major distinctions and ignore minor ones. Users specify which distinctions should be ignored in the configuration file. This is especially useful when developing an annotation scheme, as it allows users to focus on the glaring differences and temporarily ignore the more subtly differing aspects. For example, the difference between acknowledgement and agreement can be ignored by specifying the following rule:

$$\langle \text{Backward.Understanding.Acknowledgment} \rangle == \langle \text{Backward.Agreement} \rangle$$

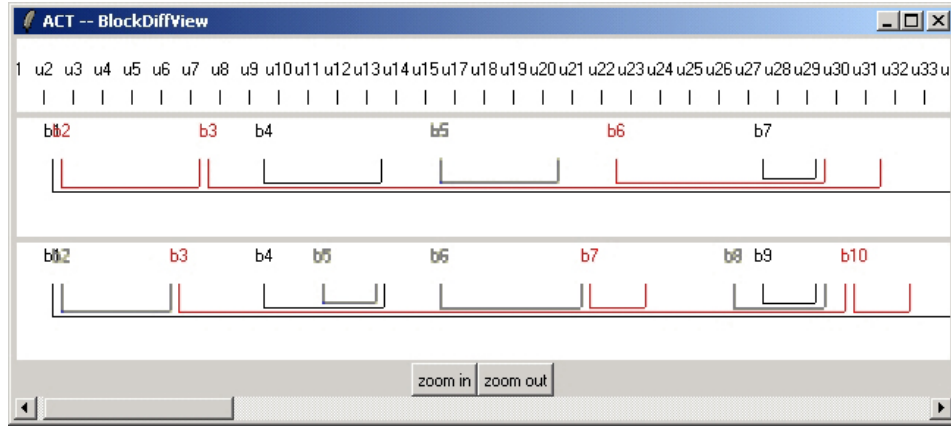


Figure 3.17: Block comparison in the BlockDiffView

More formally, suppose that an utterance has tags A from one user and tags B from another user. The tags are treated as being identical if the following holds:

$\forall a \in A, \exists b \in B$ such that a is the same as b , or there is a rule $a = b$ or $b = a$

$\forall b \in B, \exists a \in A$ such that a is the same as b , or there is a rule $a = b$ or $b = a$

As a second example, users might want to only highlight differences where they disagreed on whether an utterance had a forward or backward function. This can be specified by the following two rules, where ‘*’ can match any string:

$\langle \text{Forward.*} \rangle == \langle \text{Forward.*} \rangle$

$\langle \text{Backward.*} \rangle == \langle \text{Backward.*} \rangle$

3.6.4 Comparing Blocks

Even though UtteranceView for each user can display discourse blocks, we have found it useful to include a special view for comparing discourse segmentation, which we call *BlockDiffView* and show in Figure 3.17. Before comparing block segmentations, ACT requires that there be no differences in the utterance segmentation (and linearization), but differences in speech repairs and utterance tags are allowed. BlockDiffView shows the utterance numbers along the top, with the discourse blocks of each user displayed underneath.

To help users compare their block annotations, we categorize each block B as one of

three types:

Same: There is a corresponding block in each of the other users’ annotations that has the same start and end utterances. Examples in Figure 3.17 are **b4** and **b7** in the top user’s annotation, which are the *same* as **b4** and **b9** in the bottom user’s annotation respectively. These blocks are displayed in black.

Inconsistent: There is a block B' in another user’s annotation that overlaps with B but B' does not embed nor is embedded in B . Examples in Figure 3.17 are **b2**, **b3**, and **b6** in the top user’s annotation, and **b3**, **b7**, and **b10** in the bottom user’s annotation. This is similar to the crossing brackets metric for parser evaluations (Harrison et al., 1991). An inconsistent block is displayed in red.

Consistent: All other blocks. Examples in Figure 3.17 are **b5** in the top user’s annotation, and **b2**, **b5**, and **b8** in the bottom user’s annotation. A consistent block is displayed in yellow.

Ideally, blocks from different users’ annotations would be the same. However, when a block is not the same, but it is consistent, this could just mean that the users are annotating at different levels of granularity. On the other hand, when a block is inconsistent, it could mean that the users have very different interpretations of what is going on in the dialogue, which might be a much more serious concern. Distinguishing inconsistent blocks from consistent ones helps users focus on more important differences.

As one of the uses of ACT is for consensus annotation, we allow users to change their speech repairs, utterance segmentation, and utterance tags in ACT, similarly as can be done in DialogueView. Currently ACT does not support block modifications.

3.7 Extensibility and Implementation

After users have annotated a dialogue, they might want to analyze the annotations. DialogueView does not include any analysis capability; however, similar to a number of other annotation tools (McKelvie et al., 2001; Bernsen, Dybkær, and Kolodnytsky, 2003), it

does use XML for storing the annotations. Any analysis tool capable of reading in XML files can thus be used.

DialogueView can be customized for a specific project through a configuration file; however, the amount of customization is limited. Hence, we also allow users to write their own *plug-in* modules in Tcl/Tk that can be run from inside of DialogueView. For example, for one of our projects, we wrote a plug-in module to search for utterances that were annotated with a specific set of tags and that were in a specific context. Of course, this option requires users have good programming skills and an understanding of the internal structure of DialogueView.

DialogueView is written in Incr Tcl/Tk. Incr Tcl/Tk allows object-oriented programming, which helped us better manage the growing complexity of DialogueView. This also enabled us to reuse pieces of the codes in building ACT. A couple classes of ACT are derived from those of DialogueView, which not only saved us a lot of effort but also helped to guarantee consistency between the two tools. We use the SNACK package for audio support (Sjölander and Beskow, 2000); hence DialogueView supports audio file formats of WAV, MP3, AU, and others.³ DialogueView has been tested on Microsoft Windows (2000 and XP) and Redhat Enterprise Linux.

3.8 Chapter Summary

In this chapter, we described DialogueView, an annotation tool for segmenting dialogue into utterances, annotating speech repairs, coding utterance tags, and creating hierarchical discourse blocks. The tool presents the dialogue to users with multiple views. Each view has a corresponding abstraction, allowing users to see what is going on in details and to see the higher-level structure. The abstractions not only abstract away from the exact timing of the words, but also can skip some of the words, whole utterances, and even simplify discourse blocks to a one-line summary. Depending on the annotation task, users can choose different abstractions to display in a view. These features help users better understand and annotate dialogue.

³see <http://www.speech.kth.se/snack/> for the complete list.

In addition to describing an annotation tool, we have also advocated the use of abstraction in an annotation tool, supported by multiple views. Abstraction emphasizes certain information and hides details that are not directly relevant, which helps users focus on their annotation task. Moreover, the annotations themselves are used in formatting the views, and hence serve the additional purpose of letting users double-check their work; if the abstraction does not make sense, there might be problems with the annotations. Ideally, these design philosophies would be built into a toolkit so that they could be easily used for any annotation task with any type of human interaction.

We used DialogueView to annotate the two corpora, the Trains corpus and the MTD corpus, for this thesis research. We have found DialogueView to be reliable and easy to use. DialogueView is freely available for research and educational use. Users should first install a standard distribution of Tcl/Tk, such as ActiveTcl (<http://www.tcl.tk>), and then download DialogueView from <http://www.cslu.ogi.edu/DialogueView>. The distribution also includes some examples of annotated dialogues.

Chapter 4

Corpora and Annotations

In this chapter, we describe the two corpora that we examined in this thesis research, the Trains corpus collected at the University of Rochester (Heeman and Allen, 1995b), and the MTD corpus collected at Oregon Health & Science University (Heeman et al., 2005).

4.1 The Trains Corpus

The Trains corpus is a collection of human-human task-oriented dialogues, in which two conversants work together to formulate a plan involving the manufacturing and transportation of goods (Heeman and Allen, 1995b). One conversant plays the role of the system, who has complete domain knowledge, such as the distances between cities, how long it takes to manufacture goods, and constraints on how much cargo a train can pull; and the other conversant plays the role of the user, who has a goal in mind to achieve, such as shipping a tank of orange juice to Avon before midnight. The system was told to let the user try solving the problem.

We used the Trains utterance segmentations with minor modifications when necessary. We have also used the dialogue act annotations from Strayer et al. (2003), in which each utterance was tagged with a simplified version of the DAMSL scheme (Core and Allen, 1997). Utterances were tagged as forward or backward functions, non-contributions, or others. Forward functions included statements, questions, checks and suggestions. Backward functions included agreements, answers, acknowledgements, and repetitions. Non-contributions included abandoned and ignored utterances. The flow of conversation would not change if non-contributions were removed. Others include stalls such as “um”

and “let’s see”, and move-ons such as “okay” and “alright” with high prosody and high energy that signal the start of a new (sub-)topic.

Strayer et al. (2003) also annotated hierarchical discourse structure to the Trains dialogues, following Grosz and Sidner’s (1986) discourse structure theory. Special effort was devoted to examining whether there was a shared intention when annotating discourse segments, and discourse markers were also used to help disambiguation. Two types of discourse segments were distinguished: transaction segments and dialogue games (Carletta et al., 1997). Transaction segments typically corresponded to sub-tasks, such as stating the goal of the plan, constructing the steps in the plan, refining the plan, summarizing the plan and evaluating the plan. Dialogue games were related to adjacency pairs, which typically consisted of a first part and a second part. Each part could have one or more utterances. The second part was explicitly marked as a special type of segment. For question-answer pairs, the first part was the question, and the second part was the answer. Note that if the second part contained more than one non-trivial utterance, these utterances were annotated as a forward function instead of an answer, as the backward functions of these utterances had been captured by the second part.

Twelve Trains dialogues totalling about 50 minutes were annotated: d92a-4.4 (4’51”), d92a-5.2(13’09”), d93-17.2 (4’12”), d93-19.2 (2’42”), d93-19.5 (4’21”), d93-21.2 (4’28”), d93-22.2 (7’58”), d93-23.1 (3’01”), d93-25.3 (1’16”), d93-25.4 (1’43”), d93-12.2 (1’27”) and d93-20.3 (1’56”) (Heeman, Yang, and Strayer, 2003). In all annotations, two or more expert annotators annotated each dialogue for utterance tags and discourse structure. They then compared their annotations to reach a consensus using the tool ACT. Most of the differences between the annotators were resolved successfully.

4.2 The MTD Corpus

The MTD corpus is a collection of human-human multi-tasking dialogues (Heeman et al., 2005). We collected this corpus to better understand how people manage multiple tasks in their conversations.

4.2.1 Tasks

In the MTD corpus, conversants finish two types of tasks via conversation: an on-going task that takes a long time to finish and a momentary task that just takes several turns to complete. The momentary task is kept quite simple as we are not studying interruptions to it. Hence, this task does not need to build up much context and can be single initiative. We also want to vary the urgency of this task. The ongoing task needs to build up significant context that conversants will have to keep in mind. After an interruption, this context will be needed to finish the task, and so might need to be re-established. The tasks should also encourage both participants to equally participate as we feel this will be representative of tasks that spoken dialogue systems will be used to accomplish in the future.

In the ongoing task, a pair of players collaborate with each other to assemble as many poker hands as possible, where a poker hand consists of a full house, flush, straight, or four of a kind. Each player has three cards in hand, which the other cannot see (players are separated so that they cannot see each other.) Players take turns drawing an extra card and then discarding one, until they find a poker hand, for which the players earn 50 points; they then start over to form another poker hand. To discourage players from simply rifling through the cards to look for a specific card without talking, one point is deducted for each picked-up card, and ten points for a missed poker hand or incorrect poker hand. To complete this game, players have to converse to share card information, and explore and establish strategies based on the combined cards in their hands.

The poker game is played on computers. The game display, which each player sees, is shown in Figure 4.1. The player with four cards can click on a card to discard it. The card disappears from the screen, and a new card is automatically dealt to the other player. The player with four cards clicks the “Done Poker Hand” button to start a new game once they find a poker hand.

The momentary task is a picture game. From time to time, the computer generates a prompt for one player to find out whether the other player has a certain picture on the bottom of the display. The picture game has a time constraint of 10 seconds, 25 seconds

or 40 seconds, which is (pseudo) randomly determined. The players get five points for the picture game if the correct answer is given in time. The overall goal of the players is to earn as many points as possible from the two games.

To alert the player to the picture game, two solid bars flash above and below the player's cards. Thus the player will know that there is a pending picture game without having to take attention away from the poker game. The color of the flashing bars depends on how much time is remaining: green for 26-40 seconds, yellow for 11-25 seconds and red for 0-10 seconds. The Player can also see the exact amount of time in the heading for the picture game. In Figure 4.1, the player has to find out whether the other has a blue circle, with 6 seconds left.

4.2.2 Corpus Collection

To collect the conversations, we created a tool consisting of three processes, each running on a different computer. A central process deals the cards, decides the timing of the momentary tasks, records both channels of speech and creates a log file. The two other processes run the display shown in Figure 4.1 for each participant. The three processes communicate by sending messages over TCP/IP sockets.

Sessions lasted about one hour. After a short orientation, the participants played the poker game for about five minutes to become familiar with the rules. They then did a practice session with the poker game and the momentary task for about 15 minutes, so that they could get used to managing both tasks. Finally, they did two 15 minute sessions, which are the data that we analyzed in this thesis work. During each session, nine picture games (three for each urgency level) were prompted for each player.

A post-experiment survey was then conducted to ask whether the participants had played poker before, whether they switched the momentary tasks as soon as they arose, and how they enjoyed the games. All participants reported that they always noticed the bars immediately when they started to flash, that they never ignored a momentary task on purpose, that they waited for a good timing to switch to the momentary tasks, that task switching made the poker game more difficult, and that they enjoyed the games. All players had a bachelor degree or higher, and were familiar with computers.

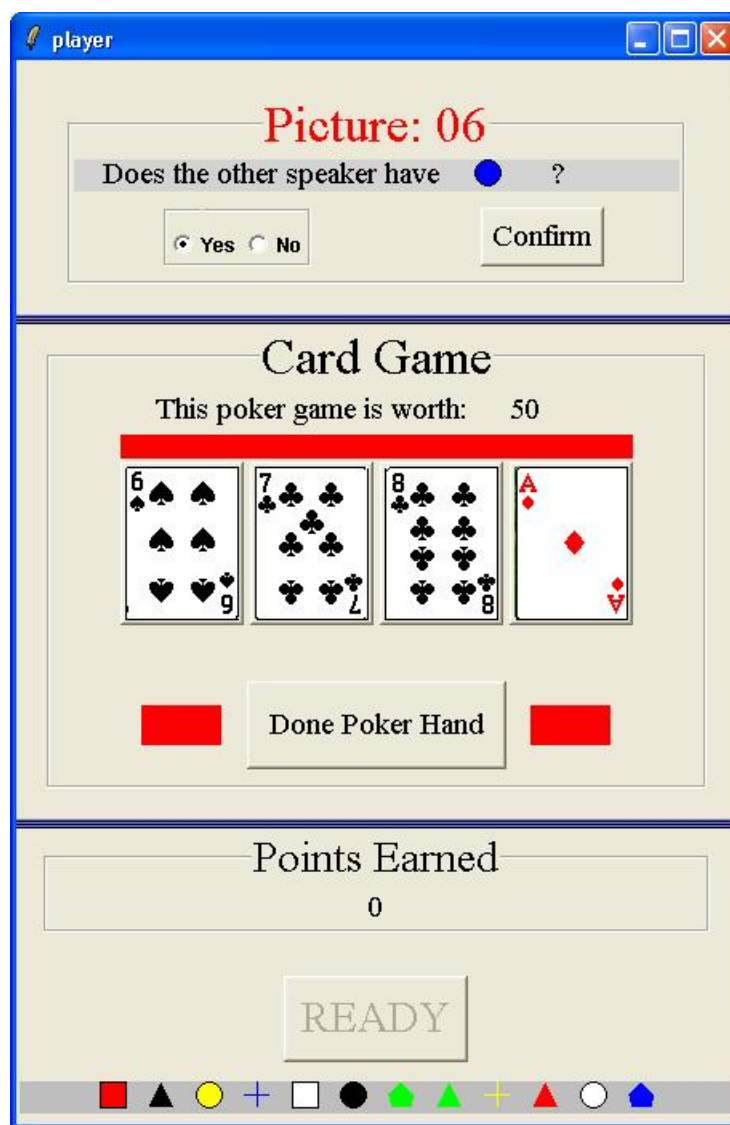


Figure 4.1: The game display for players

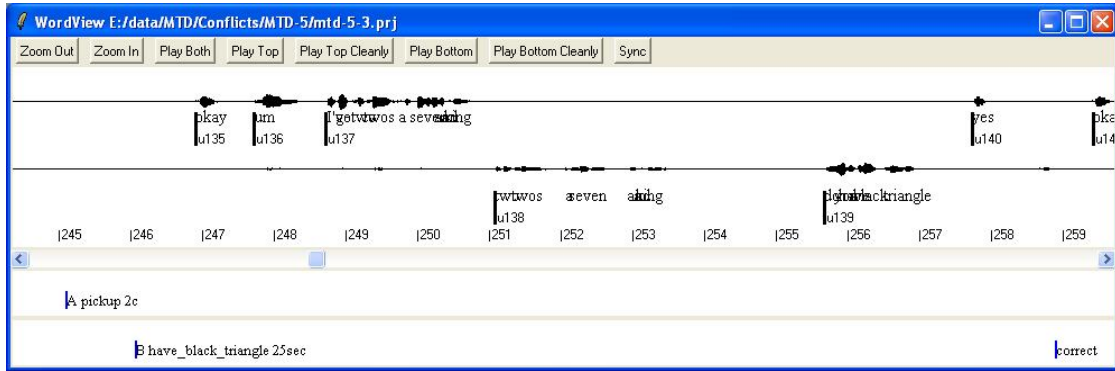


Figure 4.2: Events displayed in the WordView of DialogueView

4.2.3 Transcriptions and Annotations

Several tools were used to annotate the data. First, the tool SpeechView (Sutton et al., 1998) was used to transcribe the words that each player said. We then ran forced-alignment using the CSLU Toolkit (Sutton et al., 1998). DialogueView was used to annotate utterance boundaries, utterance tags, and hierarchical discourse structure. The four principles in Nakajima and Allen (1993) were taken into account when segmenting a dialogue into utterances. The first principle is *grammatical principle*: an utterance boundary is inserted at the completion of a grammatical clause. The second is *pragmatic principle*: each utterance unit has only one speech act. The third is *conversational principle*: an utterance ends when the speakers switch. The fourth is *prosodic principle*: an utterance boundary is inserted when there is a long pause or end-of-utterance tone.

From the log file of the collection tool, two additional text files were created for each conversation. The files contain all the events of the computer dealer and the GUI actions of the two players for each task with time-stamps. For the momentary task, it contains each question, the time it was generated, the answer, and the time it was answered; for the poker game, it contains information of when a card was dealt or discarded, and information of when a poker hand was achieved or missed. DialogueView displays this information time-aligned with the word and utterance annotations, as shown in Figure 4.2. In this example, the top player, A, picked up a two of clubs at about 245s (displayed as “A picks up 2c”); B received a momentary task to ask whether A has a black triangle at 246s

with a time constraint of 25 seconds (displayed as “B have_black_triangle 25sec”), and A correctly answered it at 259s (displayed as “correct”). This setup gives us a full view of what was happening.

Six MTD dialogues by three pairs of players, totaling about 90 minutes of conversation, were annotated with dialogue acts, using the same simplified DAMSL scheme on the Trains dialogues by Strayer et al. (2003). Utterances were annotated as forward functions, backward functions, non-contributions, or others.

The ongoing task can naturally be divided into individual poker games, in which the players successfully completed a poker hand. Each poker game can be further divided into a sequence of card segments, in which players discussed which card to discard, or a poker hand was found. As well, utterances involved in each picture game were grouped into a segment. Figure 4.3 shows an excerpt from an MTD dialogue with these annotations in the UtteranceView of DialogueView. Here b7 is a game segment in which players got a poker hand of flush (five diamonds); and b8, b10, b11, b12 and b14, inside of b8, are card segments. Also embedded in b7 are b9 and b13, each of which is an interruption segment of the picture game. Ten MTD dialogues by five pairs of players, totaling about 150 minutes of speech were annotated with these segments.

No claim is made that these segments, especially the card segments, follow Grosz and Sidner’s (1986) definition of discourse segments. To achieve a poker hand, players communicate their cards, explore different poker hands, agree on one to go for, and decide which card to discard based on this agreed strategy (Toh, Yang, and Heeman, 2006), which probably constitute the discourse goals. Each time a new card is dealt, players would re-examine whether the established strategy is still optimal, and abandon it if a more viable one emerges. Although there is a domain goal to examine the completion of a poker hand and discard a card if no poker hands found in each card segment, it is unclear whether the initiator sets up a corresponding discourse goal at the beginning of a card segment.

Most of the segments could be automatically derived from the log file. For example, the time a new hand dealt is usually the start of a new game segment; the time a new card dealt is usually the start of a new card segment. We then manually fixed some mistakes.

For example, a mis-generated segment was removed where a player simply discarded a card without any discussion; a segment boundary was moved if an utterance, typically an acknowledgement, was carried over to the next segment (i.e., the utterance was uttered after a new card was dealt, but it was about the card being discarded).



Figure 4.3: An excerpt of an MTD dialogue

Chapter 5

Managing Complex User Goals

5.1 Introduction

As discussed in Section 2.2, the theory of discourse structure proposed that conversants decompose a dialogue goal into sub-goals, and then work together to achieve each sub-goal (Grosz and Sidner, 1986). Each discourse segment corresponds to a sub-goal, which is brought up by the initiator and then recognized by the non-initiator. The theory, however, did not comment on the initiative behavior of the initiator and non-initiator, i.e. the types of utterances that the initiator of the segment or the non-initiator can make. Hence, it under-specified what conversants can do in dialogue.

The previous research we reviewed in Section 2.3.1 has suggested the possible correlation between dialogue goal and control. Whittaker and Stenton (1988) found that the conversant who initiates a topic shows initiative in more turns in discussing the topic. Cohen et al. (1998) proposed that control belongs to the initiator of a process, which is also goal-oriented. However, none of these research works gave a clear account of the relation between dialogue goal and control.

In this chapter we examine the different initiative behavior by the initiator and non-initiator of a discourse segment in details. Following Strayer et al. (2003), we hypothesize the *restricted initiative model*: control belongs to the initiator of a discourse segment. We show evidence of this model by analyzing the Trains dialogues. We also use computer simulation to show that following this model results in more efficient conversations.

5.2 Restricted Initiative Model

Figure 5.1 shows a stretch of conversation from the Trains corpus, in which two conversants are making a plan to produce and ship orange juice. The user was controlling (showing initiative) in creating the plan while the system just made acknowledgements. After the plan was created, the system showed initiative by evaluating the plan and telling the user that it would take twelve hours.

user:	then how about if I take an engine along with a boxcar over to Corning
system:	mm-hm
user:	add a tanker on
system:	mm-hm
user:	fill the boxcar with oranges and bring it back to Elmira to have it turned into juice
user:	and drop off the boxcar fill the tanker with orange juice and deliver it to Avon
system:	that will take twelve hours
user:	ok

Figure 5.1: An example for initiative and control

Such dialogue phenomena happened a lot in the Trains corpus. The conversants collaborate in each joint-goal, with one taking the lead in achieving it. According to Grosz and Sidner (1986), the joint-goal in a dialogue, or the purpose of a discourse segment, is the segment initiator's intention to create a shared plan (Lochbaum, 1998). Strayer et al. (2003) hypothesized that control is subordinate to discourse structure: the conversant who initiates a discourse segment is the one who has control during the entirety of the segment; the non-initiator plays a restricted role in achieving the goal of the discourse segment. This model was referred to as *restricted initiative*. According to this initiative model, control is the authority owned by the segment initiator that the non-initiator has to respect; the initiator and non-initiator demonstrate very different initiative behavior inside a discourse segment.

Strayer et al. have also examined the Trains dialogues to seek evidence for the restricted initiative model (Strayer, Heeman, and Yang, 2003; Heeman, Yang, and Strayer,

2003). They found that the initiator of a discourse segment shows more initiative inside the segment. The non-initiator only makes two types of contributions occasionally: collaborative completions, in which the non-initiator helps the initiator to complete an utterance, and short contributions to the discourse segment purpose. They also found that the non-initiator tends not to consecutively show initiative.

The evidence found by Strayer et al., however, was only preliminary. For example, they did not examine whether the initiator and the non-initiator alternate in showing initiative inside a discourse segment, which is an indication that the initiator might not be controlling the stretch of conversation. Moreover, in the Trains dialogues, the two conversants played two different roles, one as the system and one as the user. It is unclear whether the correlation between the initiator of a discourse segment and the person who shows more initiative is true for both roles.

5.3 Evidence from Human-Human Dialogue

In this section, we present more evidence to support the restricted initiative model from analysis of the Trains corpus. We first examined the frequency of the initiator and non-initiator showing initiative in a discourse segment. We then examined how often the initiator and non-initiator consecutively show initiative. We finally examined the initiative pattern of a discourse segment.¹

In the Trains dialogues, the two conversants played very different roles. The system had domain knowledge, and the user had a goal to solve. The user, most of the time, was responsible for creating a plan; and the system was evaluating the plan. We thus reported our results with respect to the system and the user separately, to avoid the effect that might be introduced by the different domain roles.

5.3.1 Frequency of Initiative

We first examined the frequency of showing initiative inside a discourse segment. Following Whittaker and Stenton (1988), we reason about initiative based on utterance tags.

¹Content of this section was previously published in Yang et al. (2004).

Table 5.1: Frequency of initiative

	Initiator:	
	System	User
Initiator shows initiative in first utterance	66	145
in other utterances	108	302
Non-initiator shows initiative	8	43

Utterances that are annotated as forward functions, such as statements, proposals, or questions, show initiative; utterances annotated as backward functions or other do not.

Of the 233 transaction segments and dialogue games in our annotated dialogues, 69 are initiated by the system and 164 by the user. Table 5.1 reports the number of utterances made by the initiator and non-initiator that show initiative. Column 2 gives the results where the system is the initiator, and column 3 gives the results where the user is the initiator.

We separated the first utterance in a segment, as by definition, it is always made by the initiator. In discourse segments initiated by the system, the initiator (system) shows initiative 13.5 ($108/8$) times more often than the non-initiator (user). In discourse segments initiated by the user, the initiator (user) shows initiative 7 ($302/43$) times more often than the non-initiator (system). Even though the roles of the system and user are so different, we see that both behave similarly when acting as the initiator of a discourse segment, i.e. showing overwhelming initiative inside the segment.

5.3.2 After Non-initiator Shows Initiative

We next looked at what happens after the non-initiator shows initiative. Does the non-initiator keep showing more initiative? To answer this, we examined what happens after the initiator makes an utterance that shows initiative versus what happens after the non-initiator does. For every utterance by the initiator or non-initiator that shows initiative, we examined who shows initiative next, either by making an utterance that shows initiative, or by initiating an embedded discourse segment. An embedded discourse segment is treated as showing initiative by the initiator except for second parts, which serve backward

Table 5.2: After a speaker shows initiative

	Initiator:	
	System	User
Initiator shows initiative followed by		
End of Segment	30	66
Initiator shows initiative	120	381
Non-initiator shows initiative	13	46
Ratio of Initiator to non-initiator	9.2	8.3
Non-Initiator shows initiative followed by		
End of Segment	8	16
Initiator shows initiative	6	50
Non-initiator shows initiative	3	22
Ratio of Initiator to non-initiator	2.0	2.3

functions. The results are given in Table 5.2. We restricted our analysis to what happens in the current discourse segment. If there is no utterance that shows initiative after the utterance in question and there are no discourse segments embedded in the block, we count it as end of segment, and report it in the first row of Table 5.2. Column 2 is where the system is the initiator and column 3 is where the user is.

First, consider the case where the system is the initiator of a discourse segment. After the initiator shows initiative (by making an utterance, or initiating a subsegment), the next person to show initiative (after that utterance or the subsegment) is the initiator in a ratio of 9.2:1 (120/13). In those cases where the non-initiator shows initiative, a ratio of 2.0:1 (6/3), the initiator is the next to show initiative. These two ratios are very similar to the ratios when the user is the initiator (8.3:1 and 2.3:1 respectively). These results suggest the non-initiator tends not to show initiative continuously, compared with the initiator.

5.3.3 Initiative Patterns

The previous subsection shows that when the non-initiator shows initiative, the next person to show initiative tends to be the initiator. However, it is still unclear whether the initiator is really controlling the entire segment, or how often the initiator has control over the entire segment. We thus examined the patterns of initiative in discourse segments.

Table 5.3: Patterns of initiative

	Initiator:	
	System	User
Conforming		
Only initiator shows initiative	17	25
Others	5	15
Questionable	0	11
Non-conforming	2	8

To determine the pattern of initiative in a discourse segment, we carried out the following steps. First, any embedded segments were viewed as an utterance that showed initiative by the conversant who initiated the subsegment. Second, we removed any utterances that did not show initiative. Third, we counted how many initiative utterances in a row each conversant alternately made. Thus the pattern “3-2-4-1” represents a discourse segment in which the initiator makes three utterances (or subsegments) that shows initiative, followed by two by the non-initiator, followed by four by the initiator and ending with one by the non-initiator. Hence, the initiator shows initiative seven times in the segment while the non-initiator shows initiative three times.

We classified initiative patterns into three types: conforming patterns, non-conforming patterns, and questionable patterns. Conforming patterns are patterns that satisfy the following conditions:

Majority: initiator shows at least 80% of the initiative.

Example: 5-1 Counterexample: 3-1

Isolated: non-initiator never shows initiative more than once in a row.

Example: 4-1-3-1 Counterexample: 4-2-3-1

Non-alternating: no subpattern of 1-1-1 (starting with non-initiator).

Example: 8-1-3-1 Counterexample: 8-1-1-1

A pattern is *Non-conforming* if the initiator shows initiative less than 60% of the time. All other patterns are viewed as questionable.

In our annotated dialogues, we focused on the 83 transaction segments. The results are reported in Table 5.3. For the segments initiated by the system, 22 are conforming

patterns and only 2 are non-conforming; for the user, there are 40 conforming and 8 non-conforming. In total we see that 75% of the segments have a conforming initiative pattern, 12% are non-conforming patterns, and 13% questionable.

Conforming patterns are clear examples where the initiator is controlling the discourse segment. For non-conforming patterns, it is also clear that the initiator is not driving the conversation. In a segment with a questionable pattern, the initiator tends to be controlling (showing most cases of initiative), yet there might be local extent where it is unclear who is the controller. Thus our results show that the initiator is controlling three out of four segments.

More research is needed to understand what is happening in segments with non-conforming and questionable patterns. It is also interesting to note that the majority of these segments are initiated by the user. This might be due to the domain setup that the system has all domain knowledge, but the user is responsible for creating a plan. The system might feel inclined to, and is able to, show initiative inside a user-initiated segment.

5.3.4 Discussion

From our empirical study on the Trains dialogues, we first show that the initiator shows overwhelming initiative inside the discourse segment, despite the domain roles they play. Although we are unclear about how control is mutually agreed upon by both conversants, showing initiative is an indication that a person has obtained the control or still has the control. We also show that, after the non-initiator shows initiative, the initiator tends to pick up initiative next. This suggests that the non-initiator is not controlling the segment. In other words, the control by the non-initiator is only at the utterance level. Finally, by examining the patterns of initiative in discourse segments, we found that the initiator tends to have entire control in three out of four segments. All these results are evidence supporting the restricted initiative model that control is subservient to discourse structure.

Since discourse structure is hierarchical, control is thus hierarchical too. As discourse segment purpose being pushed and popped from the stack of attentional state, control switches according to who initiates the discourse segment. If one starts an embedded discourse segment, one has control over the segment; however, when the segment ends,

the initiator of the outside segment is inclined to take back the control.

A special type of control, *delegated control*, happens in adjacency pairs. An adjacency pair typically consists of a first part and a second part. One of the functions of the second part is to fulfill a social obligation brought about by the first part (Traum and Allen, 1994), but the responder has a choice of how to fulfil it (possibly based on his belief and capability). This phenomenon can be viewed as the person (P1) who has control in the first part explicitly passing the control to the other person (P2), and it is mutually understood that P2 will give back the control to P1 once the obligation is fulfilled. Thus this control is “delegated”. Examples of an adjacency pair and delegated control are given in Figure 5.2. The user stated a condition in utterance u170 and then asked a question in u171. Both of these two utterances showed initiative, thus the user had control. As soon as the question (u171) ended, control was delegated to the system. Looking only inside the second part (u171-u176), the system was in control of the whole segment, but this control was explicitly forced by the user who was the initiator of the adjacency pair, and this control ended when the second part ended.

Segment	Speaker	U#	Utterance	Control
AP	user	u170	what if E two was delayed by an hour	u
	user	u171	would that still make it to Bath	u
SP	system	u172	um E two is the one going to Avon	s (delegated)
	user	u173	right	u (delegated)
	system	u174	right	u (delegated)
	system	u175	that would put it at Bath at one p.m.	s (delegated)
	system	u176	so that's too late	s (delegated)

Figure 5.2: Example of an adjacency pair and delegated control (d92a-5.2)

5.4 Evidence from Computer Simulation

To better apply human strategies in a dialogue system, we not only need to understand what people do in conversation (a descriptive account), but also why people are doing so (an explanatory account). Investigation of human-human dialogues, however, only reveals a descriptive account of human conventions (i.e. what people do in conversation). It is difficult to determine the reasons that underlie human conventions from simply analyzing

human-human dialogues (i.e. why people adopt these conventions). For example, in the previous section, we examine conversants' initiative behavior, but we do not know the underlying reasons for such behavior. Computer simulation might help researchers understand human conventions at a deeper level by justifying the adoption of them (see Section 2.5).²

There are several hypotheses of why people adopt the restricted initiative model:

1. The non-initiator fails to determine the initiator's sub-goal for the segment until the very end of the segment. In this case, because the non-initiator does not know what the initiator is trying to accomplish, the best that the non-initiator can do is to just follow along.
2. Even when the non-initiator knows the initiator's sub-goal, the non-initiator might still want to wait until the initiator finishes a proposal for the whole plan before commenting on it for efficiency reasons, unless there is an urgent event (c.f. the principles of plan quality and information quality (Walker and Whittaker, 1990)).
3. Other factors, such as actor role, social position, and cultural conversation rules related to politeness might also contribute to the restricted initiative. For example, a student might show restricted initiative when talking to a professor, allowing the professor to control the conversation.

The first reason is simply because the non-initiator does not know enough to meaningfully contribute, while the second is a deliberative strategy of holding off in order to let the initiator get his entire point across. In this section, we describe our preliminary experiments in using computer simulation to examine the second hypothesis. Human languages have evolved over generations: efficiency, such as minimizing collaborative effort (Clark and Wilkes-Gibbs, 1986), should account for some conventions adopted in human-human conversations.

²Content of this section was previously published in Yang and Heeman (2007b). A preliminary experiment was also published in Yang and Heeman (2004).

5.4.1 Computer Simulation: Domain

To determine whether efficiency can account for the restricted initiative strategy, we ran computer simulation experiments in which the user and system are modeled as software agents, and both know what the joint-goal is and are able to think of solutions to solve it. The task and the conversation from the Trains corpus are too complex to model directly. Hence, we used a simpler domain for our simulation, but one that is still complex enough to shed light on the initiative model used in Trains. We used a revised version of the furniture layout task (Walker, 1996).

Domain Task

The domain task involves two agents, A and B, carrying out a dialogue to arrange furniture in a room. A furniture item has three attributes: type (e.g. sofa), color (e.g. red), and value (1 to 20). Information about all furniture items is mutually known by both agents.

Each agent also has a set of constraints, initially unknown by the other agent, that it wants to satisfy. Six types of constraints are used, as shown below.

- *Item X must be in the room.*
- *Item X must not be in the room.*
- *If item X is in the room, item Y must also be in the room.*
- *If item X is in the room, item Y must not be in the room.*
- *If item X is not in the room, item Y must be in the room.*
- *If an item of type T is in the room, its color must be C.*

The variables X and Y in the constraints can be either specific items, such as a green table, or can just be the type of furniture, as in ‘a table’. Each constraint has a value between 1 and 20. The task of the two agents is to pick five furniture items for the room that result in the highest possible score where the score is the sum of the values of the five furniture items, less the values of any violated constraints from either agent.

The complexity of the task can be characterized by the following parameters.

I_o : overall number of furniture items

P_A : number of A’s constraints

P_B : number of B’s constraints

Communication Language

Similar to Sidner (1994), we use the following speech act schemas.

- (*propose Speaker Hearer Item*)
- (*inform Speaker Hearer Constraint*)
- (*accept Speaker Hearer*)
- (*reject Speaker Hearer*)
- (*acknowledge Speaker Hearer*)
- (*release_turn Speaker Hearer*)

The first five speech acts allow agents to propose items, and to accept or reject them. Agents can also inform each other of their constraints so that they can reach agreement as to what items to include. Agents also must acknowledge proposals and informs. As defined by Whittaker and Stenton, the speech acts of *propose* and *inform* show initiative while the others do not (Whittaker and Stenton, 1988).

For turn taking, we include the action “release_turn”, which the agent that currently has the turn can perform to signal that it is relinquishing the turn (Traum and Hinkelman, 1992).

Agent Reasoning

The furniture arrangement task involves complex reasoning. Assuming that an agent knows all of the constraints, choosing the best five items is an NP-complete problem. Luckily, a greedy strategy can be employed, whose performance is very close to optimal (Yang and Heeman, 2004). To get a solution with the greedy algorithm, one finds the single item that results in the highest score and commits to it, and then finds the second item such that it and the first item result in the highest score, etc.

The agents use the greedy strategy in their reasoning. They consider all constraints that they know about: their private constraints as well as the ones that have been communicated by the other agent. We also assume that agents never make mistakes and never forget constraints that have been communicated.

Problem Solving Assumptions

There are different strategies that agents might adopt to solve the problem in this domain. For example, agents can communicate their constraints with each other, and then create a best plan. Or agents can create a complete plan, and then evaluate it, and then communicate violated constraints, as in the plan mode of Yang and Heeman (2004). Or agents can decide the items to move in one by one, as in the item mode of Yang and Heeman (2004). Or agents might employ some combination of the above three strategies. In this simulation, we focus on the strategy similar to the plan mode, and thus we assume the following:

- A1: Agents will have two joint-goals: to create a plan and to evaluate a plan. Similar to the conversation in Figure 5.1, agents will first jointly create a complete plan, and then jointly evaluate the plan and communicate constraints. This procedure repeats until both agents agree on a plan.
- A2: Constraints are communicated when violated. Unlike in the artificial Trains domain, people often do not have hard constraints to satisfy. Instead, it is usually a trade-off between difference constraints, which come to people’s mind for comparison when they are violated.
- A3: An agent accepts a plan when the agent has no better alternative or the agent has no private constraints that impact the plan.

5.4.2 Initiative Strategies

In the previous section, we specify a problem-solving strategy, but we do not specify which agent does which action; in other words, we do not specify the initiative strategy. In this section, we contrast an initiative strategy based on the restricted initiative model with two others, in which control does not exist but there is only turn-taking in conversation.

Plan-Based Restricted Initiative (PI)

In the restricted initiative model, one conversant initiates a discourse segment to solve a domain goal, and takes the lead in developing the plan. After the plan is presented, the

agents then take on the goal of critiquing it, again, with one of the agents taking the lead. To implement this in the furniture domain with the plan mode, we have each agent first reason about how they would solve the problem given the constraints it knows. Then, one agent, A, takes the initiative and presents its plan one item at a time, with B just acknowledging after each item. Once the plan is fully presented, A releases initiative. B then takes on the goal of evaluating the plan. B first compares A's plan to the one that B would have proposed. If B does not accept A's plan according to assumption A3 in section 5.4.1, B will show initiative to inform A of any constraints that are unknown to A and that lower the score of A's plan but not B's. Then B rejects A's plan and releases initiative. A will then show initiative to create another plan, taking into account the new constraints informed by B. This process repeats until they reach an agreement on a plan. In this strategy, the agent without initiative can take speaking turns, but the turns simply consist of acknowledgements.

Turn Initiative 1 (TI1)

As an alternative, we developed a strategy in which agents do not keep control over an entire plan. Instead, agents alternate initiative on a turn-by-turn basis. When an agent has the turn to speak, the agent will try to show initiative. Under this strategy, there is no control in conversation to guide an agent when to show initiative.

We present two variations of this. In the first variation both agents first devise a plan on their own, and then alternate in proposing the five items of a furniture plan, and then alternate in critiquing the resulting plan. When proposing an item, the agent picks the first item from its plan that has not yet been proposed.

Turn Initiative 2 (TI2)

The TI1 strategy has the agents reason about the furniture items as if they will be able to present their plan in its entirety, but they only get to propose one item interchangeably. To overcome this mismatch, we present a second alternative in which agents reason about which furniture item to propose immediately before proposing it, taking into consideration the ones already proposed. Thus A first determines the best item, based on the constraints

it knows. Then B determines the second item, given that the item A proposed is in the plan. Then A determines the third item that is the best given the first two. This continues until all five items are chosen, followed by plan evaluation.

5.4.3 Computer Simulation Results

To show that the results hold over a range of task difficulties, we examined the three initiative strategies under different simulation parameter settings. In this section, we use $(I_o, P_A - P_B)$ to denote these different settings, for example, (20,40-40) means that each agent initially has 40 private constraints and they choose five out of 20 furniture items to move into the room. For each parameter setting, we created 100 sets of constraints and items and ran the simulation experiments for all three initiative strategies.

For evaluating the initiative strategies, we measured solution quality and communication effort. Solution quality is the score of the final agreed plan. Communication effort is the number of speech acts of proposing and informing (showing initiative) in a dialogue. We did not count others (accept, reject, acknowledge, release_turn) as such speech acts can be easily accomplished without much effort, such as uttering only one word “okay” or “yeah”.

Solution Quality

Figure 5.3 shows the solution quality of the three initiative strategies under different parameters. Figure 5.3A shows the solution quality as the number of items is varied and the number of constraints of each agent is held at 40. Figure 5.3B shows the solution quality as the number of constraints is varied and the number of items is held at 20. The solution quality of the three strategies under each setting is very similar and the differences are not statistically significant (paired ttest $p > 0.1$). Hence all three initiative strategies result in similar solution quality.

Communication Effort

Figures 5.4A and 5.4B show the communication effort of the three initiative strategies under different parameters. On average, the communication effort of PI is less than that

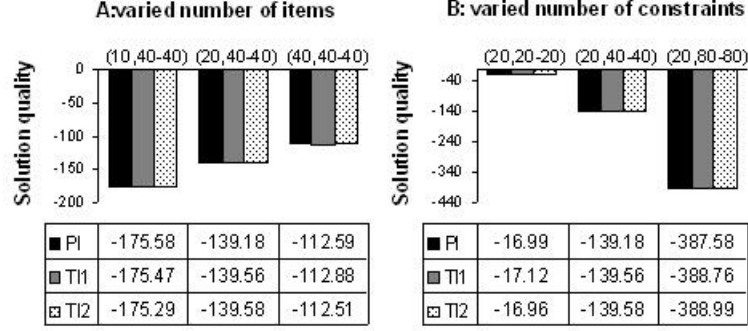


Figure 5.3: Solution quality

of TI1 and TI2. The difference is statistically significant except under the setting (20,20-20) (paired ttest, $p < 0.01$). In fact, the more complex the task is (higher number of constraints or items), the higher the difference is. For example, for the setting (20,20-20), the difference between PI and TI1 is 1.15 speech acts, which is about 8% of PI; for (20,40-40), the difference is 4.9, 25% of PI; for (20,80-80), the difference is about 10, about 37% of PI. Thus we see that the strategy PI saves communication effort, especially for complex tasks.

In terms of speech acts, not only do agents in TI1 and TI2 communicate more constraints (inform) than PI, but they also make more proposals. In the setting of (20,80-80), agents in PI make 14.95 proposals, and in TI1 make 20.10, in TI2 make 20.85. Because each plan takes five proposals, this means that agents in TI1 and TI2 go through more proposal-evaluation cycles. These results suggest that using the strategy PI, agents' proposals are more effective.

Figure 5.5 tracks how the solution quality of the proposed plans changes as a dialogue progresses for the three initiative strategies with the setting of (20,80-80). We see that for the first plan proposed, the average solution quality of PI (about -417) is lower than the other two (about -407). However, after communicating the constraints, agents working in PI are able to create a better plan (-397 vs -402). PI encourages more useful constraints to be communicated between the agents. Thus we see that although PI has a slow start, it quickly makes up for the lost ground, and overtakes the other two strategies.

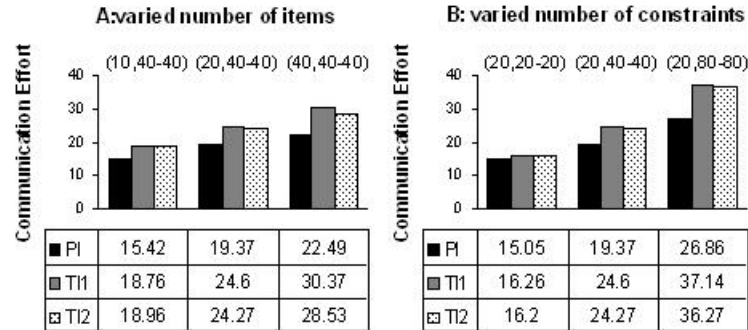


Figure 5.4: Communication effort

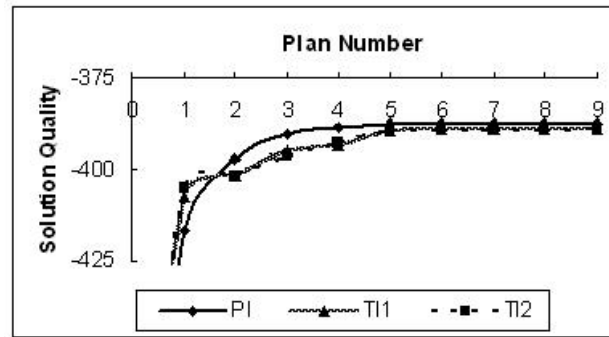


Figure 5.5: Solution quality on each plan

5.4.4 Discussion

Our empirical study in Section 5.3 on the Trains corpus suggested a model where control belongs to the conversant who initiates a dialogue (sub-)goal. Here we provided further support for this finding through the use of computer simulation. We set up a domain and implemented three plan-based problem-solving strategies, the first consistent with the restricted initiative model, and two others in which agents alternated initiative on a turn-by-turn basis, as if control did not exist. We compared the performance of these initiative strategies and found that, under certain domain settings, the restricted initiative strategy produced more efficient dialogues in terms of less communication effort without loss in solution quality. Additionally, as the complexity of the domain increased, so did the advantage of the restricted initiative strategy. These results suggest that efficiency is

a reason why people adopt the restricted initiative model.

Computer simulation opens a new window for examining human language. Our experiments shed light on using computer simulation to better understand human conventions. Using computer simulation, we are able to compare a human conventional strategy (such as PI) versus alternative strategies that are probably not natural to humans (such as TI1 and TI2), in order to understand why people adopt certain conventions in the language evolution.

The simulation experiments in this section, in fact, are pretty exploratory. The purpose of the simulation, rather than provide concrete understanding of human conventional strategy of initiative, is more to research a domain as a test-bed for examining and comparing initiative strategies. In our future work, we plan to extend our current experiments and introduce more sophisticated cognitive models on human reasoning and human memory, to better simulate human behavior. This should allow us to better understand the applicability of the restricted initiative model, and the circumstances in which it does not apply, for example, the 25% of questionable and non-conforming initiative patterns.

5.5 Chapter Summary

We envision that next-generation spoken dialogue systems will be mixed-initiative. Mixed-initiative dialogues do not mean that both conversants show initiative freely without any restrictions. Instead, mixed-initiative means that conversants contribute at the most appropriate timing when it is best suited. Under what circumstances should the system take the initiative, and under what circumstances should it let the user do so? We adopted the approach of using human conventional strategies as a starting point; and we also used computer simulation as an auxiliary approach to better understand the human strategies.

In this chapter, we showed that in the Trains dialogues, the initiator and the non-initiator play different roles in terms of showing initiative to achieve the purpose of a discourse segment. The initiator shows massive initiative consecutively, with the non-initiator mostly responding and occasionally showing initiative. Note that this is not single-initiative, as the non-initiator can show initiative by making a statement, asking a

question, or even initiating a sub-segment; for example, when there is confusion or mistake of the information communicated or the plan proposed. However, the non-initiator's initiative tends to be isolated and surrounded by the initiator's initiative. Therefore, in a discourse segment, control belongs to the initiator. The computer simulation experiments help supporting the explanation that people use this initiative strategy to minimize collaborative effort.

A good initiative strategy for computer dialogue systems should have two features. First, it should be natural for users to follow. This reduces the training effort on users and encourages users to work with the system. Second, it should be efficient in problem-solving. The restricted initiative model, derived from human-human dialogue, should be natural for users to follow. The simulation experiments suggest that the model can save communication effort. According to the *Paradise* evaluation platform, dialogue length is one of the most important factors that lead to improved user satisfaction (Walker et al., 1997b). Thus the restricted initiative model has important implications for dialogue management, as it will pave the way for building dialogue systems that can engage in mixed-initiative dialogues.

Chapter 6

Managing Multiple Tasks

6.1 Introduction

Dialogue systems have been used to perform a single task at a time, where the user finishes one task before moving to the next. However, we envision that future dialogue systems will be able to work with the user on multiple tasks at the same time, which is especially useful for real-time tasks. For instance, a driver in a car might use a dialogue system to catch up on emails, while occasionally checking upcoming traffic conditions, and receiving navigation instructions.

A couple dialogue systems that allow multi-tasking dialogues have been built (Lemon and Gruenstein, 2004; Kun, Miller, and Lenharth, 2004). However, these systems freely switch between different tasks without much signaling (see Section 2.4). Thus the user might be confused about which task the system is talking about. Multi-tasking dialogues, even in the best circumstances, will be difficult for users, as users need to remember the details of each task and be aware of task switching.

In order to build a dialogue system that supports multi-tasking dialogue, there needs to be a set of conventions for the user and the system to follow in switching between tasks. To designing such a set, we propose to start with conventions that are actually used in human-human conversations, which are natural for users to follow and probably efficient in problem-solving. Multi-tasking dialogues, where multiple independent topics overlap with each other in time, regularly arise in human-human conversation: for example, a driver and a navigator in a car might be talking about their summer plans, while occasionally interjecting road directions or conversation about what music to listen to. The

MTD corpus (Heeman et al., 2005) was thus collected to better understand the human conventions on task switching. In an MTD dialogue, players conversed on two types of tasks: an ongoing task of playing poker, and a momentary task of identifying a picture. This chapter is focused on how conversants switch from the ongoing task to a waiting momentary task, including 1) the discourse contexts where task switching occurs, and 2) the cues that are used to signal task switching.

Previous research suggested the correlation between task switching and certain discourse context; for example, conversants try to avoid task switching in the middle of an adjacency pair (Shyrovkov, Kun, and Heeman, 2007). We have also found in the MTD corpus that conversants did not always switch to a momentary task as soon as it arose, but instead waited for different amounts of time depending on the urgency level of the momentary task (Heeman et al., 2005). In this study, we hypothesize that conversants strive to switch at an opportune place in the ongoing task, and we examine the discourse context where task switching occurs for evidence to support this hypothesis.

6.2 Where to Switch

We asked the question whether players strive to switch at an opportune place in the ongoing task (poker-playing). There are three types of places where a player could switch the conversation from poker-playing to a waiting picture game: (G) immediately after completing a poker game (at the end of a game), (C) immediately after discarding a card (at the end of a card), and (E) embedded inside a card segment, where players are deciding which card to discard. In this section, we examine where task switching occurs.

6.2.1 Urgency Level and Place of Switching

We first examine the place of switching under different urgency levels. As shown in Table 6.1¹, at the urgency level of 10s, 71% of the task switching was embedded inside a card segment, 27% at the end of a card, and 2% at the end of a game; at the urgency

¹Of the total 180 picture game prompted, nine were never started by players (although in the post-experiment survey all players reported that they never ignored a momentary task on purpose). Thus we have 171 picture games.

Table 6.1: Urgency level and place of switching

	E	C	G	Total
10s	40 (71%)	15 (27%)	1 (2%)	56 (100%)
25/40s	54 (47%)	37 (32%)	24 (21%)	115 (100%)

levels of 25s and 40s, 47% embedded inside a card segment, 32% at the end of a card, and 21% at the end of a game. The difference in the places of switching between the urgency level of 10s and 25s/40s is statistically different ($\chi^2(2, N = 171) = 13.84, p < 0.001$). Compared with 10s, when players had 25s or 40s, the percentage of switching embedded inside a card segment decreases by 24%, while at the end of a card increases by 5%, and at the end of a game increases by 19%.

The urgency level of 10s requires players to start a picture game very quickly in order to complete it in time. On the other hand, when given 25s or 40s, players are in a less hurry to switch. Our results show that, when players have 25s or 40s, the percentage of switching at the end of a game increases the most, followed by switching at the end of a card, while the percentage of switching embedded inside a card segment drops. These results thus suggest that when given more time, players would try to switch at the end of a game or a card.

6.2.2 Waiting Time and Place of Switching

Waiting time refers to the time interval between when a picture game is prompted to a player and when it is actually started (i.e. asked) by the player. We examine where players switched to a picture game when the waiting time was shorter than 3 seconds versus longer than 3 seconds. We assume that when the waiting time is shorter than 3s, the player starts the picture game as soon as he or she notices it without significant waiting. The results are shown in Table 6.2. When the waiting time is shorter than 3 seconds, 69% of the task switching is embedded inside a card segment, 28% at the end of a card, and only 3% at the end of a game; when the waiting time is longer than 3 seconds, 43% is embedded inside a card segment, 35% at the end of a card, and 22% at the end of a game. The difference in the places of switching is statistically different ($\chi^2(2, N = 171) = 16.52, p < 0.001$).

Table 6.2: Waiting time and place of switching

	E	C	G	Total
$\leq 3s$	47 (69%)	19 (28%)	2 (3%)	68 (100%)
$> 3s$	44 (43%)	36 (35%)	23 (22%)	103 (100%)

When the waiting time is longer than 3 seconds, the percentage of switching inside a card decreases by 26%, while switching at the end of a card increases by 7%, and at the end of a game increases by 19%. These results suggest that players wait for the end of a game or a card to switch to a picture game.

6.2.3 Discussion

We examined the discourse context where task switching occurs, and found that (1) when given more time, players intend to switch to a picture game at the end of a (poker) game or a card; and (2) if players wait, they are waiting for the end of a (poker) game or a card to switch to a picture game. These results suggest that at the end of a game or a card is the opportune place to switch to a waiting picture game.

In fact, we also observed that after a picture game that is at the end of a game, players smoothly start a new poker game as if nothing had happened; after a picture game that is at the end of a card, players might sometimes remind each other what cards they have in hands; while after a picture game that is in the middle of a card segment, players might even repeat or clarify the previous utterances before the interruption. It is thus reasonable to assume that switching embedded inside a card segment is the most disruptive, followed by at the end of a card, and switching at the end of a game is the least. Our experiment results hence suggest that players strive to switch to a momentary task at a less disruptive place in the ongoing task. This is consistent with Clark and Wilkes-Gibbs (1986), that conversants try to minimize collaborative effort.

6.3 How to Switch

In Section 2.2.2, we discussed how people use certain cues, such as discourse markers and prosody, to signal topic shifts in single-tasking speech. This suggests that people might

also signal task switching in multi-tasking dialogues. In this section, we examine how players signal that they are switching from the ongoing task to a momentary task with discourse markers and prosody.

6.3.1 Task Switching and Discourse Markers

Close examination of the MTD corpus found that “oh” was the most often used discourse marker when switching to a picture game. Another discourse marker, “wait” (including “wait a minute”), was often used together with “oh” in the way of “oh wait”. Thus we examined the use of “oh” and “wait” in switching to a picture game.

Players used the discourse markers “oh” or “wait” 14.6% (25/171) of the time in switching to a picture game. In poker playing, 5.7% (238/4192) of utterances contain the words “oh” or “wait”, and only 4.6% (32/690) of card segments are initiated with the two discourse markers (i.e. the first utterance of a card segment has “oh” or “wait” at the very beginning). Players have a statistically higher percentage of using the discourse markers “oh” or “wait” at task switching than in poker playing ($\chi^2(1, N = 4363) = 23.19$, $p < 0.001$) or to initiate a card segment ($\chi^2(1, N = 861) = 22.09$, $p < 0.001$).

6.3.2 Task Switching and Prosody

To understand the prosodic cues in initiating a topic, traditionally researchers compared the prosody of the first utterance in each segment with other utterances (e.g. (Nakajima and Allen, 1993; Hirschberg and Nakatani, 1996)). This approach encounters two problems here. First, the words in an utterance might affect the prosody. For example, the duration and energy of “bat” are usually larger than “bit”. Thus a large amount of data are required to balance out these differences in order to get statistically significant results. Second, in the MTD corpus, players typically switch to a picture game by using a yes-no question, such as “do you have a blue circle”, while most forward utterances (c.f. Core and Allen 1997) in the poker game are statements or proposals. As questions have very different prosody than statements or proposals, a direct comparison is further biased.

Examination of the MTD corpus found that 86% (147/171) of the picture games were initiated by “do you have ...” with optional discourse markers at the beginning. While in

Table 6.3: Average prosodic values for each player

Player	pitch (Hz)		energy		duration (s)	
	SWT	PKR	SWT	PKR	SWT	PKR
4A	136	123	335	222	0.28	0.38
4B	178	156	408	453	0.32	0.30
5A	164	152	314	286	0.37	0.25
5B	214	182	221	146	0.36	0.29
6A	144	124	375	306	0.31	0.20
6B	122	117	514	428	0.25	0.23
8A	238	199	850	775	0.36	0.21
8B	150	143	231	141	0.33	0.35
9A	110	102	455	342	0.44	0.59
9B	125	122	693	641	0.32	0.24

the poker game, players used “do you have ...” 131 times to ask whether the other had certain cards, such as “do you have a queen?” This observation inspired us to compare the prosody of the phrase “do you have” in switching to a picture game and during poker-playing.² This avoids comparing prosody of different words or of different types of utterances.

We measured pitch, energy (local root mean squared measurement), and duration of each case of “do you have”. We aggregated on each player and calculate the average values. The results are shown in Table 6.3. The second and third columns show the average pitch of the phrase “do you have” for task-switching (SWT) and poker-playing (PKR) respectively. When switching to a picture game, players’ pitch is statistically higher than poker-playing ($t(9) = 4.03, p < 0.01$). We also examined each individual player’s use of pitch on switching to a picture game. For all ten players, the average pitch of “do you have” in switching to a picture game is higher than in poker-playing; and the differences are statistically significant at the alpha level of 0.075 (using ttest), except for the players 8A and 9B. These results show a strong correlation between task switching and higher pitch.

We next examine the correlation between energy and task switching. The fourth

²Note that most cases of “do you have” in poker-playing are not at the beginning of a card segment. It is also interesting to compare the prosody of “do you have” of initiating a picture game and of initiating a card segment. However, we do not have enough data for the latter.

and fifth columns in Table 6.3 show the average energy of the phrase “do you have” for task switching and poker-playing respectively. Again, when switching to a picture game, players’ energy is statistically higher than for poker-playing ($t(9) = 4.45, p < 0.01$). Nine out of the ten players’ energy of “do you have” in switching to a picture game is higher than in poker-playing. However, it is the reverse way for player 4B. Except for the players 5B and 6B, none of the differences in each individual player are statistically significant at the alpha level of 0.075. Thus, although there is a tendency that higher energy correlates to task switching, the correlation does not seem as strong as pitch.

We also examine the duration of “do you have”. The sixth and seventh columns in Table 6.3 show the results. We do not find a statistically significant difference ($t(9) = 0.98, p = 0.35$). Seven out of the ten players’ duration of “do you have” in switching to a picture game is higher than in poker-playing, and the other three player (4A, 8B, and 9A) have the reverse way. These results do not support that duration (or speaking rate) is correlated to task switching.

6.3.3 Signaling Place of Switching

In the previous subsection, we showed that pitch strongly correlates with task switching. To better understand how pitch is used in signaling task switching, we next examine its correlation with place of switching, i.e., switching at the end of a game, at the end of a card, or embedded inside a card segment. Because there are relatively less data for switching at the end of a game (see Table 6.1 and 6.2), we combine switching at the end of a game and at the end of a card (C & G) as a category.

Table 6.4 shows the average pitch of “do you have” when switching to a picture game embedded inside a card segment, at the end of a card or game segment, and during poker-playing. The difference between these three conditions is statistically significant ($F(2, 9) = 14.50, p < 0.01$). Switching embedded inside a card segment has a statistically higher pitch than switching at the end of a card or game segment ($t(9) = 4.94, p < 0.01$), which in turn has a statistically higher pitch than during poker-playing ($t(9) = 2.71, p = 0.01$).

A similar procedure is carried out for energy. However, we did not find that switching

Table 6.4: Pitch (Hz) and place of switching

Player	E	C & G	POKER
4A	137	131	123
4B	180	173	156
5A	167	161	152
5B	219	206	182
6A	146	143	124
6B	125	121	117
8A	253	230	199
8B	152	140	143
9A	111	104	102
9B	130	115	122

embedded in a card segment has a statistically higher energy than switching at the end of a card or game segment ($t(9) = 1.23$, $p = 0.12$).

6.3.4 Discussion

Consistent with previous research on topic shifts in single-tasking speech, our experiments show that switching to a momentary task correlates with the use of certain discourse markers and prosodic variations. It is not surprising that “oh” and “wait” correlate with task switching. Task switching involves a sudden change of the conversation topic, and previous research found that conversants use “oh” to mark a change of state in orientation or awareness (Heritage, 1984). “Wait” is used to mark a discontinuity in the ongoing topic, which is also required by task switching. Thus people may use these discourse markers to signal switching to a momentary task. In terms of prosodic variations, pitch seems to be the most pertinent feature. We find that task switching correlates with higher pitch. This suggests that pitch is used to signal switching to a momentary task.

Our experiments have also shown that pitch correlates to place of switching. As discussed in Section 6.2.3, task switching embedded inside a card segment is the most disruptive, switching at the end of a card is less, and at the end of a game is the least. Our results on place of interruption show that switching embedded in a card segment has a higher pitch than switching at the end of a card or a game, which in turn has a higher pitch than non-switching (poker-playing). This suggests that the degree of disruptiveness

corresponds to the value of pitch: the more disruptive place to switch, the higher is the pitch.

From our results we speculate that pitch is used to divert the hearer from the ongoing task, signaling an unexpected event (c.f. (Sussman, Winkler, and Schrgöer, 2003)). When task switching is more disruptive, the speaker uses higher pitch; probably because the hearer has a stronger expectation of the next utterance to be in the context of poker-playing. The use of higher pitch serves as a cue that the hearer should suspend the ongoing context and interpret the utterance in a new context. According to the theory of *least collaborative effort*, the effort of raising the pitch by the speaker is probably to reduce the effort of recognizing and processing the task switching by the hearer (Clark and Wilkes-Gibbs, 1986).

6.4 Chapter Summery

In this chapter we describe an empirical study of human-human multi-tasking dialogues, where people perform multiple verbal tasks overlapped in time. We examined how conversants switch from the ongoing task to a momentary task. We found that 1) conversants use discourse markers and prosodic cues to signal task switching, similar to how they signal topic shifts in single-tasking speech; 2) conversants strive to switch tasks at a less disruptive place; and 3) where they cannot, they exert additional effort to signal the task switching. These findings will provide guidelines for building future speech interfaces to support multi-tasking dialogue.

Chapter 7

Managing Initiative Conflicts

7.1 Introduction

In the previous chapters, we have examined how conversants collaborate on managing complex goals and managing multiple tasks.¹ Such collaboration requires that both conversants are allowed to show *initiative*, or direct the conversation. However, this opens up the possibility of *initiative conflicts*, where both conversants try to show initiative at the same time. Figure 7.1 gives an example of an initiative conflict from the MTD corpus. The bottom conversant said “how many threes do you have”, which was overlapped by the top conversant’s utterance “I’ll drop”. The bottom conversant was checking whether they together had a poker hand of full house; while the top conversant was suggesting to discard a card. Each utterance would lead the dialogue in a different direction. In this example, the top conversant gave up the attempt to show initiative.

Although there are some mixed-initiative systems that allow the user to make a request or state a goal, such as the ConQuest system (Bohus et al., 2007), such systems are limited in how they resolve *initiative conflicts*. For example, ConQuest always stops speaking whenever it detects that the user is speaking. However, the system, just like the user, might need to override the other conversant to correct a misconception, or to convey an idea. Only with such a capability will next-generation dialogue systems be able to truly collaborate with the user on complex goals or multiple tasks. Thus the next generation of computer dialogue systems need a better model of initiative that specifies under what

¹Content of this chapter was previously published in (Yang and Heeman, 2007a) and also has been submitted for journal publication.

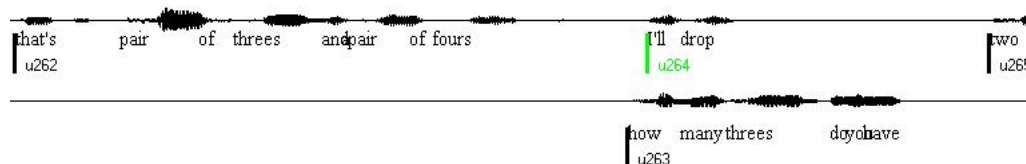


Figure 7.1: An example of an initiative conflict

situations initiative conflicts occur, what mechanisms are used to resolve them, and how to repair any impairment to the conversation that the conflicts cause.

Previous research on initiative created a lot of models (see Section 2.3), and all of these models allowed the possibility that multiple conversants want to show initiative at the same time. However, none of them addressed the problem of initiative conflicts. Research on turn-taking revealed some observations on how conversants fight for the speaking turn, yet lacked statistical evidence (see Section 2.1). In this chapter, we examine human-human dialogues in two corpora, the Trains corpus and the MTD corpus, to understand the conventions that people adopt in managing initiative conflicts. These conventions can then form the basis for a computer dialogue system, which will allow natural interactivity.

In the rest of this chapter, we first describe how we identified initiative conflicts in the two corpora. We then present the evidence that conversants try to avoid initiative conflicts in Section 7.3. We next present the evidence that initiative conflicts are efficiently resolved using volume as a device in Section 7.4. In Section 7.5, we describe the behavior of repairing speech after initiative conflicts. We discuss our findings in Section 7.6.

7.2 Identifying Initiative Conflicts

An initiative conflict occurs when both conversants show initiative at the same time, and are trying to steer the conversation in two different directions. As in Chapter 5, we used utterance tags to define whether an utterance shows initiative: forward functions show initiative while stalls and backward functions do not. Thus, overlaps involving a stall or a backward function are not counted as initiative conflicts. In the Trains corpus, there are 142 cases of overlapping speech. Of these, 96 cases involve a backward function (e.g. an acknowledgement overlapping an inform), and 10 cases involve a stall. In the MDT

corpus, there are 383 cases of overlapping speech. Of these, 182 cases involve a backward function and 21 cases involve a stall.

Of the remaining 44 cases of overlapping utterances in the Trains corpus, 28 cases are overlaps of a forward utterance and a non-contribution, or of two non-contributions. In the MTD corpus, there are 103 such cases. Non-contributions are abandoned or ignored utterances. Hence, these are prime candidates for initiative conflicts, as an expected outcome of initiative conflicts is that at least one of the speakers fails the attempt to show initiative. However, as non-contributions are often incomplete, it is not always possible to determine whether the speaker intends to show initiative with it. Nonetheless, they are our best guess of where initiative conflicts occur. Hence, we identified overlaps between a non-contribution and either another non-contribution or a forward utterance as initiative conflicts.

The remaining 8 cases in the Trains corpus and 77 cases in the MTD corpus are overlaps of two forward utterances, where both speakers show initiative. We found that these cases fall into two types: *early-starts* and *consistent-contents*. Early-starts are where the second utterance starts before the end of the first utterance, but the second utterance responds to the first utterance. Thus, no initiative conflict occurs. Consistent-contents are where speakers are talking about the same thing: for example, one saying “we are a bit early” and the other saying “we are a little better”. As both utterances steer the conversation in the same direction, no initiative conflict occurs for this type as well.

To discuss initiative conflicts, we introduce the following terms. We refer to the speaker who makes the forward utterance as the *winner* and the other who makes the non-contribution utterance as the *yielder*. We use the term *preceding-pause* to refer to the time interval between the end of the previous utterance and the beginning of the first utterance involved in the overlap (from A to B in Figure 7.2). *Offset* refers to the interval between the start times of the two overlapping utterances (from B to C). *Overlap* is where both conversants are speaking (from C to D).

There are 3 cases of initiative conflicts in the Trains corpus and 13 cases in the MTD corpus of which the preceding-pause is negative, i.e. the first overlapped utterance is started before the other conversant finishes the previous utterance. Sometimes the hearer

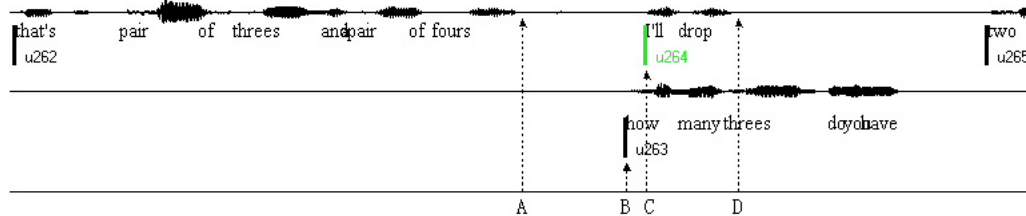


Figure 7.2: Preceding-pause (A-B), offset (B-C), and overlap (C-D) of initiative conflict

starts a little bit early to take the turn. If the original speaker does not release the turn, a conflict arises. Because these conflicts involve three utterances, we exclude them from our current analysis and save them for future research. This leaves 25 cases in the Trains corpus and 90 cases in the MTD corpus for analyzing initiative conflicts.

7.3 Avoiding Initiative Conflicts

In this section, we show that conversants try to avoid initiative conflicts, by examining both the offset of initiative conflicts and the urgency of utterances.

7.3.1 Offset of Initiative Conflicts

The offset of an initiative conflict indicates where the conflict happens in the first utterance involved in the conflict. A short offset indicates that the conflict happens at the beginning of the utterance, while a long offset indicates an interruption in the middle. Figure 7.3 shows the cumulative distribution function (CDF) for offsets of both corpora individually. The mean offset is 138ms for the Trains corpus and 236ms for the MTD corpus. In comparison to the average length of forward utterances (2596ms in the Trains corpus and 1614ms in the MTD corpus), the offset is short. Moreover, in the Trains corpus, 88% of offsets are less than 300ms (and 80% less than 200ms); in the MTD corpus, 75% of offsets are less than 300ms. Thus most initiative conflicts happen at the beginning of utterances.

Few initiative conflicts have offsets longer than 500ms. There is one instance in the Trains corpus and eleven in the MTD corpus, which in total account for 10.4% of all initiative conflicts. Six of these cases are due to ambiguous intonation. Figure 7.4 shows an example. Conversant B said “I have two aces” with end-of-utterance intonation, paused

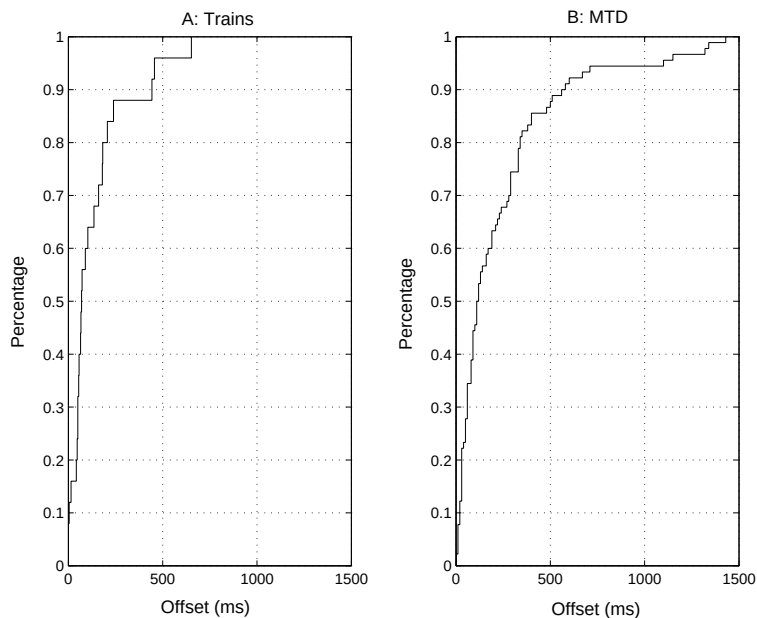


Figure 7.3: CDF plot for offset of initiative conflicts

for about half a second, and then added “and a seven”. The ending intonation and the pause probably misled conversant A to believe that B had finished, and thus A started a new forward utterance, which overlapped with B’s extension. A’s utterance was then quickly abandoned. In these cases, it is ambiguous whether B’s utterance “I have two aces ... and a seven” should be segmented into two utterances. Another four cases are because the second conversant has something urgent to say. For example, when an interruption game is timing out, conversants might interrupt, sometimes in the middle of an utterance, which results in a long offset. For the last two cases, there is not a simple reason why the second speaker chooses to interrupt so late into the first speaker’s utterance. Figure 7.5 shows an example. In listening to the excerpt, the second speaker comes across as being rude by ignoring what the first speaker is saying, and overriding it with his own utterance. Such examples are rare, and might depend on personality and social roles.

What accounts for the short offsets of initiative conflicts? We argue that this is due to conversants trying to avoid initiative conflicts. When A detects that B is talking, A should not attempt to show initiative until the end of B’s utterance in order to avoid conflicts, unless there is an urgent reason. If conversants did not take into account whether someone

B:	I have two aces	and a seven
A:	I have	.

Figure 7.4: Long offset: ambiguous intonation

B:	well let's just
A:	it's no help I think it goes away

Figure 7.5: Long offset: other

else is speaking before attempting to show initiative, we would see a lot of intrusions in the middle of utterances, which rarely happens in the two corpora. As we have shown, initiative conflicts tend to happen at the beginning of utterances. Thus initiative conflicts occur mainly due to unintentional collision, i.e. both conversants happen to start speaking almost at the same time. The fact that the offsets of most initiative conflicts are less than 300ms supports this. This 300ms is probably related to human reaction time. It has been reported that human reaction time of hitting a button for a sound stimuli is only about 160ms (Galton, 1899). However, in the task-oriented dialogues of the both corpora, conversants have higher cognitive load, and might not be paying as much attention to sound stimuli as under Galton's experimental conditions. Thus it can be expected that the reaction time for avoiding initiative conflicts is longer than 160ms (c.f. Lee et al. 2001).

7.3.2 Urgency Level and Initiative Conflicts

To further support the argument that conversants try to avoid initiative conflicts, we examined how the urgency levels for the interruption game in the MTD dialogues affected the occurrence of initiative conflicts. For the urgency level of 10 seconds, conversants started 33 interruption games, 8 of which were introduced via initiative conflicts. For 25 seconds, conversants started 36 interruption games, 5 introduced via initiative conflicts. For 40 seconds, conversants started 33 interruption games, 3 introduced via initiative conflicts. Thus the percentages of initiative conflicts for the three urgency levels are 24% for 10 seconds, 14% for 25 seconds, and 9% for 40 seconds. The urgency level of 10 seconds requires conversants to start the interruption game very quickly in order to complete it

in time. On the other hand, the urgency level of 40 seconds allows conversants ample time to wait for the best time to start the game (Heeman et al., 2005). The percentage of initiative conflicts increases as it becomes more urgent to start the interruption game. These results suggest that conversants try to avoid initiative conflicts if they can, unless there is an urgent reason. In an urgent situation, conversants probably lower the threshold of evidence that they use for guessing whether the other also wants to show initiative, which results in more initiative conflicts.

7.4 Resolving Initiative Conflicts

In this section, we present evidence that initiative conflicts, if they occur, are resolved very quickly using simple devices.

The duration of overlap indicates how quickly an initiative conflict is resolved. Figure 7.6 shows the cumulative distribution function of the overlap of initiative conflicts and the length of forward utterances in the two corpora. The average duration of overlaps is 328ms in the Trains corpus and 427ms in the MTD corpus. From Figure 7.6, we see that the duration of overlaps is much shorter than the length of forward utterances, which on average is 2596ms in the Trains corpus and 1614ms in the MTD corpus as reported in Section 7.3. On average, the duration of initiative conflicts is about 1/8 the length of forward utterances in the Trains corpus and about 1/4 in the MTD corpus. The short durations suggest that initiative conflicts are resolved very quickly.

According to Crystal and House (1990), the average length of CVC syllables is about 250ms. Thus on average, the length of initiative conflicts is about one to two syllables.² In fact, 96% of conflicts in the Trains corpus and 73% in the MTD corpus are resolved within 500ms. These observations are consistent with Schellogff's claims (2000) about turn-taking fights, that they usually take less than two syllables to resolve.

Given the short overlaps, there must be a simple mechanism that conversants use to resolve initiative conflicts. This mechanism might either be based on features present

²It would be interesting to examine the length of initiative conflicts based on syllable. However currently we do not have syllable-level alignment for the two corpora. We leave this for future research.

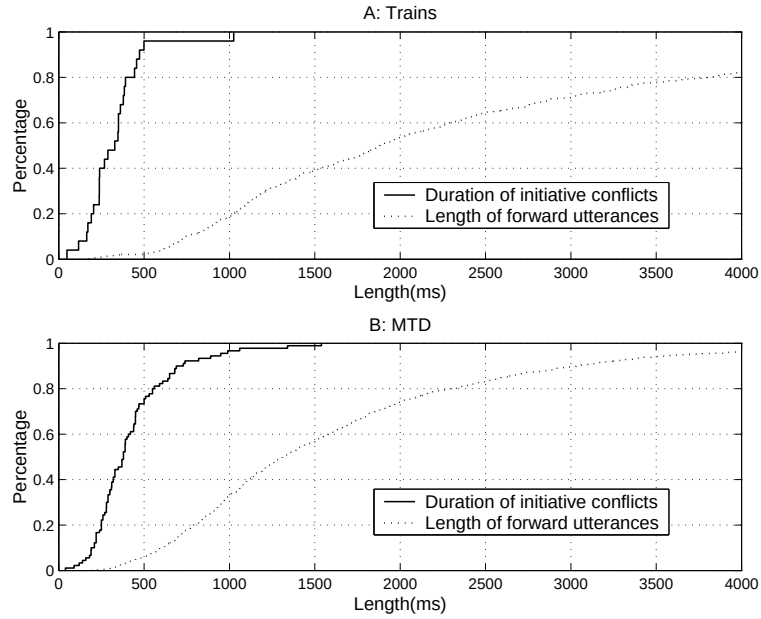


Figure 7.6: CDF plot for duration of overlaps and length of forward utterances

during the overlap, such as volume, or based on features from the context, such as discourse roles and domain roles.

7.4.1 Features during the Overlap

We first examined two features present during the overlap, volume and pitch. There are two cases of initiative conflicts in the Trains corpus and three cases in the MTD corpus in which both speakers abandoned their incomplete utterances almost at the same time, paused for more than 500ms, and then one of the speakers resumed talking. These five cases were treated as ties: no winners or yielders, and were excluded from our analysis here.

Correlation between Volume and Winning Initiative Conflicts

For a stretch of speech, volume is calculated as the mean energy of the spoken words. For each initiative conflict, we calculated each speaker's volume during the overlap, and then

normalized it with respect to the speaker’s volume throughout the whole conversation.³ We refer to this as *relative volume*. In the Trains corpus, the average relative volume of the winner is 1.06; the average relative volume of the yielder is 0.93. The difference is statistically significant ($t(22) = 3.55$, $P = 0.002$). In the MTD corpus, the average relative volume of the winner is 1.12; the average relative volume of the yielder is 0.98. The difference is also statistically significant ($t(86) = 6.45$, $p < 0.001$). These results show that the winner is the one speaking at a higher relative volume.

To strengthen our argument, we also calculated *volume ratio* as the relative volume of the winner divided by the yielder’s. The average volume ratio is 1.16 in the Trains corpus and 1.18 in the MTD corpus. If a classifier always chooses the speaker with the higher relative volume as the winner, it achieves about 79% accuracy in both corpora, which is a 29% absolute improvement over random prediction. These results further confirm the correlation between higher volume and winning initiative conflicts.

As an aside, we analyzed the 23 cases in the two corpora for which volume fails to predict the winner. Eight cases seem due to inaccurate measurement of volume. In these cases, although the energy of the winner is lower, the winner sounds to be louder around the overlap. The inaccuracy comes from several sources. For example, the microphone might pick up air flow beyond vibration: an unvoiced ‘s’ (or even breathing) could have a high energy because of the speaker blowing air into the microphone; or during a short overlap the yielder is pronouncing a stressed syllable or accented word, while the winner is pronouncing an unstressed syllable or an unaccented word. There are two cases in which one of the utterances is switching to the interruption task, and that utterance is the one that wins. There are three cases involving a question overlapping a statement and the question wins. The final ten cases are still unclear. We speculate that some of these cases might be related to unexpected interference of overlapping speech. For example, the yielder forgets what he or she wants to say because the other speaker cuts in.

Given the importance of volume in the resolution process, we examined whether it has an impact on the duration of overlap. When volume is able to predict the winner

³Normalization is necessary particularly as conversants heard each other via headsets, and the microphones were not calibrated to have exactly the same gains.

of initiative conflicts, the correlation between volume ratio and the duration of overlap is statistically different from random and shows a negative tendency (pearson R test, $r(83) = -0.27$, $p = 0.01$), that is, the lower the volume ratio, the longer the overlap of initiative conflicts. In fact, the average volume ratio for initiative conflicts shorter than 600ms is 1.27, while for those longer than 600ms is 1.13.

Volume as a Device for Resolving Initiative Conflicts

Although there is a correlation between volume and resolving initiative conflicts, it is unclear whether volume is a *cause* or an *effect* of the resolution process. Volume could be an effect if a speaker, once he or she decides to yield, is unable to cease speaking immediately, but is able to lower the volume of speech. If volume is a cause then this implies that speakers use volume as a device in resolving initiative conflicts, such as signaling their desire to show initiative. Obviously, to build a dialogue system capable of engaging in natural interactivity, it is important to understand how volume is used in initiative conflicts. In this section, we provide evidence that volume is a cause: conversants use it as a device to resolve initiative conflicts.

To understand how volume is used in resolving initiative conflicts, we examined how volume changes during the overlap. For the 16 cases of initiative conflicts whose duration is longer than 600ms, we segmented the overlap evenly in half, and calculated the volume of each half. If the hypothesis that speakers use volume as a device to resolve initiative conflicts is true, we would expect to observe:

1. Similar volumes during the first half, as the initiative conflicts are not resolved until the second half. What speakers do in the second half then becomes the critical region in resolving initiative conflicts.
2. Increased volume from the first half to the second half. If both speakers, or at least the yielder, lowers the volume, it can be an indication that volume is a side-effect of losing the turn; but if both speakers, or at least the winner increases the volume, it can be an indication that volume is used as a device to negotiate initiative.

For the first half of these long initiative conflicts, the average volume of the winner is

1.03, and the yielder is 1.02. The difference is not statistically significant ($t(15) = 0.16$, $p = 0.87$). The volume ratio of the first half, on average, is 1.02 with a standard deviation of 0.04. Thus speakers have similar volume values on the first half. In fact, if we predict the winner by the volume of the first half, we only achieve an accuracy of 56%. These results fulfil the first expectation.

For the second half, the average volume of the winner is 1.20, and the yielder is 1.02. The difference is statistically significant ($t(15) = 2.77$, $p = 0.01$). The volume ratio, on average, is 1.22 with a standard deviation of 0.08. This is much closer to the volume ratio of initiative conflicts lasting less than 600ms. In fact, if we predict the winner by the volume of the second half, we achieve an accuracy of 75%. These results suggest that the second half is the critical region where these long initiative conflicts are resolved.

We also examined how the volume of both the winner and the yielder changes from the first to the second half. For the yielder, in 8 cases, volume increases from the first half to the second half; in another 7 cases, volume decreases; and there is one case of no change. Thus it is inconclusive whether volume is a side-effect of resolving initiative conflicts. On the contrary, for the winner, the average volume increases by 17% (from 1.03 to 1.20). The increment is statistically significant ($t(15) = 4.0$, $p = 0.001$). In fact, in 13 out of the total 16 cases, the winner increases volume from the first half to the second half. These results fulfil the second expectation of volume being used as a device: a speaker increases the volume to win initiative during the conflict.

Pitch

We used the tool *WaveSurfer* (Sjölander and Beskow, 2000) to extract f0 values from the audio files. We calculated *relative pitch* similarly as we did for volume. In the Trains corpus, the average relative pitch of the winner is 1.02; the average relative pitch of the yielder is 0.96. The difference is not statistically significant ($t(22) = 0.58$, $P = 0.57$). In the MTD corpus, the average relative pitch of the winner is 1.09; the average relative pitch of the yielder is 0.98. The difference is statistically significant ($t(86) = 4.1$, $p < 0.001$). If we choose the speaker with higher pitch to be the winner, we achieve about 65% accuracy in the Trains corpus and 62% in the MTD corpus. We also examined maximum pitch

and pitch range during the overlap. We did not see a significant correlation between these features and winning initiative conflicts. These results suggest that pitch alone is less effective in resolving initiative conflicts, although there seems to be the tendency of higher pitch by the winner.

7.4.2 Features from Context

We also examined whether features from the context, such as who spoke before the overlap, discourse roles, and domain roles, have an effect on how they are resolved.

Speaker before Overlap

If we predict that the conversant who spoke the utterance prior to the conflict (e.g. speaker of u262 in Figure 7.2) yields, we achieve 55% accuracy in the Trains corpus and 61% accuracy in the MTD corpus. If we predict the conversant who spoke the first utterance involved in the overlap (e.g. speaker of u263 in Figure 7.2) wins, we achieve 60% accuracy in the Trains corpus and 53% accuracy in the MTD corpus. Given that chance is 50%, these results suggest that the conventions through which initiative conflicts are resolved are not based on who spoke before the overlap.

Role of Conversants

We examined the impact of discourse roles (initiator vs non-initiator of a discourse segment (Grosz and Sidner, 1986)) and domain roles (system versus user) on resolving initiative. We focused on the Trains corpus due to both its rich discourse structure and the difference in the roles that the system and the user have with respect to each other.

In the Trains corpus, if we predict that the initiator of a discourse segment wins the conflict, we get 65% accuracy. In system-initiated segments, the system wins all eight conflicts; however, in user-initiated segments, the user only wins seven and the system wins eight. The user does not have an advantage during initiative conflicts in its segments. Moreover, if the initiator had an advantage, we would expect the system to have fought more strongly in the user-initiated segments in order to win. However, we did not find that the relative volume of the system winning in user-initiated segments is statistically

higher than in system-initiated segments in this small sample size ($t(12) = 0.13$, $p = 0.90$). The initiator does not seem to have a privileged role in the resolution process.

From the above analysis, we see that the system wins the conflicts in 16 out of the 23 cases. Thus if we predict that the system always wins the conflicts, we achieve 70% accuracy. This is not surprising as the system has all the domain information, and is more experienced in solving problems. If the system and user want to speak at the same time, both would know that the system probably (but not always) has a more significant contribution. That the system wins most of the initiative conflicts agrees with Guinn (1998) that capability is an important feature in determining who shows initiative next.

7.5 Repairing Speech after Initiative Conflicts

According to Sacks et al. (1974), the communication channel can be used for information transmission by one conversant at a time. It has been claimed that overlapping speech, especially fights for turns, might impair the ongoing conversation (Jefferson, 1973; Schegloff, 1987). For example, the overlapped speech might not be heard by the other conversant, or conversants might forget what they want to say. Schegloff (1987) proposed that one technique to protect the conversation is to repeat the words from the beginning of the overlap. In this section, we examined whether conversants use this technique after initiative conflicts.

We found that after initiative conflicts are resolved conversants usually just continue what they have been saying, but sometimes repeat all or some of the overlapped words. In both corpora, we found a total of 14 cases of repeating words immediately after initiative conflicts are resolved, which accounts for about 13% of all initiative conflicts. Although it is slightly higher than the percentage of forward utterances containing a repetition repair, which in both corpora is about 8.5%, the difference is not statistically significant ($\chi^2(1)=2.15$, $p = 0.14$). Thus there is no strong evidence that initiative conflicts lead to more frequent repetition repairs, or that conversants are more likely to repair speech after initiative conflicts.

More analysis, however, shows that repetition repairs tend to follow initiative conflicts

with short offsets, long durations, and volume ratios close to 1. On average, initiative conflicts followed by a repetition repair have an offset of 97ms, a duration of 526ms, and a volume ratio of 1.08. In fact, if we look at initiative conflicts with offsets shorter than 250ms, durations longer than 250ms, and volume ratios lower than 1.2, the percentage of repaired initiative conflicts increases to 26%. This is about three times the percentage of forward utterances that contain a repetition repair, and the difference is statistically significant ($X^2(1)=14.83$, $p < 0.001$). Short offsets imply that neither speaker is expecting the speech from the other. Long durations imply that a high number of the words (or syllables) are overlapped, making it harder for the yielder to determine the utterance. Volume ratios less than 1.2 imply that the yielder might not have heard the utterance due to the relatively low volume of the winner. We conjecture that such initiative conflicts thus have a higher risk of impairing the conversation, which is why a higher percentage of them are repaired. Thus the winner might only repair the conversation only if doing so is warranted.

7.6 Discussion

In this chapter, we presented an empirical study of human behavior in initiative conflicts. Our first finding is that most initiative conflicts are unintentional collisions when both conversants happen to start speaking at almost the same time. Conversants try to avoid initiative conflicts except for urgency reasons. The explanation for this, we argue, is that initiative conflicts add overhead to the conversation, both in the effort of the yielder making the non-contribution utterance, the effort in resolving the initiative conflict (e.g., increasing the volume to win), and in the repair that is sometimes necessary. According to the theory of *least collaborative effort* by Clark and Wilkes-Gibbs (1986), it only makes sense for conversants to interrupt when the loss of not interrupting is higher than the cost of an initiative conflict. Thus the theory of least collaborative effort is consistent with our conclusion that most initiative conflicts are unintentional collisions, except where conversants interrupt in the middle of an utterance for urgency reasons.

The second finding is that initiative conflicts, when they occur, are efficiently resolved.

We found that volume plays an important role: the louder speaker wins. Although Duncan and Niederehe (1974) suggested that volume is redundant to cues of body movement in predicting the winner of initiative conflicts, our results show the effectiveness of using volume alone for this prediction task (where the cues of body movement are not available to the other speaker). We also show how speakers change their volume to resolve initiative conflicts. Conversants probably identify their eagerness of speaking, confidence in what they want to say, and capability of achieving the current goal by means of volume, which helps resolving initiative conflicts very quickly. The explanation is also consistent with the theory of least collaborative effort. Consider the dialogue excerpt in Figure 7.7. In this example, A had the nine of hearts, king of hearts, and queen of clubs; B had the seven of diamonds, king of diamonds, and queen of diamonds. A picked up a new card, the seven of spades. An initiative conflict happened with the overlap of utterances u3 and u4: B suggested going with their previous strategy of a full house of queens and kings (ignoring the pair of sevens), and A suggested keeping all three pairs. In this conflict, A won the turn to speak with a louder voice. However, when B got the turn to speak in u5, B resumed the attempt to direct the conversation towards his proposal. In fact A and B conducted a negotiation in utterances u5 to u11 to decide which poker hand to pursue. This example shows that yielding in an initiative conflict does not mean that the conversant loses control of the conversation. Instead, winning an initiative conflict is just one means by which conversants direct the conversation. As most utterances only take several seconds, it does not make much sense for conversants to engage in a complex process to resolve initiative conflicts, when they can wait for the winner to finish, and then make their contribution.

The third finding is that domain settings have an impact on conversants' initiative behavior. There are more frequent initiative conflicts in the MTD corpus than in the Trains corpus; and initiative conflicts are resolved faster in the Trains corpus than in the MTD corpus. Moreover, the domain roles of the conversants also affect their initiative behavior as we found that the system wins more initiative conflicts in the Trains corpus. Likewise, in a teacher-student conversation, one might expect that the teacher interrupts

u1	A:	I got seven of spades
u2	B:	I have a seven as well
u3	B:	but we're + going for +
u4	A:	+so now we have three+ pairs
u5	B:	we're going for queen king
u6	A:	yeah
u7	B:	and drop the seven then
u8	A:	alright I'll drop the nine
u9	B:	is it a diamond?
u10	A:	no it's a heart
u11	B:	okay

Figure 7.7: Negotiating initiative

the student more often than vice versa, but also that the teacher wins more initiative conflicts. Capability, culture, and social relationship probably are some underlying elements that influence when and under what conditions conversants would seek initiative.

The last finding is that initiative conflicts do not seem to always cause an obstacle to the ongoing conversation. In most cases, after initiative conflicts are resolved, conversants continue as if nothing had happened. However, initiative conflicts under certain conditions, such as short offsets, long durations, and similar volumes, might have a higher risk of impairing the conversation. A technique that conversants employ to protect the conversation is to repeat the overlapped words immediately after the conflict.

7.7 Chapter Summery

Next-generation spoken dialogue systems should be mixed-initiative, as will be managing complex goals and multiple tasks. Initiative conflicts naturally arise in mixed-initiative and multi-tasking conversations. In this chapter, we have examined human conventional behavior of initiative conflicts. Our findings will help us build next-generation mixed-initiative spoken dialogue systems to manage complex goals and multiple tasks.

Chapter 8

Conclusion

The contribution of this thesis is twofold. First, from the perspective of linguistics, it provides insight into human language in terms of how people jointly direct the flow of conversation. We examined how people collaborate in leading the conversation flow to achieve the dialogue goal, how people switch the conversation flow to a more urgent task, and how people get into initiative conflicts and resolve the conflicts.

Second, from the perspective of human-computer interaction, this thesis provides guidelines to build next-generation dialogue systems. Human conventions are a good starting point of building a dialogue manager. Once we understand the human conventions in directing the conversation flow, we can implement them in a dialogue system to allow for natural and efficient human-computer interactivity.

8.1 Human Conventions on Directing the Conversation Flow

We envision that next-generation dialogue systems will be working with the user on complex goals and multiple tasks. One problem with current dialogue systems is that they lack an effective model of initiative. The research of this thesis thus aimed to create a model to guide the system's initiative behavior of jointly directing the conversation flow with the user. We proposed to start with conventions actually used in human-human dialogue, which are natural for users to follow and probably efficient in problem-solving. To facilitate the research on modeling human-human dialogue, we have built the tool DialogueView to graphically display a dialogue in multiple views, with each view presenting an abstraction of the dialogue. DialogueView can be used for annotating speech repairs,

utterance boundaries, utterance tags, and hierarchical discourse structure.

In this thesis, we have reported a series of empirical studies on human-human dialogues. We first examined human initiative behavior on decomposing a complex goal into sub-goals and then achieving each of them. We annotated the Trains corpus with who showed initiative on each utterance and who proposed a sub-goal. We found statistical evidences that when pursuing a sub-goal, 1) the conversant who proposes it shows initiative frequently and consecutively; and 2) the other conversant shows initiative occasionally and intermittently. These findings suggest that the conversant who proposes a sub-goal is directing the conversation towards achieving it (unless it is delegated). We then ran computer simulation experiments to show that following this model results in more efficient conversations.

We next examined human initiative behavior on switching the conversation to a more urgent task in multi-tasking dialogue. We first collected a corpus of multi-tasking dialogues (the MTD corpus) in which pairs of conversants perform overlapping verbal tasks, and then annotated where conversants switch from the on-going task to a momentary task, either at the end of a poker game (least abrupt), at the end of deciding which card to discard, or in the middle of discussing a card (most abrupt). We then measured the timing, pitch and volume associated with the task switches. We found statistical evidences that conversants raise pitch to signal the switch: the more abrupt the switch, the higher the pitch.

Initiative conflicts, where both conversants try to direct the conversation at the same time, naturally arises in mixed-initiative conversations. The last part of this thesis examined human behavior on initiative conflicts. We identified initiative conflicts as two overlapping utterances in which one is showing initiative and the other is abandoned or ignored. We measured the timing, duration, volume, and pitch that associate with initiative conflicts. First, we found that the offset of the utterances involved in initiative conflicts tend to be less than 300ms, and that initiative conflicts are more likely to occur when conversants have an urgent event. These findings suggest that initiative conflicts are unintentional collisions and that conversants try to avoid them unless for urgency reasons. Second, we found that the conflicts tend to last less than two syllables, and that volume

correlates with who wins the conflicts. Furthermore, in the longer conflicts, the winner tends to increase the volume during the conflicts. These findings suggest that initiative conflicts are efficiently resolved through a negotiation process, using volume as a device. Finally, we also found that after the resolution, conversants might repair the initiative conflicts to remedy the impaired conversation.

8.2 Application to Dialogue Systems

Dialogue systems have promising applications in mobility required, hands-busy or eyes-busy environments, and for blind people. The findings of this thesis on human-human dialogues have implications for building next-generation dialogue systems.

Our findings in managing complex user goals can serve as guidelines for implementing a dialogue manager to support mixed-initiative dialogue. For example, if the user proposes a sub-goal, or starts working on a sub-goal, the system might play a role as an assistant. However, if the system proposes a sub-goal, the system should try to play a more active role to achieve it. Nevertheless, the system can delegate the goal to the user by a request if the system is not capable of achieving it.

Our findings in interruption can serve as guidelines for implementing a dialogue manager to support multi-tasking dialogue. For example, for the recognition part, when detecting the discourse markers *oh* or *wait*, the system might use a more general grammar to accept possible utterances from a new task; for the synthesis part, the system might use higher pitch to alert the user when switching tasks.

Our findings in initiative conflicts can also serve as guidelines for implementing a dialogue manager. First, the system should try to avoid initiative conflicts, unless the system has something urgent to say. To achieve this, a dialogue manager will need to precisely evaluate the urgency of utterances, and be prepared to halt its production of an utterance, if the user starts speaking first. Second, the system should also vary the volume based on the importance of the utterance. If, however, initiative conflicts occur due to unintentional collision, the system can compare the volumes to decide whether to continue or back-off, and use volume as a device to negotiate initiative with the user. Finally, if

the system wins the initiative conflict, it should repair the speech immediately after the conflict if the conversation risks being impaired, especially for situations that involve short offset, long duration, and similar volumes.

It is also important to note that people use different prosodic cues to signal different phenomena. Pitch is used to signal the switch of task, while volume is used to signal the eagerness of showing initiative. When building a dialogue system, we have to take into account such subtle differences of what to signal and how to signal it.

Our future work is to implement these human conventions into a dialogue system to provide natural and efficient human-computer interactivity for complex goals and multiple tasks. Once such a system is built, we will run user study to evaluate the improvement.

References

- Allen, James F. 1999. Mixed-initiative interaction. *IEEE Intelligent Systems*, pages 14–16, September–October.
- Allen, James F. and Mark G. Core. 1997. DAMSL: Dialog annotation markup in several layers. Unpublished Manuscript.
- Ayers, Gayle M. 1992. Discourse functions of pitch range in spontaneous and read speech. Presented at the Linguistic Society of America Annual Meeting.
- Bangerter, Adrian and Herbert H. Clark. 2003. Navigating joint projects with dialogue. *Cognitive Science*, 27:195–229.
- Barras, C., E. Geoffrois, Z. Wu, and M. Liberman. 2001. Transcriber: Development and use of a tool for assisting speech corpora production. *Speech Communication*, 33(1-2):5–22.
- Beach, Cheryl M. 1991. The interpretation of prosodic patterns at points of syntactic structure ambiguity: Evidence for cue trading relations. *Journal of Memory and Language*, 30(6):644–663.
- Bernsen, Niels Ole, Laila Dybkær, and Mykola Kolodnytsky. 2003. An interface for annotating natural interactivity. In J. Van Kuppevelt and R. W. Smith, editors, *Current and New Directions in Discourse and Dialogue*. Kluwer Academic Publishers, chapter 3, pages 35–62.
- Bird, Steven, David Day, John Carofolo, John Henderson, Christophe Laprun, and Mark Liberman. 2000. ATLAS: A flexible and extensible architecture for linguistic annotation. In *Proceedings of 2nd LREC*, pages 1699–1706, Greece.
- Bird, Steven, Kazuaki Maeda, Xiaoyi Ma, and Haejoong Lee. 2001. Annotation tools based on the annotation graph API. In *Proceedings of ACL/EACL 2001 Workshop on Sharing Tools and Resources for Research and Education*, pages 31–34, France.
- Boersma, Paul and David Weenink. 2005. Praat: Doing phonetics by computer. Technical report, Institute of Phonetics Sciences of the University of Amsterdam, March. <http://www.praat.org>.
- Bohus, Dan, Sergio Grau Puerto, David Huggins-Daines, Venkatesh Keri, Gopala Krishna, Rohit Kumar, Antoine Raux, and Stefanie Tomko. 2007. ConQuest: an open-source dialog system for conferences. In *Proceedings of NAACL short paper session*, pages 9–12, Rochester NY.

- Butterworth, Brian. 1972. Hesitation and semantic planning in speech. *Journal of Psycholinguistic Research*, 4:75–87.
- Byron, Donna K. and P. Heeman. 1997. Discourse marker use in task-oriented spoken dialog. In *Proceedings of the 5th EUROSPEECH*, pages 2223–2226, Rhodes, Greece.
- Carberry, Sandra and Lynn Lambert. 1999. A process model for recognizing communicative acts and modeling negotiation subdialogues. *Computational Linguistics*, 25(1):1–53.
- Carletta, Jean. 1992. Planning to fail, not failing to plan: Risk-taking and recovery in task-oriented dialogue. In *Proceedings of 14th COLING*, pages 896–900, Nantes France.
- Carletta, Jean, Stefan Evert, Ulrich Heid, Jonathan Kilgour, Judy Robertson, and Holger Voormann. 2003. The NITE XML toolkit: Flexible annotation for multi-modal language data. *Behavior Research Methods, Instruments, and Computers*, pages 353–363. Special Issue on Measuring Behavior.
- Carletta, Jean, Amy Isard, Stephen Isard, Jacqueline C. Kowtko, Gwyneth Doherty-Sneddon, and Anne H. Anderson. 1997. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23(1):13–31.
- Cassidy, Steve and Jonathan Harrington. 2001. Multi-level annotation in the Emu speech database management system. *Speech Communication*, 33:61–77.
- Chu-Carroll, Jennifer and Michael K. Brown. 1998. An evidential model for tracking initiative in collaborative dialogue interactions. *User Modeling and User Adapted Interaction*, 8:215–253.
- Clark, Herbert H. and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognitive Science*, 22:1–39.
- Cohen, Philip R. and Hector J. Levesque. 1994. Preliminaries to a collaborative model of dialogue. *Speech Communication*, 15:265–274.
- Cohen, Philip R. and C. Raymond Perrault. 1979. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3:177–212.
- Cohen, Robin, C. Allaby, C. Cumbaa, M. Fitzgerald, K. Ho, B. Hui, C. Latulipe, F. Lu, N. Moussa, D. Pooley, A. Qian, and S. Siddiqi. 1998. What is initiative? *User Modeling and User Adapted Interaction*, 8:171–214.

- Core, Mark G. and James F. Allen. 1997. Coding dialogues with the DAMSL annotation scheme. In *Working Notes: AAAI Fall Symposium on Communicative Action in Humans and Machines*, pages 28–35, Cambridge.
- Crystal, Thomas H. and Arthur S. House. 1990. Articulation rate and the duration of syllables and stress groups in connected speech. *Journal of Acoustical Society of America*, 88:101–112.
- Duncan, Starkey Jr. 1972. Some signals and rules for taking speaking turns in conversations. *Journal of Personality and Social Psychology*, 23:283–292.
- Duncan, Starkey Jr. 1974. On the structure of speaker-auditor interaction during speaking turns. *Language in Society*, 2:161–180.
- Duncan, Starkey Jr. and George Niederehe. 1974. On signalling that it's your turn so speak. *Journal of Experiment Social Psychology*, 10:234–247.
- Flammia, Giovanni. 1998. *Discourse Segmentation of Spoken Dialogue: An Empirical Approach*. Ph.D. thesis, Massachusetts Institute of Technology.
- Galton, F. 1899. On instruments for (1) testing perception of differences of tint and for (2) determining reaction time. *Journal of the Anthropological Institute*, 19:27–29.
- Grosz, Barbara J. and Julia Hirschberg. 1992. Some intonational characteristics of discourse structure. In *Proceedings of 2nd ICSLP*, pages 429–432.
- Grosz, Barbara J., Aravind Joshi, and Scott Weinstein. 1995. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 2(21):203–225.
- Grosz, Barbara J. and Candace L. Sidner. 1986. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204.
- Guinn, Curry I. 1998. An analysis of initiative selection in collaborative task-oriented discourse. *User Modeling and User Adapted Interaction*, 8:255–314.
- Hagen, Eli. 1999. An approach to mixed initiative spoken information retrieval dialogue. *User Modeling and User Adapted Interaction*, 9:167–213.
- Haller, Susan and Timothy Fossum. 1999. Using protocols to model mixed initiative interaction. In *Proceedings of AAAI Workshop on Mixed Initiative Intelligence*.
- Hargrove, Robert. 1995. *Masterful Coaching*. John Wiley & Sons Canada Ltd., 1 edition.

- Harrison, P., S. Abney, D. Fleckenger, 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 ACL Workshop on Evaluating Natural Language Processing Systems*, pages 71–77, Berkely CA.
- Heeman, Peter A. and James F. Allen. 1995a. Dialogue transcription tools. Trains Technical Note 94-1, URCS, March. Revised.
- Heeman, Peter A. and James F. Allen. 1995b. The Trains spoken dialogue corpus. CD-ROM, Linguistics Data Consortium.
- Heeman, Peter A. and James F. Allen. 1999. Speech repairs, intonational phrases and discourse markers: Modeling speakers' utterances in spoken dialogue. *Computational Linguistics*, 25(4):527–571.
- Heeman, Peter A., Fan Yang, Andrew L. Kun, and Alexander Shyrokov. 2005. Conventions in human-human multithreaded dialogues: A preliminary study. In *Proceedings of IUI (short paper session)*, pages 293–295, San Diego CA.
- Heeman, Peter A., Fan Yang, and Susan E. Strayer. 2002. DialogueView: A dialogue annotation tool. In *Proceedings of 3rd SIGdial*, pages 50–59, Philadelphia PA.
- Heeman, Peter A., Fan Yang, and Susan E. Strayer. 2003. Control in task-oriented dialogues. In *Proceedings of 8th EUROSPEECH*, pages 209–212, September.
- Heritage, John. 1984. A change-of-state token and aspects of its sequential placement. In J. Maxwell Atldnson and John Heritage, editors, *Structures of social action: Studies in conversation analysis*. Cambridge University Press, chapter 13, pages 299–345.
- Hirschberg, Julia and Christine H. Nakatani. 1996. A prosodic analysis of discourse segments in direction-giving monologues. In *Proceedings of 34th ACL*, pages 286–293.
- Hobbs, Jerry R. 1985. On the coherence and structure of discourse. Technical Report CSLI-85-37, Center for the Study of Language and Information, Stanford University.
- Jefferson, Gail. 1973. A case of precision timing in ordinary conversation: Overlapped tag-positioned address terms in closing sequences. *Semiotica*, 9:47–96.
- Kamp, Hans and Uwe Reyle. 1993. *From Discourse to Logic*. Kluwer.
- Kipp, Michael. 2001. Anvil: A generic annotation tool for multimodal dialogue. In *Proceedings of 7th EUROSPEECH*, pages 1367–1370, Denmark.

- Kun, Andrew L., W. Thomas Miller, and William H. Lenharth. 2004. Computers in police cruisers. *IEEE Pervasive Computing*, 3(4):34–41, October-December.
- Lee, John D., Brent Caven, Steven Haake, and Timothy L. Brown. 2001. Speech-based interaction with in-vehicle computers: The effect of speech-based e-mail on drivers attention to the roadway. *Human Factors*, 43(4):631–640.
- Lemon, Oliver and Alexander Gruenstein. 2004. Multithreaded context for robust conversational interfaces: Context-sensitive speech recognition and interpretation of corrective fragments. *ACM Transactions on Computer-Human Interaction*, 11(3):241–267.
- Linell, Per. 1998. *Approaching Dialogue: Talk, Interaction and Contexts in Dialogical Perspective*. John Benjamins Publishing.
- Litman, Diane J. and James F. Allen. 1987. A plan recognition model for subdialogues in conversations. *Cognitive Science*, 11:163–200.
- Lochbaum, Karen E. 1998. A collaborative planning model of intentional structure. *Computational Linguistics*, 24(4):525–572, December.
- MacWhinney, Brian. 2000. *The CHILDES Project: Tools for Analyzing Talk*. Mahwah, NJ:Lawrence Erlbaum Associates, third edition.
- Mann, William C. and Sandra A. Thompson. 1987. Rhetorical structure theory: A theory of text organisation. In L. Polanyi, editor, *The Structure of Discourse*. Ablex Norwood, pages 1–81.
- McFarlane, Daniel C. 1999. Coordinating the interruption of people in human-computer interaction. In *Proceedings of INTERACT*, pages 295–303, The Netherlands.
- McKelvie, D., A. Isard, A. Mengel, M.B. Moeller, M. Grosse, and M. Klein. 2001. The MATE Workbench - An annotation tool for XML coded speech corpora. *Speech Communication*, 33(1-2):97–112. Special Issue: Speech Annotation and Corpus Tools.
- Moser, Megan and Johanna D. Moore. 1995. Investigating cue selection and placement in tutorial discourse. In *Proceedings of 33rd ACL*, pages 130–135.
- Müller, Christoph and Michael Strube. 2003. Multi-level annotation in MMAX. In *Proceedings of 4th SIGdial*, pages 198–207, Sapporo Japan.
- Nakajima, Shin'ya and James F. Allen. 1993. A study on prosody and discourse structure in cooperative dialogues. Technical report, Rochester, NY, USA. technical report.

- Passonneau, Rebecca J. and Diane J. Litman. 1997. Discourse segmentation by human and automated means. *Computational Linguistics*, 23(1):103–139.
- Pitrelli, John F., Mary E. Beckman, and Julia Hirschberg. 1994. Evaluation of prosodic transcription labeling reliability in the ToBI framework. In *Proceedings of 3rd ICSLP*, pages 123–126, Japan.
- Polanyi, Livia. 1988. A formal model of the structure of discourse. *Journal of Pragmatics*, 12:601–638.
- Power, Richard. 1979. The organization of purposeful dialogues. *Linguistics*, 17:107–152.
- Ramshaw, Lance A. 1991. A three-level model for plan exploration. In *Proceedings of 29th ACL*, pages 36–46, Berkeley CA.
- Reichman-Adar, Rachel. 1984. Extended person-machine interface. *Artificial Intelligence*, 22(2):157–218.
- Rose, Caolyn Penstein, Barbara Di Eugenio, Lori S. Levin, and Carol Van Ess-Dykema. 1995. Discourse processing of dialogues with multiple threads. In *Proceedings of 33rd ACL*, pages 31–38.
- Sacks, Harvey. 2004. An initial characterization of the organization of speaker turn-taking in conversation. In Gene H. Lerner, editor, *Conversation Analysis: Studies From The First Generation*. John Benjamins Pub Co, pages 35–42.
- Sacks, Harvey, Emanuel A. Schegloff, and Gail Jefferson. 1974. A simplest systematics for the organization of turn-taking for conversation. *Language*, 50(4):696–735.
- Schegloff, Emanuel A. 1987. Recycled turn beginnings: a precise repair mechanism in conversation's turn-taking organization. In G. Button and J. R.E. Lee, editors, *Talk and Social Organization*. Clevedon, England: Multilingual Matters Ltd., chapter 3, pages 70–85.
- Schegloff, Emanuel A. 2000. Overlapping talk and the organization of turn-taking for conversation. *Language in Society*, 29:1–63.
- Schiffrin, Deborah. 1987. *Discourse Markers*. Cambridge University Press.
- Schober, Michael F. and Herbert H. Clark. 1989. Understanding by addressees and overhearers. *Cognitive Science*, 21:211–232.
- Searle, J. R. 1969. *Speech acts: An essay in the philosophy of language*. Cambridge University Press, Cambridge.

- Shyrovkov, Alexander, Andrew Kun, and Peter Heeman. 2007. Experiments modeling of human-human multi-threaded dialogues in the presence of a manual-visual task. In *Proceedings of 8th SIGdial*, pages 190–193.
- Sidner, Candace L. 1994. An artificial discourse language for collaborative negotiation. In *Proceedings of the 12th AAAI*, pages 814–819.
- Sjölander, Kåre and Jonas Beskow. 2000. WaveSurfer: An open source speech tool. In *Proceedings of 6th ICSLP*, pages 4:464–467, Beijing China.
- Smith, Ronnie W. 1993. Effective spoken natural language dialogue requires variable initiative behavior: an empirical study. In *AAAI93 Fall Symposium On Human-Computer Collaboration*, pages 101–106.
- Stent, Amanda. 2000. Rhetorical structure in dialog. In *Proceedings of the 2nd International Natural Language Generation Conference (Student Paper)*.
- Strayer, Susan E., Peter A. Heeman, and Fan Yang. 2003. Reconciling control and discourse structure. In J. Van Kuppevelt and R. W. Smith, editors, *Current and New Directions in Discourse and Dialogue*. Kluwer Academic Publishers, chapter 14, pages 305–323.
- Sussman, E., I. Winkler, and E. Schrgöer. 2003. Top-down control over involuntary attention switching in the auditory modality. *Psychonomic Bulletin & Review*, 10(3):630–637.
- Sutton, Stephen, Ronald Cole, Jacques de Villiers, Johan Schalkwyk, Pieter Vermeulen, Mike Macon, Yonghong Yan, Ed Kaiser, Brian Rundle, Khaldoun Shobaki, Paul Hosom, Alex Kain, Johan Wouters, Dominic Massaro, and Michael Cohen. 1998. Universal speech tools: The CSLU toolkit. In *Proceedings of 5th ICSLP*, pages 3221–3224, Australia.
- Swerts, Marc. 1995. Combining statistical and phonetic analyses of spontaneous discourse segmentation. In *Proceedings of the 12th ICPHS*, volume 4, pages 208–211.
- Swerts, Marc and Mari Ostendorf. 1995. Discourse prosody in human-machine interactions. In *Proceedings of ESCA workshop on spoken dialogue systems: theories and applications*, pages 205–208, Visgo Denmark.
- Toh, Siew Leng, Fan Yang, and Peter A. Heeman. 2006. An annotation scheme for agreement analysis. In *Proceedings of 9th ICSLP*, pages 201–204, Pittsburgh PA.

- Traum, David R. and James F. Allen. 1994. Discourse obligations in dialogue processing. In *Proceedings of 32th ACL*, pages 1–8.
- Traum, David R. and Elizabeth A. Hinkelman. 1992. Conversation acts in task-oriented spoken dialogue. *Computational Intelligence*, 8(3):575–599. Special Issue: Computational Approaches to Non-Literal Language.
- Traum, David R. and Christine H. 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 workshop Towards Standards and Tools for Discourse Tagging*, pages 101–108, College Park MD.
- Walker, Marilyn and Steve Whittaker. 1990. Mixed initiative in dialogue: An investigation into discourse segmentation. In *Proceedings of 28th ACL*, pages 70–76.
- Walker, Marilyn A. 1996. The effect of resource limits and task complexity on collaborative planning in dialogue. *Artificial Intelligence Journal*, 85:181 – 243.
- Walker, Marilyn A., Donald Hindle, Jeanne Frome, Giuseppe Di Fabbrizio, and Craig Mestel. 1997a. Evaluating competing agent strategies for a voice email agent. In *Proceedings of 5th EUROSPEECH*, pages 2219–2222, Rhodes Greece.
- Walker, Marilyn A., Diane J. Litman, Candace A. Kamm, and Alicia Abella. 1997b. PARADISE: A framework for evaluating spoken dialogue agents. In *Proceedings of 35th ACL*, pages 271–280, Madrid, Spain, July.
- Whittaker, Steve and Phil Stenton. 1988. Cues and control in expert-client dialogue. In *Proceedings of 28th ACL*, pages 123–130.
- Yang, Fan and Peter A. Heeman. 2004. Using computer simulation to compare two models of mixed-initiative. In *Proceedings of 8th ICSLP*, pages 213–216.
- Yang, Fan and Peter A. Heeman. 2007a. Avoiding and resolving initiative conflicts. In *Proceedings of NAACL/HLT*, pages 268–275.
- Yang, Fan and Peter A. Heeman. 2007b. Exploring initiative strategies using computer simulation. In *Proceedings of INTERSPEECH*, pages 106–109.
- Yang, Fan, Peter A. Heeman, and Kristy Hollingshead. 2004. Towards understanding mixed-initiative in task-oriented dialogues. In *Proceedings of 8th ICSLP*, pages 217–220.

- Yang, Fan, Peter A. Heeman, Kristy Hollingshead, and Susan E. Strayer. 2008. Dialogue-view: Annotating dialogues in multiple views with abstraction. *Natural Language Engineering*, 14(1):3–32.
- Yang, Fan, Peter A. Heeman, and Susan E. Strayer. 2003. Acoustically verifying speech repair annotation. In *Proceedings of DISS'03*, pages 95–98, Sweden.
- Yang, Fan, Peter A. Heeman, and Susan E. Strayer. 2004. Evaluation of “clean display” in DialogueView. Technical Report CSLU-04-001, Center for Spoken Language Understanding, OGI School of Science & Engineering, June.
- Yang, Fan, Susan E. Strayer, and Peter A. Heeman. 2002. ACT: A graphical dialogue annotation comparison tool. In *Proceedings of 7th ICSLP*, pages 1553–1556, Denver CO.

Biographical Note

Fan Yang was born in Shantou Guangdong China on Nov 25, 1976. He went to Subei High School, where he ranked fifth in Guangdong province on the national university entrance exam. He then went to Sun Yat-sen University for his undergraduate study on computer science. After graduation with a bachelor degree of science, being a top 2% students, Fan was recommended for admission to the Chinese Academy of Sciences (CAS) for advanced study, where he started his interest in artificial intelligence and computational linguistics. Under the guidance of Prof. Ruqian Lu, Fan explored a wide spectrum of natural language processing, from NLU, NLG, dialogue management, to knowledge representation and reasoning. Fan decided that he was most interested in dialogue processing. Thus with a master degree of engineering from CAS, Fan went to the country on the other side of the Pacific Ocean, U.S.A., to pursue his PhD in Oregon Health & Science university. Of course Fan encountered cultural shock and language problems. It was lucky that Fan's advisor, Peter A. Heeman, gave him a lot of patience and encouragement; Fan learned a lot, such as running scientific experiments and writing good papers, from Peter.

It was fate that in the year of 2006, Fan met the beautiful lady, Huana Zheng. They got married at Las Vegas, NV on Jun 6th, 2007, and live a happy life ever since.

List of publications:

1. **Fan Yang**, Peter A. Heeman, Kristy Hollingshead, and Susan E. Strayer. 2008. Dialogueview: Annotating dialogues in multiple views with abstraction. *Natural Language Engineering*, 14(1):3–32.
2. **Fan Yang**, Gokhan Tur, Elizabeth Shriber. 2008. Exploiting Dialogue Act Tagging and Prosodic Information for Action Item Identification. In *Proceedings of ICASSP*. To appear.

3. **Fan Yang** and Peter A. Heeman. 2007. Avoiding and resolving initiative conflicts. In *Proceedings of NAACL/HLT*, pages 268–275.
4. **Fan Yang** and Peter A. Heeman. 2007. Exploring initiative strategies using computer simulation. In *Proceedings of INTERSPEECH*, pages 106–109.
5. **Fan Yang**, Junlan Feng, and Giuseppe Di Fabbrizio. 2006. A Data Driven Approach to Relevancy Recognition for Contextual Question Answering. In *Proceedings of NAACL workshop on Interactive Question Answering*, New York NY.
6. Toh, Siew Leng, **Fan Yang**, and Peter A. Heeman. 2006. An annotation scheme for agreement analysis. In *Proceedings of INTERSPEECH*, pages 201–204, Pittsburgh PA.
7. Heeman, Peter A., **Fan Yang**, Andrew L. Kun, and Alexander Shyrokov. 2005. Conventions in human-human multithreaded dialogues: A preliminary study. In *Proceedings of Intelligent User Interface (short paper session)*, pages 293–295, San Diego CA.
8. **Fan Yang** and Peter A. Heeman. 2004. Using computer simulation to compare two models of mixed-initiative. In *Proceedings of 8th ICSLP*, pages 213–216.
9. **Fan Yang**, Peter A. Heeman, and Kristy Hollingshead. 2004. Towards understanding mixed-initiative in task-oriented dialogues. In *Proceedings of 8th ICSLP*, pages 217–220.
10. Heeman, Peter A., **Fan Yang**, and Susan E. Strayer. 2003. Control in task-oriented dialogues. In *Proceedings of 8th eurospeech*, pages 209–212, September.
11. **Fan Yang**, Peter A. Heeman, and Susan E. Strayer. 2003. Acoustically verifying speech repair annotation. In *Proceedings of DISS'03*, pages 95–98, Sweden.
12. Strayer, Susan E., Peter A. Heeman, and **Fan Yang**. 2003. Reconciling control and discourse structure. In J. Van Kuppevelt and R. W. Smith, editors, *Current and New Directions in Discourse and Dialogue*. Kluwer Academic Publishers, chapter 14, pages 305–323.

13. **Fan Yang**, Susan E. Strayer, and Peter A. Heeman. 2002. ACT: A graphical dialogue annotation comparison tool. In *Proceedings of 7th ICSLP*, pages 1553–1556, Denver CO.
14. Heeman, Peter A., **Fan Yang**, and Susan E. Strayer. 2002. DialogueView: A dialogue annotation tool. In *Proceedings of 3rd SIGdial workshop on Dialogue and Discourse*, pages 50–59, Philadelphia PA.
15. Lu, Ruqian, Songmao Zhang, Lengning Liu, **Fan Yang**, etc. 2002. Children Turing Test. In *Proceedings of 2nd China-Japan Natural Language Processing*, Peking China.
16. **Fan Yang**. 2001. A Model of Fuzzy Commonsense Knowledge Base. *Computer Engineering and Applications (CE&A, in Chinese)*, 37(21): 106-108 (2001).
17. **Fan Yang** 2000. Commonsense about Emotion based Anaphora Analysis. In *proceedings of the International Symposium on Future Software Technology*, pages 347-349, Guiyang China.
18. **Fan Yang** and Ruqian Lu. 2000. The Combination of CBR and Commonsense to Machine Translation. In *proceedings of Conference on Intelligent Information Proccession (IFIP Congress)*, pages 268-271, Peking China.