A Computational Theory of Grounding in Natural Language Conversation

by

David R. Traum

Submitted in Partial Fulfillment

of the

Requirements for the Degree

Doctor of Philosophy

Supervised by

Professor James F. Allen

Department of Computer Science
College of Arts and Science

University of Rochester
Rochester, New York

1994
Curriculum Vitae

David Rood Traum was born August 3, 1963 in New York City. Shortly thereafter, he began to use and understand natural language, little realizing that decades later he would be spending so much of his time analyzing how it works. He moved with his family to South Florida in 1968, and remained until leaving for Cambridge, Massachusetts to attend Harvard College in 1981. He graduated with a Bachelor of Arts in Applied Mathematics in 1985.

After short stays in Europe, back in Cambridge, and Los Angeles, he decided to pursue further study and went to San Jose State University, receiving a Master of Science in Computer Science – Mathematics in December, 1987. While there, he developed an interest in research while working on problems in graph theory with Sin-Min Lee, and an interest in Computational Linguistics from a course and discussions with Sam Mchombo. These interests were then fused after attending the LSA Linguistic Institute at Stanford University in the summer of 1987.

He enrolled in the PhD program at the University of Rochester in 1988, receiving along the way, through no fault of his own, a second Master’s degree in 1990. While at Rochester, he was a teaching assistant for courses in data structures and AI programming, and a research assistant for James Allen. He now hopes that this document will allow him to complete yet another degree and move on once again.
Acknowledgments

After spending so much time in graduate school, trying to thank all of the people who have helped along the way seems almost as daunting a task as writing the dissertation itself. First, and foremost, I'd like to thank my advisor James Allen. He first introduced to me action-based approaches towards language and has been a constant collaborator in all of my work since. He has often helped me find the way through the difficult times, and his continual push for clarity and simplicity has allowed me to get much further than it often seemed I could.

I also want to thank the other members of my thesis committee. Len Schubert's clear insight and perspective were of great help in assessing the merits of both my own work and that of others. He also taught me a lot about semantic approaches to language and why we should care about them, even though the pragmatics is what we're "really" after. Mike Tatenhaus provided perspective from another branch of cognitive science, helping me understand some of the psycholinguistic approaches to discourse processing. Graeme Hirst shared an interest in trying to apply the insights from Conversation Analysis within a computational perspective and was very helpful in suggesting relevant work I hadn't yet come across. Graeme has also taken pains from soon after meeting me to introduce me to "kindred spirits" in the research community (including some of his own students), from which I've received great benefit.

My collaborators also deserve much credit. Alice Kyburg, Elizabeth Hinkelman, Keri Jackson, Lou Hoebel, Stephane Guez, and James were a great group to work with on a language system before TRAINS. Nat Martin and Stephen Feist built the Armtrak simulator that allowed me to explore ideas about action in a multi-agent environment in a concrete fashion. Frustrations in trying to use existing speech act taxonomies to label the dialogues Shin'ya Nakajima had collected and was studying the prosodic features of led to the formulation of the conversation act theory presented here. Derek Gross and Massimo Poesio also made valuable contributions to the early development. Elizabeth suggested that this would be a good framework within which to frame arguments about non-literal speech acts. Text we subsequently passed back and forth between Rochester and first Chicago and later Saarbrücken ended up here as Chapter 4. Working on the TRAINS system has been extremely valuable along a number of dimensions. James and Len deserve a large amount of credit for getting so many strong-willed, independent, and creative people to work so well together so productively. Interacting so closely with George Ferguson, Peter Heeman, and Massimo helped refine lots of ideas on the specifics of interaction between the implementation described in Chapter 6 and adjacent parts of the system. Chung Hee Hwang, Graham Katz, Janet Hitzeman, Marc Light, Nat,
and Tsuneaki Kato also made important contributions to the functioning of the whole system.

I would also like to thank many colleagues for stimulating conversations and comments on previous expositions of my work. It is impossible to note everyone who deserves mention here, but Susan Brennan, Phil Cohen, Julia Hirschberg, Ed Hovy, Karen Lochbaum, Susan McRoy, and Candy Sidner have been particularly helpful to me in sharpening some of the ideas presented here.

The Computer Science department at Rochester has been a great place to work, which I shall surely miss. The faculty and students work together in a supportive, and productive manner, but are always ready to go off and engage in needed diversion. All of the staff has been responsive and helpful. Many thanks to Peggy Frantz, Jill Forster, Peggy Meeker, Pat Marshall, Luidy Bukys, Jim Roche, and Tim Becker. Special thanks to Brad Miller who responded extremely promptly to sometimes inadequate bug reports and, in order to accommodate my needs, made many extensions to RHET beyond what it was ever intended to do.

The Cognitive Science group at Rochester has been equally beneficial, and I greatly value the interactions I’ve had with this group. Tom Bever deserves special commendations for inviting great visiting speakers over to his house for “students only” dinners. This allowed not only getting to know and talk to prominent scientists in a more informal setting, but built a great community among us students. I also want to mention Joel Lachter as an important influence from the beginning who was always willing to talk about the larger issues, as well as run a great bridge game.

This material is based upon work supported by ONR/DARPA under grant number N00014-92-J-1512, by ONR under research grant number N00014-90-J-1811, and by NSF under grant number IRI-9003841.

Finally, I’d like to thank my family for not forgetting about me during the busy times, and for accepting the “wrong kind” of doctor.
Abstract

The process of adding to the common ground between conversational participants (called grounding) has previously been either oversimplified or studied in an off-line manner. This dissertation presents a computational theory, in which a protocol is presented which can be used to determine, for any given state of the conversation, whether material has been grounded or what it would take to ground the material. This protocol is related to the mental states of participating agents, showing the motivations for performing particular grounding acts and what their effects will be.

We extend speech act theory to account for levels of action both above and below the sentence level, including the level of grounding acts described above. Traditional illocutionary acts are now seen to be multi-agent acts which must be grounded to have their usual effects.

A conversational agent model is provided, showing how grounding fits in naturally with the other functions that an agent must perform in engaging in conversation. These ideas are implemented within the TRAINS conversation system.

Also presented is a situation-theoretic model of plan execution relations, giving definitions of what it means for an action to begin, continue, complete, or repair the execution of a plan. This framework is then used to provide precise definitions of the grounding acts in terms of agents executing a general communication plan in which one agent must present the content and another acknowledge it.
# Table of Contents

**Curriculum Vitae**

**Acknowledgments**

**Abstract**

**List of Tables**

**List of Figures**

## 1 Introduction

1.1 The Grounding Problem .................................. 3
1.2 Thesis Statement ........................................ 4
1.3 Grounding in Human and Computer Communication .......... 4
1.4 Contributions of the Dissertation .......................... 6
1.5 Outline of Dissertation ................................. 6

## 2 Foundational Work

2.1 Speech Acts and Plans .................................... 9
2.2 Models of Mental States .................................. 15
2.3 Conversation Analysis .................................... 22
2.4 Previous attempts to incorporate CA in NLP systems ...... 24
2.5 Grounding in Conversation and the Contribution Model .... 26
2.6 Deficiencies of the Contribution Model .................... 29

## 3 Towards a Computational Theory of Grounding

3.1 Tracking Grounding with Discourse Units ................. 31
3.2 A Finite-State Theory of Grounding ....................... 36
3.3 A Cognitive Model of Grounding Act Processing .......... 43
3.4 Connecting the Cognitive and FA Sequence Models ......... 47
3.5 Deficiencies of the Finite Automaton Grounding Model .... 49
3.6 Chapter Summary ........................................ 53
4 **An Overview of Conversation Acts**  
4.1 The Need for Multiple Levels of Action in Conversation .......... 55  
4.2 Levels of Conversation Acts .................................... 57  
4.3 Conversation Acts in an Example Conversation ..................... 61  
4.4 Recognizing Conversation Acts .................................. 73  
4.5 Related Classification Schemes ................................. 76  

5 **Dialogue Management**  
5.1 Mental and Conversational State .................................... 79  
5.2 Updating the Conversational State .................................. 86  
5.3 The Discourse Actor .................................................. 87  
5.4 Capabilities of the Dialogue Management Model ..................... 92  

6 **Implementation**  
6.1 TRAINS System Overview ........................................... 93  
6.2 Knowledge Representation ........................................... 95  
6.3 Other Aspects of the Discourse Context ............................ 102  
6.4 Conversation Act Implementation .................................. 105  
6.5 Discourse Actor Actions ............................................. 117  
6.6 Annotated Trace of a Sample Conversation .......................... 118  
6.7 Discussion ............................................................. 134  

7 **Towards a Formal Theory of Plan Execution**  
7.1 Requirements for a Theory of Plan Execution ....................... 137  
7.2 A Sketch of Rational, Plan-based Behavior .......................... 138  
7.3 Single-Agent Plan Execution ....................................... 139  
7.4 Sub-Plans and Sub-Plan Executions ................................ 147  
7.5 Multi-Agent Plan Execution ....................................... 149  
7.6 Communication Recipes and Grounding Acts ........................ 152  

8 **Conclusions and Future Research**  
8.1 Summary ............................................................. 163  
8.2 Future Work ......................................................... 163  

Bibliography 167
# List of Tables

2.1 Clark & Marshall’s Methods of Achieving Copresence for Mutual Knowledge 
2.2 Adjacency Pairs
2.3 [Clark and Schaefer, 1989, p. 267]: Types of Evidence of Understanding
3.1 DU Transition Diagram
3.2 Summary of Discourse Unit States
3.3 X’s actions
3.4 Y’s actions
3.5 Constraints on Grounding Acts
4.1 Conversation Act Types
4.2 DU Acts from Conversation
5.1 Actions to Perform Based on Ungrounded Material
5.2 Actions to Perform Based on Discourse Goals
6.1 TRAINS-93 Obligation Types and Sources
6.2 Surface Speech Act Rules
6.3 Turn-taking Act Effects
# List of Figures

3.1 Transition Network for Contributions ........................................ 32  
3.2 Initial Transition Network for DUs ........................................... 33  
3.3 Transition Network for DUs with self-initiated self-repair and cancels . 34  
3.4 Transition Network for DUs with recursive repairs ...................... 34  
3.5 Transition Network for REPAIR Sub-DU .................................... 35  
3.6 Transition Network for REQ-REPAIR Sub-DU .............................. 35  
3.7 Modified Transition Network for DUs with recursive repairs .......... 36  
3.8 Finite-State Network with single level of repairs ....................... 37  
3.9 Transition Network to and from State 1 ................................. 38  
3.10 Transition Network to and from State 2 .................................. 39  
3.11 Transition Network to and from State 3 .................................. 39  
3.12 Transition Network to and from State 4 .................................. 40  
3.13 Transition Network to and from State F .................................. 40  
3.14 Architecture of X's Model of Conversation with Y ..................... 44  

4.1 Trains World Set-up for Example Conversation .......................... 61  
4.2 TRAINS Domain Conversation with Intonational Features – First Part 63  
4.3 TRAINS Domain Conversation with Intonational Features – Second Part 64  
4.4 Conversation with Grounding Acts .......................................... 66  
4.5 Top-Level Trains Conversation Plan ........................................ 69  
4.6 Domain Plan for Moving Oranges to Bath ................................ 70  
4.7 Domain Plan for Moving Oranges to Bath ................................ 72  
4.8 Trains World Set-up for Example Conversation .......................... 72  

5.1 Abstract Architecture of Conversation System ........................... 80  
5.2 Belief and Proposal Contexts ............................................... 81  
5.3 Discourse Script for TRAINS Conversations ............................ 83
<table>
<thead>
<tr>
<th>Discourse Context</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>after utterance 8</td>
<td>183</td>
</tr>
<tr>
<td>before utterance 14</td>
<td>185</td>
</tr>
<tr>
<td>after utterance 14</td>
<td>186</td>
</tr>
<tr>
<td>after utterance 15-2=4</td>
<td>188</td>
</tr>
<tr>
<td>after utterance 15-5=7</td>
<td>189</td>
</tr>
<tr>
<td>after utterance 15-8=10</td>
<td>191</td>
</tr>
<tr>
<td>deciding to acknowledge DU-6</td>
<td>191</td>
</tr>
<tr>
<td>before producing utterance 16</td>
<td>191</td>
</tr>
<tr>
<td>after interpreting utterance 16</td>
<td>192</td>
</tr>
<tr>
<td>after interpreting utterance 17</td>
<td>193</td>
</tr>
<tr>
<td>before producing utterance 18-3</td>
<td>193</td>
</tr>
<tr>
<td>after interpreting utterance 18-3</td>
<td>194</td>
</tr>
<tr>
<td>after interpreting utterance 19</td>
<td>195</td>
</tr>
</tbody>
</table>
1 Introduction

Clark [1992] discusses the difference between what he calls the product view of language, encompassing studies of linguistic competence, language structure, and semantics, with the action view of language, encompassing studies of linguistic performance, language use, and pragmatics. He gives three tenets for the study of language use:

- In language use, utterances are more basic than sentences.
- In language use, speaker’s meaning is primary, and word or sentence meaning are derivative.
- Speaking and listening aren’t autonomous activities, but parts of collective activities.

The action view begins with Austin’s observation that utterances in conversation are speech acts, and as such should be treated as part of a theory of action [Austin, 1962]. This observation and the subsequent research program on speech acts within the philosophy of language has been followed up on by researchers in AI, treating speech acts in the same manner as other actions that an agent can perform, recognize another agent performing, and reason about. The traditional approach has been to view speech acts as changes to the cognitive state of another agent, analogous to the way physical actions change the state of the physical world. Speech act operators have been devised using the formalisms from AI planning systems, so that deciding what to say can be seen as utterance planning, and interpretation of the intention behind an utterance can be viewed as plan recognition.

Given that utterances are actions, what kinds of actions are they? What are the enablement and generation conditions and what are the effects? One difficulty with formalizing speech acts in this way is that, as Clark’s third tenet claims, all speech acts are collaborative acts: they require both a speaker to make an utterance and a listener to understand and accept it in some way. The result of a speech act is thus, in some ways, negotiated by the conversational participants. A successful formulation of speech acts will have to be based on a theory of multi-agent action.

The collaboration process is further complicated by the fact that participants cannot infallibly recognize the mental states of the other participants: the hearer of an utterance cannot know for sure that he has understood the intent of the speaker (speaker meaning),
and knowing this fact about the hearer, the speaker cannot be sure that she has been understood. In general, many of the effects of speech actions will be to the cognitive states of conversational participants. Because each agent will not have direct access to the cognitive states of other agents, effects of speech acts and the intentions behind them are intrinsically uncertain.

Most previous natural language understanding systems have largely ignored the problem of coordinating understanding and have assumed that the intent of the utterance can be recognized merely by observing the occurrence of the utterance, using only the form of the utterance itself plus background context including knowledge of the other participant. Since most of these systems have been built as part of a question-answering system, with system responses resulting from a single database retrieval and in which complicated discourse interactions aren’t possible, or as part of a story understanding system, in which there is no facility for interaction and off-line processing can be performed at leisure.

In contrast, the study of conversation shows that there is quite a rich system for coordinating understanding. For example, studies of conversations in the TRAINS domain\(^1\) show that about half of the utterances in a conversation are related to coordinating understanding rather than being domain-level utterances on the topic of the conversation [Allen and Schubert, 1991]. There is an acknowledgement system to make sure that utterances are heard and understood. There is an acceptance system so that proposals and information can be agreed on. There are facilities for clarification and repair of potential or actual misunderstandings. These facilities have been the object of extensive study in the field of Conversation Analysis, but have only recently been adopted in AI conversation systems (see Section 2.4).

A difficulty in formulating descriptions of actions (such as speech acts) in a multi-agent setting is determining what point of view is being described. Three common points of view are:

1. The objective (what “really” is) (sentence meaning)
2. The point of view of the performing agent (speaker meaning)
3. The point of view of an observing agent

Ideally, one would like all three to coincide, i.e., the actor decides what she wants to do, performs an action which accomplishes this intention, and the intention and action are correctly recognized by the observer. Unfortunately, as mentioned above, this kind of situation is not guaranteed. The actor may have incorrect beliefs, or may fail in her action, so that what she believes she did is not what she “really” did. The observer also has limited knowledge, and may misinterpret an action. He also has no access to and only limited evidence about the mental state of the actor, and may not recognize what is intended.

For linguistic communication, what “really” happened is of less importance than that the conversing agents reach some sort of understanding. The question of whether

---

\(^1\)Described below in Section 4.3.
the meaning of an utterance has some objective status aside from what is intended and recognized by the agents is controversial and not really relevant here. All that is important for communication is that one agent used a particular locution to convey some content to another agent, and that the speaker's intention to convey the content becomes mutually understood (grounded) by both agents, regardless of any objective meaning of the utterance.

These distinctions have not generally been made in most speech act work, so it is often difficult to tell the ontological status of many proposed acts: are they objective phenomena, observable and testable by an (ideal) observer? Are they part of the mental states of the agents, consciously used and necessary for getting at what was intended? Is a set of speech acts to be interpreted as an objective description of the conversation process, or a psychological model of communicating agents (or both)?

A complete theory of action in conversation is a project far too ambitious to be undertaken in one dissertation. Instead, I pick out one important subproblem, the grounding problem, and show how an action-oriented approach can shed light on its difficulties. I approach this problem from several different angles: how best to characterize particular utterance types as actions, how to use this theory of action within a computer system which converses in natural language, and how to formally characterize conversational actions within a more general logic of action. For each of these problems, I present the skeleton of a more general theory, and show in detail how the grounding problem can be approached within this framework.

1.1 The Grounding Problem

It is uncontroversial that successful communication requires some degree of common ground between the communicators. In order to communicate in a natural language, for instance, the communicators must have some agreement on the meanings of words, the legal combinations of words, and the meanings of these combinations. Not only must a speaker know these, she must assume that her hearer knows them and knows that she knows them, etc. This type of common ground is often called mutual knowledge, mutual belief, shared information, and other similar terms. In addition to this basic type of common ground, researchers have used concepts like mutual belief as components in theories of a wide range of social and linguistic phenomena, including linguistic meaning [Schiffer, 1972], definite reference [Clark and Marshall, 1981], conventions [Lewis, 1969], social norms [Bach and Harnish, 1979], and coordinated activity (many authors including [Levesque et al., 1990]).

An important question is how this common ground is augmented by engaging in conversation. This process is termed Grounding by Clark and Schaefer [1989]. The famous Byzantine generals problem illustrates that actual mutual knowledge cannot be achieved in a situation in which communication is fallible [Halpern and Moses, 1990]. Yet, for the most part, conversants seem to get by well enough. Clark and Schaefer suggest that such mutuality need not be absolute, but merely sufficient for the current purposes.
1.2 Thesis Statement

We provide a computational model of how conversants reach a state of mutual understanding of what was intended by the speaker of an utterance. Most previous computational NL systems and AI theories have ignored the problem, assuming that this happens more or less automatically as a result of the speaker and hearer being together, and instead have concentrated on the problem of correctly interpreting the meaning of the utterance. Instead we say that it is less important to come up with the "correct" interpretation than to get an approximately correct interpretation, and then use repairs to patch things when the interpretation is not close enough for current purposes (the grounding criterion of Clark and Schaefer [1989]). This approach seems increasingly necessary as researchers are finding many important problems in natural language processing and planning which are intractable for the optimal case [Perrault, 1984; Chapman, 1987; Reiter, 1990]. Several researchers in other fields have come up with similar schemes for presenting a post-hoc analysis of grounding in conversation, but have not presented formal models which would show how an agent could achieve common ground on-line.

1.3 Grounding in Human and Computer Communication

The grounding problem occurs not only for natural language communication, but any time two agents need to coordinate on transfer of information. A simple example is one of the computer communication protocols. These protocols not only allow transmission of information (in data packets called information frames, or I-frames, from a sender to a receiver), but allow the receiver to send back acknowledgements and negative acknowledgements, as well as allowing the sender to repair transmission errors. Most protocols use a single acknowledgement scheme in which the sender assumes the data was correctly received if an acknowledgement is received and the receiver assumes the sender believes the data was correctly received when the receiver sends the acknowledgement. If the sender does not receive an acknowledgement, then it resends the packet. Likewise, if the sender gets a negative acknowledgement (signaling a corrupted or missing packet) it can resend the packet.

We can say that the communicating systems have grounded a particular I-frame when an acknowledgement for it is received (the receiver of the original I-frame assumes the packet is grounded unless/until it gets a retransmission). There are many similarities between these protocols and grounding in natural language conversation. For instance, both incorporate different styles of acknowledgement: in computer communication, one style of acknowledgement, called echo-checking, is to have the responder repeat back the original message. This also happens frequently in human conversation, as in the exchange in (1).

(1)  S: It would get there at 4.  
     M: It would get there at 4.
For both types of communication, this is a fairly expensive style of acknowledgement and is used primarily in cases where it is very important to demonstrate that the receiver got the message exactly right and when there is plenty of time. Another style of acknowledgement is to send a separate message type, an Ack message. In English, the following utterances typically have the meaning that the previous utterances were understood: “uh-huh”, “yeah”, “right”, “okay.”

There is also a third type of acknowledgement common to both media. In computer communication in which information is being transmitted both ways, some protocols allow an alternative to sending a separate Ack message. This is to piggyback the acknowledgement onto the next I-frame sent by the receiver of the initial I-frame. There is a similar strategy in natural language conversation, in which a message will implicitly acknowledge a previous message, when the acknowledging message would not be sensible without the initial message. A simple example of this type of phenomena is an answer to a question, as in (2).

(2) M: How long does it take to load oranges?
   S: An hour.

Another example would be a follow-up response such as (3).

(3) M: Let’s ship the oranges.
   S: And we’ll have E3 pick them up.

NL conversation also has a rich supply of forms for negative acknowledgement, forms which specify lack of receipt of a clear message, such as, in English, “What?” “Pardon?” and “Please repeat.”

There are, however a number of differences between NL conversation and computer communication. First of all, in NL conversation, rather than having a fixed “frame” size, utterances can be of varying lengths, with wide-ranging information contents. Also, unlike data communications where I-frames have ID numbers, in NL conversation there is generally no way of identifying a particular utterance, or of linking a general acknowledgement to a specific utterance.

Furthermore, grounding in NL conversation generally occurs at the level of meaning level, rather than form. In most cases, it is perfectly reasonable to acknowledge an utterance in which the gist of the utterance is understood, even though the precise form was not heard correctly. Also, if the form was heard correctly, but is not comprehensible in the utterance situation, this is a situation for repair. In contrast, most data communication is acknowledged if and only if the actual form was received correctly, regardless of whether the receiver can fathom the meaning of what was sent.

Finally, natural language has a much more flexible system of repair. Rather than just repeating a previous utterance, a speaker can repair only the problematic parts of an utterance, as in (4), In which the speaker changes the question from one of tanker contents to tanker location.

(4) S: Do you have to have the tanker full?
   : uh there?
Also, in addition to asking for a retransmission or not responding, the listener of an utterance can also repair or request repair of a specific part of the utterance, as in (5).

(5) \[ \text{S: and then which route do you wanna take?} \]
    \[ \text{M: took to Avon?} \]
    \[ \text{S: yeah} \]

These increases in flexibility come at a cost of precision. It is thus harder to determine with a given level of certainty that a particular utterance was understood correctly. Neutral acknowledgements like "uh-huh" signal, at best, a claim that the previous utterances were understood, without being able to specify that, say, utterances 1, 2, and 4 were understood while 3 was never received.

The challenge, then could be phrased as follows: is there a protocol in natural language conversation for grounding? What is the form of this protocol, and how can it be used by a computer system to allow it to engage in natural conversation? What are the message types, how are they recognized, and how do transmission and receipt of these messages affect the states of the participating agents?

1.4 Contributions of the Dissertation

This dissertation meets the challenge from several directions. First, a set of abstract actions for grounding is induced from study of task-oriented dialogues. These actions are somewhat orthogonal to traditional speech acts concerned with illocutionary force and propositional content. They take place at an utterance level rather than sentence level, and are concerned with how the utterance affects the grounding process. This level of action is shown to fit coherently within a multi-level model of action in conversation including also levels for traditional speech acts as well as turn-taking acts and higher-level discourse coherence acts.

A protocol is designed which can determine, for any sequence of grounding acts, whether the content expressed by the utterances which comprise the acts is grounded or not. This protocol is also compared to mental models of language participants to provide rationales for taking particular actions. The protocol is then implemented within a NL conversation system, allowing the system to reason about the state of grounding and perform appropriate acknowledgements and repairs. Finally, the grounding acts used in the protocol are given precise definitions within the context of a general theory of plan execution.

1.5 Outline of Dissertation

Chapter 2 gives an overview of some of the previous research programs on which the current work builds. It also presents a more detailed analysis of the complexities of speech acts, user models, multi-agent interaction, action in conversation, and grounding.
Chapter 3 presents the framework of the grounding theory, providing a protocol for grounding in conversation, giving its rationale for construction, and its relation to discourse context and the mental state of the conversational participants.

Chapter 4 describes the theory of conversation acts, a multi-stratal theory of action in conversation, and shows how grounding fits in as a level of conversation acts, along with levels for traditional speech acts, turn-taking acts, and argumentation acts.

Chapter 5 describes an architecture for dialogue management that can be used to control the functioning of a natural language interface. This architecture includes a user model and use of conversation acts (including grounding acts) to update this model.

Chapter 6 describes the implementation and functioning of these ideas within the TRAINS Conversation System. Included is a trace of the system running on a sample dialogue, which is concluded in Appendix A.

Chapter 7 presents a formal theory of plan execution which is able to represent repair and sub-action. This theory is then used to formalize the grounding acts from Chapter 4.

Chapter 8 summarizes the contributions made in this dissertation and describes some natural extensions to the work described in the previous chapters.
2 Foundational Work

This chapter presents an overview of some of the previous research programs on which the work in the following chapters builds. It also presents an analysis of the complexities of speech acts, multi-agent interaction, action in conversation, and grounding. Section 2.1 describes some of the foundational work from both philosophy of language and AI in defining speech acts and formalizing them as planning operators in computational systems. Section 2.2 summarizes some of the work on modelling the mental states of agents, particularly belief, mutual belief, intentions and plans, joint intentions, and obligations. Most importantly for grounding is the problem of representation and acquisition of mutual belief between agents, which is also generally taken as one of the main intended effects of speech acts. Section 2.3 relates some of the most important insights from the subfield of Sociology known as Conversation Analysis which are relevant for the present topic. Section 2.4 describes previous attempts to incorporate ideas from Conversation Analysis into natural language processing systems. Section 2.5 examines the proposals put forth by Clark and his colleagues for a descriptive model of grounding. Finally, Section 2.6 presents some deficiencies with Clark’s model of grounding which must be overcome to provide a protocol for grounding.

2.1 Speech Acts and Plans

2.1.1 Foundational Philosophical and Linguistic Speech Act Work

Austin

Austin observed that utterances are not just descriptions of states of affairs, but are used to do things [Austin, 1962]. Under felicitous circumstances, utterances can change the mental and interactional state of the participants. Speaking is acting, and in speaking the speakers are producing speech acts. There are multiple types of action performed in speaking, and Austin distinguished several. Locutionary acts are the act of saying something, including a phonetic act – producing certain noises, a phatic act – producing words belonging to a vocabulary in a constructions conforming to a grammar, and a rhetoric act – using the product of the phatic act with a particular sense and reference.

Illocutionary acts are those acts performed in saying something, for example, asking or answering a question, giving some information, etc. Most of the subsequent work on
speech acts has been on illocutionary acts. Illocutionary acts are taken to be composed of an *illocutionary force*, which specifies the type of action (e.g., requesting, suggesting, warning, apologizing, informing), and a *propositional content* which specifies the details of the action (e.g., what it is that the hearer is being requested to do). Illocutionary acts are not always directly deducible from the locutionary acts which generate them. *Indirect speech acts* are those in which the act performed is other than what would be expected from a compositional account of the content. Austin gives a bridge example in which an utterance of “I bid three clubs” is used to inform a partner that the speaker has no diamonds.

*Perlocutionary acts* are those which are performed by saying something – actions which achieve effects which are special to the particular situation of utterance rather than the conventional nature of the communication. Examples include persuasion, surprise, and deterrence.

**Searle**

Searle extends and refines Austin’s work on illocutionary acts. He observes that Austin’s decomposition of speech acts into *illocutionary force* and *propositional content* shows up in the different kinds of negation that can be performed in a sentence [Searle, 1969]. For example, the sentence “I promise to come” has two negations: a *propositional negation*, “I promise not to come”, in which the illocutionary act (promise) is the same, but in which the content is negated, and “I do not promise to come”, in which the propositional content is the same, but the illocutionary act is no longer a promise, but a refusal to make a promise. Searle further decomposes the *propositional content* of an act as a combination of *predicating* and *reference* acts. Neither of these stand alone but are performed only in concert with illocutionary acts.

Probably the most important contribution was an attempt to provide necessary and sufficient conditions for the performance of illocutionary acts. He presented these as *constitutive rules* (like the rules which define the games of football or chess) of various sorts. *Normal input-output conditions* concern the conditions of intelligible speaking and understanding, including knowing the conventions of languages, paying attention, etc. *Propositional content conditions* describe restrictions on the content, e.g., for a promise the content must be a future action. *Preparatory conditions* involve the constraints on the world that make the speech act useful. *Sincerity conditions* involve alignment of the speaker’s actual attitudes (belief, desire, etc.) with the attitudes expressed by the act. *Essential conditions* involve the speaker’s intentions in performing the act – what she was trying to do. Searle also adds a condition based on Grice’s notion of non-natural meaning [Grice, 1957], that the effect of the act is in part produced by the hearer’s recognition that the utterance is intended to produce this effect by means of the recognition of the intention.

Searle also improves Austin’s classification of types of illocutionary acts [Searle, 1976]. Austin’s classification (into *verdictives, exercitives, commissives, expositives, and behaviitives*) is fairly haphazard, based more on similarity of illocutionary verbs than the acts themselves. Searle proposes an alternate taxonomy based on the purposes
of the acts. Searle’s taxonomy includes: *representatives*, which commit the speaker to the truth of an expressed proposition, *directives*, which involve getting the hearer to do something, *commissives*, which involve committing the speaker to some course of action, and *expressives*, which convey a psychological state of the speaker.

2.1.2 Foundational AI Speech Act Work

Bruce

Bruce was the first to try to account for Speech Act theory in terms of AI work on actions and plans [Bruce, 1975]. He defined natural language generation as *social action*, where a *social action* is one which is defined in terms of beliefs, wants, and intentions. He also presented *Social Action Paradigms* which showed how speech acts could be combined to form larger discourse goals. He showed how acts such as *Inform* or *Request* could be used in achieving intentions to change states of belief.

Allen, Cohen, and Perrault

Cohen and Perrault [1979] defined speech acts as plan operators which affect the beliefs of the speaker and hearer. They set up write that any account of speech acts should answer the following questions:

- Under what circumstances can an observer believe that a speaker has sincerely and successfully performed a particular speech act in producing an utterance for a hearer?
- What changes does the successful performance of a speech act make to the speaker’s model of the hearer, and to the hearer’s model of the speaker?
- How is the meaning (sense/reference) of an utterance $x$ related to the acts that can be performed in uttering $x$?

They also suggest that a theory of speech acts based on plans should specify the following:

- A planning system: a formal language for describing states of the world, a language for describing operators, a set of plan construction inferences, a specification of legal plan structures. Semantics for the formal languages should also be given.
- Definitions of speech acts as operators in the planning system. What are their effects? When are they applicable? How can they be realized in words?

These issues are still central to the work going on in discourse planning.

Cohen and Perrault’s models of mental states consist of two types of structures: *beliefs* and *wants*. *Beliefs* are modal operators which take two arguments: an agent who is the believer, and a proposition which is believed. They also follow Hintikka [1962],
augmenting the belief structure to include quantified propositions. Thus an agent can believe that something has a value without knowing what that value is, or an agent can believe another agent knows whether a proposition is true, without the first agent knowing if it's true or not. **Wants** are different modal operators which can nest with beliefs. **Wants** model the goals of agents.

Perrault and Cohen then proceeded to make a first stab at satisfying these issues. The planning system they use is a modified version of STRIPS [Fikes and Nilsson, 1971]. They maintain STRIPS's method of dealing with the frame problem: assuming that nothing can change the world except the explicit changes mentioned by the effects of an operator. They describe two different types of preconditions, both of which must hold for the action to succeed. **cando** preconditions indicate propositions which must be true for the operator to be applicable. **Want** preconditions are meant to cover sincerity conditions. In order to successfully perform an action, the agent (speaker) must want to do that action. They model the speech acts REQUEST and INFORM, within their planning system.

Allen and Perrault [1980] use essentially the same formalism as Cohen and Perrault, but for a slightly different purpose. They investigate the role of plan inference and recognition in a cooperative setting. They show how the techniques of recognizing another agent's plans can allow one to recognize an indirect speech act in a coherent and relevant manner. The planning system is again, basically a STRIPS system. There are preconditions and effects, and a body, which is a specification of the operator at a more detailed level.

**Litman and Allen**

Litman and Allen extend Allen and Perrault's work to include dialogues rather than just single utterances, and to have a hierarchy of plans rather than just a single plan [Litman, 1985; Litman and Allen, 1990]. They describe two different types of plans: domain plans and discourse plans. Domain plans are those used to perform a cooperative task, while discourse plans, such as clarification and correction, are task-independent plans which are concerned with using the discourse to further the goals of plans higher up in the intentional structure. They also use a notion of **meta-plan** to describe plans (including discourse plans) which have other plans as parameters. Using these notions, Litman and Allen are able to account for a larger range of utterances than previous plan-based approaches, including sub-dialogues to clarify or correct deficiencies in a plan under discussion. There is still no facility for explaining acknowledgment, as the assumption of perfect understanding is maintained.

**Non-monotonic Theories of Speech Acts**

Perrault [1990] takes as a starting point the problem that the utterance itself is insufficient to determine the effects of a speech act. All effects of utterance actions are based in part on the prior mental states of the agents as well as what was actually uttered. However, formalizing the precise conditions which must hold is a tricky endeavor, because
of the many possible contingencies. Thus an axiom stating the effects of an utterance in declarative mood must take account of the possibilities of lies, failed lies, and irony as well as standard information-giving acts. Perrault’s approach is to state the effects in terms of Default Logic [Reiter, 1980], so that the simple, most common effects can be derived directly, unless there is some defeater. He has a simple axiomatization of belief, intention and action, along with some normal default rules, including a Belief Transfer rule which says that if one agent believes that another agent believes something the first agent will come to believe it too, and a Declarative rule, which states that if an agent said a declarative utterance, then it believes the propositional content of that utterance. This simple schema allows Perrault to derive expected consequences for the performance of a declarative utterance in different contexts.

Although the formalization is simple and elegant, it still contains a number of serious difficulties. Foremost is the lack of a serious treatment of belief revision. Although intuitively, speech acts are used to change beliefs, Perrault’s framework can only handle the case of new beliefs being added. As well as not allowing the kind of discourses in which one agent would try to change the beliefs of another, the logic also has the strange property that one agent can convince itself of anything it has no prior beliefs about merely by making an utterance to that effect in the presence of another agent! The logic also does not lend itself to a computational implementation, since one would need a complete, inductive proof scheme to make all of the necessary deductions.

Appelt and Konolige [1988] reformulate Perrault’s theory in terms of Hierarchic Autoepistemic Logic [Konolige, 1988]. This reformulation has the advantages of implementability and the ability to order the defaults to overcome the problems that Perrault had with normal default logic, but it also loses the simplicity of Perrault’s framework. It is hard to see whether Appelt and Konolige are trying to describe something from the point of view of an ideal observer or from a participant in the conversation. In formulating their theory, they also resort to some unintuitive devices such as the beliefs of an utterance.

**Cohen and Levesque**

Cohen and Levesque have been attempting to solve a number of problems relating to formal characterizations of Speech Acts, through the use of a logic of action and mental attitudes. [Cohen and Levesque, 1990b] lays out the framework of the basic theory of rational action. It is based on a dynamic modal logic with a possible worlds semantics. They give axiomatizations for modal operators of beliefs and goals, and then derive intentions as persistent goals, those to which an agent is committed to either bring about or realize are unachievable.

[Cohen and Levesque, 1990c] uses this logic to show how the effects of illocutionary acts can be derived from general principles of rational cooperative interaction. They claim, contrary to [Searle and Vanderveken, 1985], that communicative acts are not primitive. They define what it means for an agent to be sincere and helpful, and give characterizations of imperatives and requests. They claim that recognizing the illocutionary force of an utterance is not necessary, that all that is important is that the
hearer do what the speaker want, not that he recognize which act the speaker performed as a part of this process. They thus appear to be claiming that illocutionary acts should be seen as descriptive models of action, not as resources for agents. They conclude with a description of how Searle and Vanderveken’s conditions on acts can be derived from their rational agent logic.

[Cohen and Levesque, 1990a] extends the framework to handle Performatives. They define all illocutionary acts as attempts. Performatives are acts which have a request component and an assertion component, and the assertion component is made true merely by the attempt, not the success of the action. Thus request is a performative verb, while frighten is not (because it requires a successful attempt and the success is beyond the control of the speaker), and lie is paradoxical when used performatively, because the explicit mention defeats the aim.

[Cohen and Levesque, 1991a] present an analysis of why confirmations appear in task-oriented dialogue. Using their theory of joint intentions developed in [Levesque et al., 1990] (described below in Section 2.2.4), they state that the participants in one of these task-oriented dialogues have a joint intention that the task be completed. It is part of the definition of joint intention that if one party believes the object of intention to be already achieved or to be unachievable, that party must strive to make the belief mutual. It is this goal of mutual belief which drives the agent to communicate a confirmation. Although this is perhaps the first attempt in the computational literature which is explicitly concerned with a plan-based account of the generation of confirmations, it is noticeably lacking in several respects. It has no mention of how the intention to make something mutually believed turns into an intention to perform a confirmation. There is also some distance still from the logic to actual utterances. It is not explained just what would count as a confirmation, and how one might recognize one.

Cohen and Levesque have provided a nice formal logic with which to precisely state and analyze problems of multi-agent coordination and communication, but it is difficult to see how it could be used by a resource-bounded agent in planning its actions or recognizing the intentions of others.

2.1.3 Multi-Agent Planning

A speech act theory which can account for conversations must include at least the following extensions to classical planning (e.g. STRIPS):

- temporal reasoning, including reasoning about overlapping and simultaneous actions
- uncertainty: attempted actions may fail to achieve their desired results, unexpected results may follow.
- multiple agents, each with individual knowledge, goals, etc.
- cooperation among agents
- real-time resource-bounded reasoning
integration of planning and acting

There is a large amount of research dedicated to addressing these problems, much more than can be summarized here. [Traum and Allen, 1991] explores some of the complexities involved in reasoning and acting in a richer environment. The annual European workshops on Modeling Autonomous Agents in a Multi-Agent World (MAAMAW) (reprinted in [Demazeau and Muller, 1990; Demazeau and Muller, 1991; Werner and Demazeau, 1992]) contain a variety of approaches to these problems.

2.2 Models of Mental States

One of the key features of speech acts as opposed to physical actions is that their main effects are on the mental and interactional states of agents, rather than on the state of some external domain. This section relates some of the most relevant work on representing and reasoning about several of these mental states, including belief, intentions and plans, mutual belief, shared plans, and obligations. This work serves as a background for the theory of Conversation Acts presented in Chapter 4 and a basis for the dialogue manager implementation in Chapter 5.

2.2.1 Belief

Belief and Knowledge are the attitudes that have been most studied in both the philosophical and AI literatures. Although in informal language, the word “knowledge” is often used to mean a strong belief, it has acquired a technical meaning which includes both belief and truth (often among other conditions). Since, in realistic situations, an agent does not generally have access to the actual state of affairs, we will generally refer to belief, though this is also the attitude generally modelled by knowledge bases. The literature on belief modelling is far too vast to attempt even a cursory summary, but we will mention some of the more influential approaches and some of their defects as models of actual belief.

The beliefs of an agent is that agent’s own model of how things are. In representing the beliefs of an agent, it is important not only to represent explicit facts that the agent is aware of, but also what else these facts suggest to the agent about the world. This usually managed by allowing some inference rules to augment explicit beliefs with other natural consequences. A predicate in simple first-order logic is not powerful enough to serve as a representation of belief, since the object of belief is not a set of simple terms, but is a formula itself. The usual solution, first proposed by Hintikka [1962], is to use a modal logic for belief. Hintikka later presented a semantics for belief based on accessibility functions over possible worlds [Hintikka, 1971]. A proposition is believed by an agent if that proposition is true of all the worlds in the accessibility relationship for that agent.

While this framework has been very influential, it has some serious drawbacks as an actual model of human belief. For one thing, it predicts that all logical consequences of beliefs will also be believed. While, in general, this can be a convenient facility, it is far
too strong. Unfortunately, there have yet to be developed adequate restrictions on this type of inference so that just the “correct” consequences will be modelled.

2.2.2 Intention

Intention is an attitude relating an agent to actions that she has decided to perform. Intentions also play a causative role in the occurrence of the actions and serve also to constrain the further deliberation of the agent. Bratman presents probably the most detailed theory of intention and its relation to other attitudes and processes [Bratman, 1987; Bratman, 1990]. Cohen and Levesque present a theory of intention based on more primitive commitments [Cohen and Levesque, 1990b]. Konolige and Pollack present a formalization of intentions within a non-normal modal logic [Konolige and Pollack, 1993].

Intentions are also important in utterance interpretation and plan recognition in general. Many analyses of utterances are concerned not just with what happened, but with what the speaker intended to do in performing the action. Also important is the structuring of individual intentions into plans and the Intentional Structure of Discourse [Grosz and Sidner, 1986] — how the intentions behind utterances in a discourse are related.

2.2.3 Mutual Belief

Most of the theories of speech acts as plans reported in Section 2.1 have as some of the main effects of speech acts the addition of some new mutual beliefs. Mutual beliefs are also taken to be some of the prerequisites for felicitous utterance of speech acts. But just what are mutual beliefs? This section reviews some of the proposals for how to represent the properties of mutual beliefs in terms of simpler beliefs, and how one could acquire new mutual beliefs. As with belief and knowledge, there have been a variety of names for a cluster of related concepts, including “mutual knowledge”, “common knowledge”, “mutual belief”. In the discussion below, we generally use the term used by the author under discussion, but treat these terms as synonymously meaning mutual belief, in which what is mutually believed by a group of agents is not necessarily actually true.

Formulations of Mutual Belief

While people agree for the most part about the intuitions underlying the phenomenon of mutual belief, there have been a variety of different ways proposed of modeling it. Barwise [1989] compares model theories for three different formulations.

Schiffer uses what Barwise calls “the itemize approach” [Barwise, 1989, p. 202]. He defines mutual knowledge between two agents A and S of a proposition \( p \) as \( K_{SPA} K_{A} p \wedge K_{S} p \wedge K_{ASP} \wedge K_{A} K_{S} K_{A} p \wedge K_{S} K_{A} K_{S} K_{A} p \wedge \cdots \). It is thus an infinite conjunction of nested beliefs. This approach has since been adopted by many others, including Allen [1983b] and Perrault, who provides an elegant default
logic theory of how to obtain each of these beliefs given prior knowledge and a conversational setting [Perrault, 1990].

Barwise credits Harman with the *fixed-point approach*. Harman formulates mutual knowledge as “knowledge of a self-referential fact: A group of people have mutual knowledge of p if each knows p and we know this, where this refers to the whole fact known” [Harman, 1977, p. 422]. As Barwise points out, the fixed-point approach is strictly stronger than the iterate approach, because it includes as well the information that the common knowledge is itself common knowledge. It also replaces an infinite conjunction with a self-referential one.

The final approach discussed by Barwise is the *shared-situation approach*. He credits it to Lewis. Lewis formulates rules for common knowledge as follows [Lewis, 1969, p. 56]:

Let us say that it is *common knowledge* in a population P that X if and only if some state of affairs A holds such that:

1. Everyone in P has reason to believe that A holds.
2. A indicates to everyone in P that everyone in P has reason to believe that A holds.
3. A indicates to everyone in P that X.

This schema is also used by Clark and Marshall, and is apparently the one which Barwise himself endorses.

[Cohen, 1978] uses a *belief spaces* approach to model belief. Each space contains a set of propositions believed by an agent. Nested belief is represented by nested spaces. There is a space for the system’s beliefs (SB) which can contain a space for the system’s beliefs about the user’s beliefs (SBUB) which in turn can contain a space for the system’s beliefs about the user’s beliefs about the system’s beliefs (SBUBSB). If Cohen were to adopt the iterated approach directly, it would require an infinity of belief spaces. Instead, he takes the space one deeper than the deepest which contains any non-mutual beliefs, and points it to its parent space, thus creating a loop, where each even nesting is the same as every other even nesting. Now each of the nested beliefs in the iterated approach can be generated or seen to be present in his belief spaces, by iterating through the loop. This approach shares some features with the fixed-point approach (the self-referentiality) and it allows quick determination of whether mutual belief exists (by searching for a loop), unlike the iterated approach, but it is in fact not as strong as the fixed-point approach because the higher-order implications of the fixed-point approach, such as mutual belief about the mutual belief, cannot be represented.

A slight modification is to add a separate kind of space, a *mutual belief* space to represent mutual beliefs. This is the approach taken by Bruce and Newman [1978]. The Rhetorical knowledge representation system [Allen and Miller, 1991] also uses a mutual belief space, but disallows nested beliefs within a mutual belief space, giving essentially the power of Cohen’s system. This also seems to be the approach used by Maida [1984].
How can Mutual Belief be Achieved?

If Mutual Belief includes at least the infinite conjunction of nested beliefs, there is a problem as to how to achieve mutual belief, or to recognize when it has been achieved. Several researchers have put forth proposals, as described below, yet none seem completely satisfactory.

Perrault uses an extremely strong set of assumptions to drive his default theory [Perrault, 1990]. He has an axiom of observability which states that if an agent is "observing" another agent, then he will recognize all actions (such as declaring a certain proposition) performed by that agent. Agents also have complete memory of prior beliefs, and persist their beliefs into the future (Perrault's theory can not handle belief revision). He also has two default rules, a belief transfer rule which states that if one agent believes that a second agent believes something, then the first agent should come to believe it (assuming it doesn't conflict with his prior beliefs), and a declarative rule which states that if an agent declares a proposition, then he believes it to be true. With Perrault's set-up, one can derive all the nested beliefs of the iterated approach, assuming there were no prior contradictory beliefs. In the case of some prior inconsistent beliefs, such as in the case of a lie or ironic assertion it also derives the correct set of beliefs. However, from a computational perspective it is difficult to see how an agent using Perrault's framework could recognize mutual belief without an infinite amount of computation (or at least some kind of inductive proof procedure for default logic). Perrault also doesn't mention what might happen in the case where his assumptions are too strong.

Clark and Marshall [1981] describe two kinds of heuristics to get at mutual knowledge in a finite amount of time. Truncation heuristics look at just a few of the nested beliefs, and then infer mutual belief if all of those check out. Copresence heuristics involve the agents recognizing that they and the object of mutual knowledge are jointly present. Clark and Marshall discount the truncation heuristics as implausible, since it is hard for people to reason overtly about nested beliefs. Also, the situation that usually provides evidence for the beliefs checked by the truncation heuristic is usually what would be used directly by the copresence heuristics.

They list four main ways of achieving the copresence necessary for mutual belief, with subdivisions of some of these. Their table with the auxiliary assumptions [Clark and Marshall, 1981, p. 43] is reproduced as Table 2.1.

Community co-membership is achieved when two agents mutually know that they are part of some community (e.g. people, squash players, computer scientists, etc.). Universality of knowledge refers to the assumption that certain things will be mutually known by everyone in a community. These two assumptions together, that two agents A and B are part of a community and that everyone in this community mutually knows x, are sufficient to conclude that A and B mutually know x.

The simultaneity assumption is that the agents are simultaneously in the same situation. The attention assumption is that the agents are paying attention to the shared situation. The rationality assumption is that the agents are rational, and can draw normal inferences. If the situation is a case of physical copresence, then if it is a case
Basis for mutual knowledge | Auxiliary assumptions
---|---
1. Community membership | Community co-membership, universality of knowledge
2. Physical copresence
   a. Immediate | Simultaneity, attention, rationality
   b. Potential | Simultaneity, attention, rationality, locatability
   c. Prior | Simultaneity, attention, rationality, recallability
3. Linguistic copresence
   a. Potential | Simultaneity, attention, rationality, locatability, understandability
   b. Prior | Simultaneity, attention, rationality, recallability, understandability
4. Indirect copresence
   a. Physical | Simultaneity, attention, rationality
   b. Linguistic | Simultaneity, attention, rationality, (locatability or recallability), associativity, understandability

| Table 2.1: Clark & Marshall’s Methods of Achieving Copresence for Mutual Knowledge |

of immediate copresence these three assumptions are sufficient, (e.g. Ann and Bob are looking at a candle, and looking at each other looking at the candle, so a definite reference of *the candle* is felicitous). If the situation is in the past, then an additional assumption of recallability is necessary: it's not enough that the situation occurred, they have to remember it. If the situation hasn’t happened, but very easily could, then locatability is needed. For example, say Ann and Bob are in a room with the candle, but not looking at it; then a reference is felicitous, assuming that the candle is locatable: the reference itself would provide the impetus for achieving the shared situation.

For linguistic copresence (reference to an object mentioned in prior discourse) an additional assumption is required, understandability. This is that the utterance which introduces the object can be understood as having done so. The final type of mutual knowledge is a mixture of community co-membership and one of the other two. An example of this is when a candle has been introduced, and then a definite reference to the price, or the wick is made. An additional assumption of associativity is needed to be sure that the hearer can make the connection.

There is still a problem with Clark and Marshall’s characterization, in fact, the same problem which motivated them to take up mutual knowledge in the first place. Their conditions for potential copresence are not sufficient. Taking the example they use to show the insufficiency of any finite set of nested beliefs for definite reference, we can see it is also insufficient for potential coreference. Assuming a prior episode of Ann and Bob looking in the morning newspaper and seeing that *A Day at the Races* is playing at the Roxy theater, if Ann, later sees a correction in the evening paper that the movie will be
Monkey Business, it would not be a felicitous reference to say "the movie at the Roxy" to mean Monkey Business. This is true even if Bob has seen the correction, and Ann knows Bob has seen the correction, and she knows he knows she knows he has seen the correction. As long as the sequence is finite, the chain always bottoms out, and we are left with A Day at the Races being the more felicitous, according to Clark and Marshall. But this is precisely the situation with potential coreference. In normal circumstances, Ann can ask if Bob has seen the movie even if she doesn't know if he knows what it is, as long as he can locate what the movie is — perhaps the paper is in front of him. But if we have the prior circumstance of joint knowledge of another referent, we have a problem, no matter how locatable the intended referent is. There are several difficulties in using Clark and Marshall's account: we must not only pick out the unique object in the situation which the definite description refers to, we must also pick out the (unique?) situation in which we can find such an object. This suggests first of all that Clark and Marshall's assumptions for potential copresence are insufficient, and secondly, that perhaps, as Johnson-Laird suggests [1982, p. 41], their examples do not show that mutual knowledge is necessary. Clark and Carlson [1982, p. 56] counter that they are talking about mutual expectation and belief as much as knowledge, and thus Johnson-Laird's proposed counterexamples are not problems for their account. There is still the following potential difficulty: as shown above, the assumptions for potential copresence are not sufficient; therefore, something else is needed. Perhaps this something else can also get them out of the original problem without recourse to mutual knowledge. To use their formulation, one must at least work out the relationships between different basis situations, and provide a method of choosing which of competing assumptions should be made.

Clark and Marshall recognize that reference can fail and can be repaired. They distinguish two types of repair, which they term horizontal repair and vertical repair. Horizontal repair refers to giving more information about the item, but keeping the basis (the type of copresence) the same, whereas vertical repair is giving a new basis (with presumably fewer assumptions) such as pointing out an item to change physical copresence from potential or prior to immediate.

While the above assumptions may be sufficient for an expectation of mutual belief and felicitous use of a definite referring expression, they are not sufficient to provide actual mutual belief because of the possibility of error (and possible repair). If A makes a reference, she cannot be sure that it will be understood by B. Because of this, even if B believes he understands the reference (he still might be mistaken) he cannot be sure that A believes he does. Each further nested statement introduces more and more uncertainty, and after a while, one of them must be certain to be disbelieved. There is a wealth of linguistic evidence that understandability and attention (or "observability" in Perrault's scheme) are not just mutually assumed. Statements in discourse are often acknowledged by the listener to provide the speaker with evidence that he has been heard and understood. Utterances like "okay", "uh-huh", and "mmh" are often used to acknowledge the previous utterance. With observability assumed, there would be no need to ever make such utterances.

Perner and Garnham [1988] show some additional problems with Clark and Mar-
shall’s copresence heuristics. They end up proposing something very much like the shared situation approach from [Lewis, 1969], with the additional restriction that the indications to the population that the situation holds be based on mutual beliefs.

Halpern and Moses [1990] present several notions of group knowledge, ranging from implicit group knowledge to full mutual knowledge. They also offer a proof that mutual knowledge is unachievable in an unsynchronized noisy environment, where communication is not guaranteed. They also investigate weaker notions of common knowledge that are achievable.

2.2.4 Coordinated Activity, Shared Plans and Joint Intention

Although most agree that some sort of collective intentional intentional attitude is useful in formalizing an account of cooperative behavior, it is still fairly controversial what the properties of such an attitude should be and how this collective intention is related to the individual intentions of the participants. Important questions include: how do shared intentions guide individual action, and how can individual beliefs and intentions come together to form shared intentions? This section reviews some previous proposals. Our own attempts at formalizing this notion are presented in Section 7.5, below.

Lewis [1969] defined a Convention as a situation in which there is some regularity R in behavior in a population, and everyone conforms to R, everyone expects everyone else to conform to R, and everyone prefers to conform to R, given that everyone else will. A typical example is which side of the road to drive on. In England it is the left side, in America, the right. It doesn’t really matter to the drivers which side to drive on, as long as everyone agrees. Coordinated activity is thus seen as individual intention in a state of mutual knowledge about norms. Knowledge of conventions serve to make it in the mutual self-interest of each of the members of the population to follow along.

Grosz and Sidner [1990] take basically the same viewpoint. They extend Pollack’s definition of a Simple Plan [Pollack, 1990], to that of a SharedPlan, which is formalized as a set of mutual beliefs about the executability of actions and the intentions of particular agents to perform parts of that action. They also present some conversational default rules based on cooperativeness to use communication to add to the shared beliefs.

Searle [1990] starts with the intuition that collective intention is not just a summation of individual intentions. He wants to distinguish between just following a convention and actual cooperative activity. He postulates that we-intentions are a primitive form of intentionality, not reducible to individual intentions. There is still a problem of how we-intentions can produce the individual intentions necessary for an individual to act.

Cohen and Levesque present their own theory, not in terms of individual intentions (which also aren’t primitive in their theory) but in terms of mutual belief and weak mutual goals [Levesque et al., 1990; Cohen and Levesque, 1991b]. Their formulation says that the individuals each have the goals to perform the action until they believe that either it has been accomplished or becomes impossible. Also in the event of it becoming completed or impossible, the agents must strive to make this belief mutual. This framework is also used to explain certain types of communicative behavior such as confirmations as described above in section 2.1.
2.2.5 Obligations

Obligations represent what an agent should do, according to some set of norms. The notion of obligation has been studied for many centuries, and its formal aspects are examined using deontic logic. Generally, obligation is defined in terms of a modal operator often called permissible. An action is obligatory if it is not permissible not to do it. An action is forbidden if it is not permissible. An informal semantics of the operator can be given by positing a set of rules of behavior R. An action is obligatory if its occurrence logically follows from R, and forbidden if its non-occurrence logically follows from R. An action that might occur or not occur according to R is neither obligatory nor forbidden. Von Wright, [1951] presented one of the most influential early systems of axioms relating these operators. McCarty [1986] provides an account based on actions rather than states, which also allows the set of obligations to change dynamically, as agents make choices which might incur new obligations. Conte and Castelfranchi [1993] show how obligations share some features of goals and some of beliefs, but cannot be reduced completely to either.

2.3 Conversation Analysis

The primary aim of the subfield of sociology known as Conversation Analysis\(^1\) (henceforth CA) has been to study actual conversations and inductively discover recurring patterns found in the data. Although the professed aims seem to be to steer away from intuitions or prior formalization, CA has produced a number of useful insights for how natural language conversation is organized, and which features of conversation a conversant should orient to. Although the conversation analysts do not formulate it in this way, they examine some of the properties of conversation which show it to be the result of interactions among multiple autonomous agents. The rest of this section is devoted to a brief overview of some of the most relevant findings for designing a computational system to converse in natural language.

2.3.1 Turn-taking

Sacks et al., [1974] present several observations about the distribution of speakers over time in a conversation. Although there are frequently periods of overlap in which more than one conversant is speaking, these periods are usually brief (accounting for no more than and often considerably less than 5% of the speech stream [Levinson, 1983, p. 296]. Conversation can thus be seen as divided into turns, where the conversants alternate at performing the role of speaker. The “floor” can be seen as an economic resource whose control must be divided among the conversants. Although in general conversation (as opposed to more formal communicative settings such as debates, court trials, or classes) there is no predetermined structure for how long a particular turn will last, there are locally organized principles for shifting turns from conversant to conversant.

\(^1\)This gloss of some of the findings of Conversation Analysis comes mainly from [Levinson, 1983]
Turns are built out of *turn constructional units*, which correspond to sentential, clausal, phrasal or lexical syntactic constructions [Sacks *et al.*, 1974, p. 702]. Following a turn constructional unit is a *transition relevance place*, which is an appropriate moment for a change of turn. Subsequent turns can be allocated by one of two methods: either the current speaker can select the next one (as in a question directed to a particular individual), or the next speaker can self-select, as in an interruption or restarting after an unmarked pause.

One important observation about turn-taking is that it is locally managed. The length and structure of a turn is an emergent property of interaction rather than a predetermined structure. The length of one speaker's turn will be determined by the speaker and other conversants who might end things at different times by taking over. A speaker can direct another to speak, but this does not by itself effect a transfer of the turn – the other must pick it up as well. Keeping the stream of talk to mostly be used by a single speaker at a given time is a coordination problem similar to that of two motorists crossing each other’s path (though with less drastic consequences for failure). Schegloff [1987] presents an explanation of how conversants re-utter overlapped talk at the beginning of turn transitions.

### 2.3.2 Adjacency Pairs

Adjacency pairs are pairs of utterances that are [Levinson, 1983, p. 303]:

1. adjacent
2. produced by different speakers
3. ordered into a *first part* and a *second part*
4. typed so that a particular first requires a particular (range of) second(s)

Typical examples of adjacency pairs are question-answer, greeting-greeting, offer-acceptance, and assessment-agreement.

First parts and second parts are connected not by some sort of grammar rule for legal conversations, but because the first will make the second *conditionally relevant*. The following utterance by the speaker after the utterer of the first should be either a second, an explanation that the second is not forthcoming, or something preparatory to a second, e.g., a clarification question. Utterances that come between a first and its second are called *insertion sequences*.

There are two types of seconds that can follow a first. These are known as *preferred* and *dispreferred* responses. Preferred responses are generally direct follow-ups and are unmarked. Dispreferred responses are generally marked with one or more of the following: pauses, prefices (such as “uhh” or “well”), insertion sequences, apologies, qualifiers (e.g., “I’m not sure but . . .”), explanations [Levinson, 1983, p. 334]. Table 2.2 (from [Levinson, 1983, p. 336]) shows some common adjacency pairs with preferred and dispreferred seconds.
First Parts:

Request  Offer/Invite  Assessment  Question  Blame

Second Parts:

Preferred:  
acceptance  acceptance  agreement  expected  denial

Dispreferred:  refusal  refusal  disagreement  unexpected  admission

Table 2.2: Adjacency Pairs

Adjacency pairs can thus serve as contextual resources for interpreting utterances. If a first part has been made, it makes a second conditionally relevant. The next utterance can be checked to see if it forms a plausible second. Markedness or its absence can be seen as pointing to the preferred or dispreferred second.

2.3.3 Repairs

Repairs can be characterized as attempts to fix previous utterances that are perceived to be (possibly) insufficient for conveying what was intended. Repairs include both clarifications, in which new information is added, and corrections, in which changes are made. Repairs are classified as to who made them (self or other), initiated them (self or other), and how many utterances they are removed from the utterance that they are repairing. In the first (same) turn we can have only self-initiated self-repair. In the second turn, we can have other-repair or other-initiated self-repair. There is also third-turn repair (when the Initiator subsequently determines, in virtue of the other’s previous utterance, that he has been misunderstood), and fourth-turn repair, (when the other later realizes that his own interpretation was in error). One can initiate a repair by the other conversant with a next turn repair initiator (or NTRI), which seems to be basically the same as a clarification question. Schegloff et al. [1977] show that a preference scheme exists for when to perform a repair. The highest preference is to perform self-initiated self-repair in the same turn. The next most preferred is to perform self-initiated self-repair in the transition space between turns. Then other-initiated self-repair in the next turn, by means of an NTRI. The least preferred is other-initiated other-repair.

2.4 Previous attempts to incorporate CA in NLP systems

Although there has been an awareness of the work from Conversation Analysis among some AI researchers for some time (at least since [Hobbs and Evans, 1979]), it is only
recently that several researchers have begun attempting to incorporate the findings from Conversational Analysis into computational systems for understanding natural language or human-computer interaction.

Suchman [1987] contrasts the classical planning framework, characterized by complete knowledge and forming fully specified plans, with the situated nature of real-world action, in which too much is changeable and unknown to plan in complete detail far in advance. In the situated view, “plans are best viewed as a weak resource for what is primarily ad hoc activity” [Suchman, 1987, p. ix]. She also presents some of the observations and methods of Conversation Analysis, and uses them to analyze the behavior of a computer program for instructing users of a photocopier, which functions by attributing a state of certain sensors on the machine to a step in one of the possible plans for making different kinds of copies. She finds that many of the problems the users have in understanding the instructions of the system come about as a result of the system not conforming to typical patterns of conversation usage. The system would intend one thing which would be understood as another by the users. Suchman calls for system designers and researchers in conversation planning to use the rules of conversation as resources that the system should orient on.

This call to design interfaces which are based on the observations of conversation analysis has been taken up by several of the researchers whose work appears in [Luff et al., 1990]. Frohlich and Luff [1990] have tried to use the principles of CA in building The Advice System, a natural language expert system front-end. Although the system uses mouse-controlled menu-based input, and exercises fairly authoritarian control over what can be said, it at least pays lip service to the findings of CA, including adjacency pairs, turn constructional units, repairs (including NTRIs), standard openings and closings (including pre-closings), and preferred and dispreferred responses. They have “a declarative definition of the interaction between user and system” [Frohlich and Luff, 1990, p. 201] composed of elaborate logical grammar rules which specify legal conversations down to low-level details. These rules serve both to update the context as the conversation progresses and to help the system choose what to do next.

Although the Advice System seems to be a step in the right direction, there are several problems with it in practice. First, it is much too restrictive in its input to be called real conversation. Its notion of utterance types is restricted to Questions, Answers, and Statements. Although the designers consider all possible combinations of any of these by speaker and hearer, they reject far too many as being impossible. Though something can only be an answer if there is an outstanding question, and the existence of an outstanding question will tend to make a next utterance be seen as an answer, there seems to be no reason to outlaw statements immediately following questions uttered by the same agent. This is a very common pattern for repairs (e.g., the same speaker successively uttering “Where is the Engine? I mean Engine E3.”). The only point at which a user can interrupt is at a Transition Relevance Place, whereas in real conversation, that is merely the most common and expected place. The menu-based input also trivializes the interpretation problem, and it’s unclear why the user should ever have to make a repair. Empirical testing will show if users find the Advice System usable or not, but it may well suffer from the same problems that Suchman’s copier
system suffers from: using familiar patterns in unfamiliar ways, ending up misleading
the user.

Raudaskoski [1990] describes an attempt to study local repair mechanisms in a
telephone message-leaving system. She allowed five different kinds of repair initiators,
and ran a simulated experiment where a person acted as intermediary, typing input
which was spoken by the user, and reading the responses of the system over the phone.
The experiment didn't work very well, mainly because the system could only interpret
a very small set of inputs, which the users generally went beyond. Also, the repair
mechanisms didn't work very well: the full variety was not used, and those that were
used often led to misunderstanding. One of the system's repair initiators seemed too
much like a confirmation, so the user thought she had succeeded in leaving the message
and went on to the next message but the system was still hoping to get the user to start
over.

Cawsey [1990] describes the EDGE system, which is an expert/advice giver, that
combines AI planning with CA local interactions. It has a model of discourse structure
based on that of Grosz and Sidner [1986], and plan schemas, which it uses to construct
explanations. It also allows local interactions, including forcing the user to mouse-click
acknowledgement after every utterance, and allowing the user to break in with repair
initiators. Planning is done when required (e.g. to fill in the content for a user requested
repair), not in advance.

Cawsey [1991] uses the endorsement based ATMS model for belief revision presented
by a belief revision scheme presented in [Galliers, 1990] to model third and fourth-turn
repair. The belief revision scheme keeps a set of endorsements with each assumption
and, when conflict occurs, throws out the assumption set with the weakest endorsement.
Cawsey uses a Speech act plan-recognition system based on that of [Perrault and Allen,
1980], but makes the interpretations assumptions, subject to change if conflict occurs.
Thus the system can change its interpretations of prior utterances to bring them in line
with new evidence.

Probably the most ambitious synthesis of ideas from both Conversation Analysis
and AI theories of speech acts and mental states is that of McRoy [1993]. Each speech
act is paired with both a set of intentions, representing the mental attitudes expressed
by the speaker in performing the speech act, and a set of expectations, representing
the acts which are expected to follow the speech act in the course of a dialogue. These
expectations come from the adjacency structure pairs from CA. A uniform abductive
framework is used to model the discourse structure revision and to be able to generate
appropriate speech acts, interpret surface speech act utterances using discourse coherence,
detect and repair misunderstandings, including third and fourth-turn repair.

2.5  Grounding in Conversation and the Contribution Model

Clark and several of his colleagues have looked at coordination and collaborative activity
in conversation, making explicit reference to both the traditions of Conversation Anal-
ysis and Speech Act Theory [Clark and Wilkes-Gibbs, 1986; Clark and Schaefer, 1989; Brennan, 1990; Clark and Brennan, 1990; Clark, 1992]. They try to identify several principles serving to guide collaborative behavior to account for the kinds of things observed by the Conversation Analysts.

One of the points that they make is that conversants need to bring a certain amount of common ground to a conversation, in order to understand each other. They call the process of adding to this common ground *grounding*. Grounding can be seen as adding to the mutual beliefs of the conversants (in fact it is glossed this way in [Clark and Schaefer, 1989]), but it seems reasonable to make a distinction. Though mutual belief, as defined by any of the proposals described in Section 2.2.3, is probably sufficient for common ground, it may be that only some weaker notion is actually necessary, and that we can have some sort of common ground without full mutual belief.

Clark and Schaefer [1989] present a model for representing grounding in conversation using *contributions*. Contributions are composed of two parts. First, the contributor specifies the content of his contribution and the partners try to register that content. Second, the contributor and partners try to reach the *grounding criterion*, which Clark and Schaefer state as follows, "The contributor and the partners mutually believe that the partners have understood what the contributor meant to a criterion sufficient for the current purpose" [Clark and Schaefer, 1989, p. 262]. Clark and Schaefer divide the contribution into two phases as follows (for two participants, A and B) [Clark and Schaefer, 1989, p. 265]:

**Presentation Phase**: A presents utterance u for B to consider. He does so on the assumption that, if B gives evidence e or stronger, he can believe that B understands what A means by u.

**Acceptance Phase**: B accepts utterance u by giving evidence e' that he believes he understands what A means by u. He does so on the assumption that, once A registers evidence e', he will also believe that B understands.

Clark and Schaefer claim that once both phases have been completed, it will be common ground between A and B that B understands what A meant. Each element of the contribution may take multiple conversational turns and may include whole embedded contributions. Rather than a straightforward acceptance, B can instead pursue a repair of A's presentation, or ignore it altogether. B's next turn, whether it be an acceptance, or some other kind of utterance, is itself the presentation phase of another contribution. Thus A must accept B's acceptance, and so on.

There are different types of evidence which can be given to show understanding. The main types considered by Clark and Schaefer are shown in Table 2.3, in order from strongest to weakest.

The strength of evidence needed for grounding depends on several factors, including the complexity of the presentation, how important recognition is, and how close the interpretation has to be. They try to avoid infinite recursion in accepting acceptances by invoking the following *Strength of Evidence Principle*: The participants expect that, if evidence e_0 is needed for accepting presentation u_0, and e_1 for accepting presentation of e_0, then e_1 will be weaker than e_0.
Clark and Wilkes-Gibbs [1986] present a **Principle of Least Collaborative Effort** which states that “In conversation the participants try to minimize their collaborative effort – the work that both do from the initiation of each contribution to its mutual acceptance.” This principle is contrasted with Grice’s maxims of quantity and manner, which concern themselves more with least effort for the speaker.

Clark and Brennan [1990] show how the principle of least collaborative effort can help in explicating the preferences for self-repair shown by [Schegloff et al., 1977] (described above in Section 2.3). They also show how this principle predicts different types of grounding mechanisms for different conversational media, depending on the resources that are available and their costs in those different media.

Brennan [1990] provides experimental evidence for how grounding takes place in conversational tasks, and the principles described above. She has a computer-based location task, where one party (called the director) must describe where on a map the other (the matcher) is to point his cursor. The experiment is broken down along two dimensions: familiar vs. unfamiliar maps, to change the grounding criterion, and trials where the director can see where the matcher is vs. trials where the director cannot, and must rely on verbal descriptions from the matcher, to change the strength and type of evidence available for accepting presentations. As might be expected, participants took longer to describe and find locations on the unfamiliar map, and the grounding process was shorter where more direct evidence was available.

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Display</strong></td>
</tr>
<tr>
<td>2</td>
<td><strong>Demonstration</strong></td>
</tr>
<tr>
<td>3</td>
<td><strong>Acknowledgement</strong></td>
</tr>
<tr>
<td>4</td>
<td><strong>Initiation of relevant next contribution</strong></td>
</tr>
<tr>
<td>5</td>
<td><strong>Continued attention</strong></td>
</tr>
</tbody>
</table>

Table 2.3: [Clark and Schaefer, 1989, p. 267]: Types of Evidence of Understanding
2.6 Deficiencies of the Contribution Model

Although the contribution model is perhaps the first explicit model of how grounding takes place and why acknowledgements occur, it still is lacking in a number of particulars. For one thing, it is often hard to tell whether a particular utterance is part of the presentation phase or the acceptance phase. Self-initiated self-repair and other-agent completions are considered part of the presentation phase, but other-repair and other-initiated self-repair are part of the acceptance phase. Either one can have embedded contributions, in the form of insertion sequences or clarification sub-dialogues, so, in the case of an other-initiated self-repair, it's hard to tell whether it is part of the presentation phase or the acceptance phase. We often need to look at large segments of the conversation, both before and afterwards before deciding how a particular utterance fits in.

Since Clark and Scahefer assert that each signal (including acceptances) must itself be a presentation which needs acceptance, it is not clear that contributions are ever really complete. For example, in the simplest case, contributions by turns, Speaker A's first turn is a presentation part of a first contribution. Speaker B's following turn is an acceptance part of that contribution, but also is the presentation part of a next contribution. What is unclear is whether this second utterance must be accepted in order to fulfill its acceptance function. If so, as Clark and Shaefer seem to imply, then not even the next utterance by A will completely ground the first contribution: this acceptance of the acceptance will itself need to be accepted, and so on, ad infinitum. If it is possible, as they suggest, that some acceptances need not be accepted themselves, then this opens the possibility that the acceptance part of an utterance need not be itself accepted (though any actual intended next contribution which is part of the same utterance would still have to be accepted). Some utterances which are merely acceptances might not be presentations as well.

The model also seems insufficient to use as a guide for an agent in a conversation deciding what to do next based on what has happened before. Realizing that a presentation has been made but has not yet been accepted can lead one to initiate the acceptance phase, but it's not clear when a presentation or acceptance is complete, or whether the knowledge of being in the presentation phase or acceptance phase has any consequences for what should be uttered.

In the next chapter, we will attempt to solve these problems using a modified model of grounding. We replace the presentation and acceptance phases with more atomic utterance level acts that build up complete contributions according to a grammar which will show, at every stage in the process, whether the contribution is grounded, what acts are allowed to follow, and what it will take to become grounded.
3 Towards a Computational Theory of Grounding

This chapter presents the foundations of a computational theory of grounding that is based on how individual utterance level actions contribute towards the grounding process. The central theoretical constructs introduced are the discourse unit (DU), which is the level of structure at which conversational content is grounded, and grounding acts, which are the actions performed in producing particular utterances which contribute to this groundedness. This chapter presents a theory of how DUs are constructed from a sequence of grounding acts, providing constraints on possible grounding act sequences, a decision procedure for determining whether a discourse unit is grounded, and setting up expectations for preferred sequences which will lead to a grounded unit from a particular ungrounded but accessible discourse unit. It thus forms a theory of grounding which is competitive with that of Clark and Schaefer described in Section 2.5.

Section 3.1 builds up a recursive transition network model of discourse unit construction, starting from a model like Clark and Shaefer’s contribution model and amending it to handle single utterance grounding acts. Section 3.2 presents a finite-state model which is roughly equivalent to the recursive model but computationally simpler. It then continues by describing how this model can be used as a protocol for grounding. Section 3.3 describes a cognitive model of grounding that is based on the mental states and processes of agents involved in conversation. The grounding acts in the previous section are related to their effects on the cognitive model and the reasons why an agent might decide to produce them to accomplish conversational purposes. Section 3.4 shows how this model can be used to explain the distribution of utterance acts described in Section 3.2. Section 3.5 presents some problematic conversation sequences for this account of grounding, considers some alternative accounts and suggests some simple solutions.

3.1 Tracking Grounding with Discourse Units

From a processing point of view, the main deficiency of Clark and Schaefer’s contribution model (described in Section 2.5) of grounding is that there is no easy way to tell the “state” of the current contribution while engaged in a conversation. Although we might represent a contribution as a transition network such as that in Figure 3.1, with a grounded contribution being one in the final state, F, this is not sufficient to monitor on-line conversation.
We know that to start a contribution, a **presentation** must be performed primarily by one agent, whom we will call the *Initiator* (I) and then an **acceptance** must be performed primarily by the other agent, whom we will call the *Responder* (R), but what is less obvious is how to tell when a presentation has been performed. Another way of looking at this question is: given an utterance by the initiator, how does it function as part of the current contribution? Does it start, continue, or complete a presentation? Unfortunately, there is no way to recognize whether a presentation is complete, just by looking at an utterance itself. Consider the following sequences in examples (6) and (7).

\begin{enumerate}
\item I: Move the boxcar to Corning  
  I: and load it with oranges  
  R: ok
\item I: Move the boxcar to Corning  
  R: ok  
  I: and load it with oranges  
  R: ok
\end{enumerate}

Since, according to Clark and Shaefer, the sequence in example (6) is a single contribution, that means that the presentation phase must encompass both of the utterances by the initiator. However, in example (7), there are two contributions with two separate presentations by I, and thus the first utterance by the initiator is a complete presentation. Since these sequences are identical up to the second utterance, there is, in general, no way to tell whether a presentation is complete until another action starts. This becomes a more serious matter because a contribution must be composed of actions by both participants, and thus there must be some way for the individual participants to determine an appropriate next action, given the current state.

Since the notion of a **presentation** does not seem to provide much help in processing contributions, we will replace **contributions** with a similar notion: **discourse units**. Like contributions, discourse units (DUs) also represent the unit of conversation at which grounding takes place, but they are composed of individual utterance-level actions.

\footnote{Clark & Shaefer would also analyze the acknowledgements by R as the presentation phases of separate contributions.}

---

**Figure 3.1: Transition Network for Contributions**

![Transition Network for Contributions](image-url)
rather than (potentially recursive) phases. The opening utterance of a DU will be called an **initiate** act, subsequent utterances which add on new material in a presentation (such as the second utterance in (6)) will be called **continue** acts. We will also call the act of claiming understanding of a previous utterance an **acknowledgement** (or **ack** act), regardless of the type of utterance it is and whatever other functions it might have in other discourse units. Thus a first approximation at a transition network for DUs (ignoring repairs) would look like figure 3.2, where the actor is given in parentheses (I for initiator, R for responder).

![Transition Network for DUs](image-url)

Figure 3.2: Initial Transition Network for DUs

This network would be sufficient to determine the grounding status of any sequence of connected utterances followed by their acknowledgement. From this basic model, we can now add in a facility to allow repairs by both parties. First, we will want to add a “dead” state, to cover cases in which the initiator retracts the proposed contribution or the conversation proceeds in such a way that the material can no longer be grounded (but would have to be brought up again in a new contribution). We call an utterance that makes the material in the DU ungroundable a **cancel** act.

It will also be a simple matter to incorporate self-initiated self-repair to this network. These repairs will change the material under consideration (whereas a **continue** would just add additional material), but, still, the responder can acknowledge the updated contents with a single acknowledgement. Figure 3.3 gives an amended network which allows cancels and self-initiated self-repairs.  

To allow other-initiated self-repair and other repair, we need a slightly more complex theory. Unlike the simpler cases above in which, once a DU has been initiated, it could always be grounded with a single acknowledgement by the responder, these cases will require other utterances before being grounded. The question thus arises as to how much extra complexity is needed. Will adding an extra few states to the network suffice, or is a recursive network (or something more) required?

For presentational simplicity, we will start with a recursive transition network, following Clark and Schaefer’s model of nested contributions. A repair by the responder must be acknowledged by the initiator in order to be grounded. Thus, we add a sub-

---

2It remains open to question whether the link from the final state, F, to the dead state, D, in Figure 3.3 and subsequently should be allowed. This is equivalent to the question of whether an initiator can cancel a proposed contribution after it has been acknowledged, or whether the effect of changing assumed mutual belief can result only from the grounding of new DUs. Further study of these sequences is still required to answer the question conclusively.
network that looks identical to our original DU network, except that the initial act is a repair instead of an initiate (one can think of this repair as an initiate of a repair insertion sequence). In addition, this sub-DU will have the initiator and responder switched, since it is the responder of the main unit who initiates this repair. Another recursive network needed is one for other-initiated self-repair. In the case of a repair request, rather than acknowledging, the initiator will have to perform a repair before the responder can acknowledge the whole DU.

We represent recursive transitions (as opposed to single utterance transitions) in all caps. In addition, the agent who plays the role of initiator of the sub-network is shown within square brackets. Thus Repair[R] indicates a transition in which the Repair network must be followed and in which the responder of the current network is the initiator of the Repair network. Figure 3.4 shows the modified DU network, Figure 3.5 shows the network for (other) repairs, and Figure 3.6 shows the network for repair requests (other-initiated self-repairs), abbreviated as *REQ-REPAIR*.
Figure 3.5: Transition Network for REPAIR Sub-DU

Figure 3.6: Transition Network for REQ-REPAIR Sub-DU
Following the REPAIR[R] means traversing the REPAIR sub-DU network, from the start state (RS) to the final state (RF). This complete traversal will cause the sub-DU network to be popped and the parent network will now be in the resulting state. Also, if the repair is cancelled (ending up in state RD), then it is as if the repair network were never started, the sub-DU network is popped, and the parent network is in the state it was in when things started. Note that these extra networks are recursive since the REPAIR and REQ-REPAIR networks also allow repairs and repair requests.

We will also add two more capabilities to the main network. First, we’d like to allow requests for acknowledgement by the initiator. These will signal that the speaker believes the current unit is complete and wants an acknowledgement before proceeding. Also, we will allow further acts from the final state. After an acknowledgement, it is usually possible to add extra acknowledgements, while remaining grounded. It is also possible for a short while to reopen a completed DU with a further repair. We thus amend Figure 3.4 to Figure 3.7. This ability helps to alleviate one of the problems with Clark and Shaefer’s Contribution model. Since a contribution was not deemed complete until nothing more was added to it, the acceptance phase was not considered complete after an acknowledgement, because the initiator could still acknowledge in return or reject the acknowledgement with a repair. However, these actions are not mandatory, so we represent the state of the DU after each act, and can determine whether a DU is considered grounded by tracking the sequence of acts that have occurred as part of the DU.

![Transition Network](image)

Figure 3.7: Modified Transition Network for DUs with recursive repairs

### 3.2 A Finite-State Theory of Grounding

An important question is whether the recursion is really necessary. Is this unbounded depth of nesting of repairs actually used in dialogue, or do participants always get lost after a few levels and have to cancel and start again? If so, then a finite-state model
might be more efficient. A more serious question is whether such nesting really occurs at all, or whether everything is at the top level: does one really have to pop out of each level of repair before proceeding with the previous repair? In this section, we develop a finite-state model which is competitive with the recursive model concerning coverage of the observed sequences in the TRAINS corpus.

For a finite-state model, we do need at least some extra states beyond those of Figure 3.7 to represent the results of repair. We can begin by folding in the repair networks in Figures 3.5 and 3.6. After a reqRepair by the responder in the main DU (the initiator of the REQ-REPAIR sub-DU), we need a repair by the initiator (the responder of the sub-DU) to get back to state 1 in the main DU. We can achieve this by adding a state 2 to the network in Figure 3.7, which is reached by a reqRepair by the responder and from which a repair by the initiator will return the DU to state 1. This state 2 thus corresponds to state RR1 in the first level REQ-REPAIR network. In addition, we will need another state for the result of a repair by the responder, corresponding to state R1. We will call this state 3. The resulting network will look something like Figure 3.8, although, for readability, this only shows the transitions to and from state 1.

![Figure 3.8: Finite-State Network with single level of repairs](image)

This network is identical to the recursive network for single depths of nesting. However, we can extend the finite-state model to cover additional depths of nesting. First, we will also want to allow the possibility of a repair request by the initiator of the DU after some sort of response by the responder. This would correspond to a REQ-REPAIR[I] network from state F, or a REQ-REPAIR[R] network from the first level of either the REQ-REPAIR[R] (state 2) or REPAIR[R] (state 3) networks. This will be
state 4, an analogue of state 2, in which a repair by the responder is expected.

Finally, we will need to decide what should happen when further repairs are performed. A plausible option is to treat all repairs as cumulative, operating on the content on the floor, and merely (perhaps) shifting the initiative and discourse obligations. State 1 represents initiative with the initiator and no discourse obligations (beyond those for the responder to acknowledge); state 2 represents initiative with the initiator and a discourse obligation for the initiator to repair; state 3 represents initiative with the responder and and no discourse obligations (beyond those for the initiator to acknowledge); state 4 represents initiative with the responder and a discourse obligation for the responder to repair.

So, to Figure 3.8, we add a link from state 3 for a repair by the initiator to state 1. In addition, in state 3, we consider the DU grounded if the initiator acknowledges the repair, without requiring an additional acknowledgement by the responder. Thus we change the arc for Ack(I) to go directly from state 3 to state F instead of state 1. Figure 3.9 shows the network from the point of view of state 1, Figure 3.10 from that of state 2, Figure 3.11 from that of state 3, Figure 3.12 from that of state 4, and Figure 3.13 from that of state F. Table 3.1 shows a transition table for the entire network.

Agents may take different roles in different DUs in a mixed-initiative conversation. A completed discourse unit is one in which the intent of the initiator becomes mutually understood (or grounded) by the conversants. An open DU is one which has been initiated but to which agents may still contribute.

While there may be some confusion among the parties as to what role a particular
Figure 3.10: Transition Network to and from State 2

Figure 3.11: Transition Network to and from State 3
Figure 3.12: Transition Network to and from State 4

Figure 3.13: Transition Network to and from State F
Next Act | In State
--- | ---
Initiate^I | S 1 2 3 4 F D
Continue^I | 1 1 4
Continue^R | 2 3
Repair^I | 1 1 1 4 1
Repair^R | 3 2 3 3 3
ReqRepair^I | 4 4 4 4
ReqRepair^R | 2 2 2 2 2
Ack^I | F 1^* F
Ack^R | F F^* F
ReqAck^I | 1 1
ReqAck^R | 3 3
Cancel^I | D D D D D
Cancel^R | 1 1 D

*repair request is ignored

Table 3.1: DU Transition Diagram

An utterance plays in a unit, whether a discourse unit has been completed, or just what it would take to complete one, only certain patterns of actions are allowed, as shown in Table 3.1. Impossible actions are represented in the table by blanks. For instance, a speaker cannot acknowledge his own immediately prior utterance. He may utter something (e.g. “ok”) which is often used to convey an acknowledgement, but this cannot be seen as an acknowledgement in this case. Often it will be seen as a request for acknowledge by the other party. Similarly, a speaker cannot continue an utterance begun by another agent. The speaker could produce an utterance which contains a syntactic continuation of, or conceptually related material to another utterance by another agent, but this would not be a continue act. Depending on context, it would be interpreted as either an acknowledgement (e.g., if one is just completing the other’s thought), a repair (if one is correcting to what should have been said), or an initiate of a new DU (if it provides new information).

If one is in a state and recognizes an impossible action by the other agent, there are two possibilities: the action interpretation is incorrect, or the other agent does not believe that the current DU is in the same state (through either not processing a previous utterance or interpreting its action type differently). Either way, this is a cue that repair is needed and should be undertaken. One also always has the option of initiating a new DU, and it may be the case that more than one DU is open at a time. If a DU is left in one of the non-final states, then its contents should not be seen as grounded.

As shown above, we can identify at least seven different possible states for a DU to be in. These states can be distinguished by their relevant context: what acts have been
performed and what is preferred to follow, as shown in Table 3.2. State S represents a DU that has not been initiated yet. State F represents one that has been grounded, though we can often add more to this unit, as in a further acknowledgement or some sort of repair. State D represents an abandoned DU, ungrounded and ungroundable. The other states represent DUs which still need one or more utterance acts to be grounded. State 1 represents the state in which all that is needed is an acknowledgement by the responder. This is also the state that results immediately after an initiation. However, the responder may also request a repair, in which case we need a repair by the initiator before the responder acknowledges; this is state 2. The responder may also repair directly (state 3), in which case the initiator needs to acknowledge this repair. Similarly the initiator may have problems with the responder’s utterance, and may request that the responder repair; this is state 4.

It is important to note that these states are states only of the DU itself. We are not claiming that aspects of discourse other than grounding could be modeled with a finite-state automaton. Also, these states do not refer to the “state” of a computational agent engaged in conversation. Any computational agent (e.g., that proposed in Chapter 5) will have to pay attention to more than just grounding, and even a grounding module will need to pay attention to multiple DUs, each of which will have an associated state.

This finite-state machine has been constructed by analyzing common sequences of utterances in the TRAINS corpus, guided by intuitions about possible continuers and what the current state of knowledge is. It serves as an alternative structural account of grounding to Clark and Schaefer’s contribution model. This network serves mainly as guide for interpretation, though it can also be an aid in utterance planning. It can be seen as part of discourse segmentation structure maintained as part of the context of a conversation. It can be a guide to recognizing which acts are possible or of highest probability, given the context of which state the conversation is currently in. It can also be a guide to production, channeling the possible next acts, and determining what more is needed to see things as grounded. It is still mainly a descriptive model; it says nothing about when a repair should be uttered, only what the state of the conversation is when one is uttered. We can evaluate the correctness of this model by checking to see how it

Table 3.2: Summary of Discourse Unit States

<table>
<thead>
<tr>
<th>State</th>
<th>Entering Act</th>
<th>Preferred Exiting Act</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>——</td>
<td>Initiate\textsuperscript{I}</td>
</tr>
<tr>
<td>1</td>
<td>Initiate\textsuperscript{I}</td>
<td>Ack\textsuperscript{R}</td>
</tr>
<tr>
<td>2</td>
<td>ReqRepair\textsuperscript{R}</td>
<td>Repair\textsuperscript{I}</td>
</tr>
<tr>
<td>3</td>
<td>Repair\textsuperscript{R}</td>
<td>Ack\textsuperscript{I}</td>
</tr>
<tr>
<td>4</td>
<td>ReqRepair\textsuperscript{I}</td>
<td>Repair\textsuperscript{R}</td>
</tr>
<tr>
<td>F</td>
<td>Ack\textsuperscript{(I,R)}</td>
<td>Initiate\textsuperscript{(R,R)} (next DU)</td>
</tr>
<tr>
<td>D</td>
<td>Cancel\textsuperscript{(I,R)}</td>
<td>Initiate\textsuperscript{(R,R)} (next DU)</td>
</tr>
</tbody>
</table>

\textsuperscript{3}Except in the obvious case after a ReqRepair (states 2 and 4).
would divide up a conversation, and whether it seems to handle acknowledgements and repairs correctly. We can also evaluate its utility for processing: whether it serves as a useful guide or not. The type of behavior it describes can also be analyzed in terms of the preconditions and effects of actions, as sketched in Section 3.3, but having an explicit model of the nature given here may serve to help repair interactions, and make processing more efficient.

### 3.3 A Cognitive Model of Grounding Act Processing

In order to model the grounding process, including the planning, production, and recognition of the grounding acts from the previous section, we must use several of the mentalistic notions described in Section 2.2, including beliefs and mutual beliefs, intentions, and obligations. In addition, we will assume the existence of several mental processes that can manipulate these attitudes as the agents engage in conversation. We will require an utterance planner to produce from communicative intentions, natural language utterances that can help realize those intentions in the current context. Once the utterance has been planned, we need a language producer to actually perform the utterances. Finally, we will need a language interpreter which can observe linguistic behavior and decide what actions were performed.

Conversation will generally start from intentions of an agent to have some content added to the common ground (mutually believed). Then the utterance planner will decide on utterances and the language producer will act in accordance with the decisions. The language interpreter will monitor the produced utterances to see if the actions are in accord with the intentions: interpreting the utterances by both the same agent and the other agent will also provide new information about the beliefs and intentions of the agents. Utterances may also introduce obligations, including discourse obligations, which involve further participation in the conversation.

Figure 3.14 gives a schematic of the types of cognitive states necessary for tracking grounding using the actions introduced above in Section 3.1. This figure represents the grounding process from the point of view of an agent X. The other agent that X is communicating with is Y. The boxes represent distinct mental attitudes, which are related by different inference processes. Performance of grounding acts can affect the contents of these attitudes, including the transfer information from one to another. Box 6 represents (X’s beliefs about) mutual beliefs of X and Y. Box 1 represents information that X intends be mutually believed. Box 3 represents (X’s beliefs about) Y’s beliefs about what X intends be mutually believed. Box 2 represents (X’s interpretations of) the contents of utterances that X makes in order to change Y’s beliefs and thereby bring about mutual beliefs. Box 4 represents (X’s interpretations of) the contents of Y’s utterances, and Box 5, (X’s beliefs about) Y’s intentions. Also shown are the discourse obligations of the agents. The arrows represent the main information flow that results from the performance of grounding acts.

The grounding process is started when one party or the other makes an utterance which initiates a discourse unit. X will decide to initiate a DU if there is something in
Box 1 which is not (either partially, other than Box 6, or completely) part of any of the other attitudes represented in this diagram, and the proper contextual factors apply (X has or can take the turn and there are no outstanding discourse obligations or other goals which are more immediate). X will invoke the utterance planner to come up with an utterance or sequence of utterances that will convey X's intention to Y. When X actually performs the (first) utterance, we will have in Box 2 the interpretation of that utterance. This will most likely be the same as was intended (if the utterance planner is good), but might not be, due to resource constraints on the utterance planning and language production processes. If, somehow, the interpretation of the utterance is different from what was intended, that will provide the basis for planning some sort of repair that can clarify or correct the previous utterance. In addition, it might prove desirable to split up the conveyance of content into several utterances, in which case the content of Box 2 will be only a part of the intended content in Box 1.

Starting with the interpretation of X's own utterance in Box 2, if the interpretation is a "content" act, such as initiate, continue, or repair, then the next step is to do plan recognition and inference using (X's beliefs about) Y's beliefs, to see what Y is likely to believe about X's intentions. The result of this plan recognition, including the act

---

4It might seem that this step is unnecessary, and that the utterance planner should have included this reasoning to produce the utterance in the first place. While, it is true, an ideal utterance planner would always calculate the precise effect of the utterance on the intended audience, and a perfect language production system would always produce exactly the planned utterance, our model will also cover the real-world case in which adequate planning is not always performed and execution errors are possible. Some of the stumbling blocks to perfect planning include complexity and time criticality, incomplete
interpretation and its implicatures, will be placed in Box 3. Now if Box 3 contains the same content as Box 1, X believes that her communication was adequate, and must wait for (or prompt with a reqAck) an acknowledgement from Y that Y correctly understood. If, on the other hand, the contents of Box 3 are not the same as those of Box 1, that is further motivation to make a subsequent utterance, which might involve more utterance planning. The subsequent utterance may come out as a repair, a continue, or even a new initiate, depending on the particular differences between actual intentions and the perceived intentions attributed to the other agent. These subsequent utterances would also be subject to the same processes of interpretation and inference leading back through Box 2 to Box 3.

When Y produces an Utterance, X’s interpretation of this utterance goes into Box 4. If there is some problem with the interpretation process, such as no plausible interpretation, or no evidence to choose between two or more possible interpretations, this will provide the basis for the utterance planner to come up with some sort of repair initiator, most probably a reqRepair, but perhaps (if contextual factors indicate what Y should have said) a repair.

Once X thinks that she understands Y’s utterance, what happens next depends on the actions that X thinks Y has performed. If Y has performed an ack, then the appropriate contents of Box 3 (i.e., the contents that Y acknowledged) are moved to Box 6 (MB). If the utterance is an initiate, continue, or repair, then X will do plan recognition and inference and put the results in Box 5. X can make the contents of Box 5 grounded by uttering an acknowledgement, moving these contents on to Box 6. If Y’s utterance is either a request for acknowledgement or a request for repair, this will give evidence for more inference to be performed on the contents of Boxes 3 and 5 (taking into account new inferences about Y’s beliefs arising from the observation of Y’s utterance), as well as adding discourse obligations for X to perform or respond to the requested action.

Table 3.3 summarizes the reasons for X to perform particular grounding acts, and shows the effects of performance. For each of X’s actions, after coming up with the intention to perform the action, X will first plan the utterance and then perform it, then interpret the actual utterance and place the results in Box 2. Further effects depend on the type of action, and are described in the third column.

Table 3.4 shows the effects of Y’s actions. For all of these actions, the interpretation of the utterance will start in Box 4. Acts by Y may also provide additional evidence about Y’s mental state, and thus present an opportunity for X to revise her beliefs about the contents of Boxes 3 and 5 from previous utterances. This may lead to an incentive for X to produce a third-turn repair (if the changes are to Box 3) or fourth-turn repair (if the changes are to Box 5).

---

information, and simultaneous action. It might be that more information is available to the reasoner after the utterance is produced than is present at the time the utterance is planned.
<table>
<thead>
<tr>
<th>Action</th>
<th>Reason</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continue</td>
<td>Item in [1], part but not all of item in [3]</td>
<td>Put contents of act in [3]</td>
</tr>
<tr>
<td>Repair</td>
<td>Either Item in [2] or [3] doesn’t match the appropriate item in [1] or Item in [4] is unclear (either no interpretation, no preferred interpretation, or interpretation doesn’t match expectations) but there is enough context to say what it should be</td>
<td>Delete indicated part of item, move replacement content to [3]</td>
</tr>
<tr>
<td>ReqRepair</td>
<td>Item in [4] is unclear (either no interpretation, no preferred interpretation, or interpretation doesn’t match expectations)</td>
<td>Add Discourse Obligation for Y to repair</td>
</tr>
<tr>
<td>ReqAck</td>
<td>Item in [3] matches item in [1], Y has passed up a chance to acknowledge</td>
<td>Add Discourse Obligation for Y to ack</td>
</tr>
</tbody>
</table>

Table 3.3: X’s actions

<table>
<thead>
<tr>
<th>Action</th>
<th>Effects</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiate</td>
<td>Put contents of act in [5]</td>
</tr>
<tr>
<td>Continue</td>
<td>Put contents of act in [5]</td>
</tr>
<tr>
<td>Repair</td>
<td>Delete indicated part of act in [3] or [5], move replacement content to [5]</td>
</tr>
<tr>
<td>ReqRepair</td>
<td>Add Discourse Obligation to repair</td>
</tr>
<tr>
<td>ReqAck</td>
<td>Add Discourse Obligation to ack; ReqRepair if unsure what Y wants acknowledged</td>
</tr>
</tbody>
</table>

Table 3.4: Y’s actions
3.4 Connecting the Cognitive and FA Sequence Models

Table 3.5 shows constraints on performance by X or Y of the grounding acts. The acts are superscripted with I or R, depending on whether the speaker is acting as the initiator or responder, as in Table 3.1. These constraints are all relative to the knowledge of X, as represented in Figure 3.14.

<table>
<thead>
<tr>
<th>Actions</th>
<th>Conditions for X</th>
<th>Conditions for Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initiate(^I)</td>
<td>Item in [1], not elsewhere</td>
<td>none</td>
</tr>
<tr>
<td>Continue(^I)</td>
<td>Part of item in [3], part in [1]</td>
<td>Item in [5]</td>
</tr>
<tr>
<td>ReqRepair(^I)</td>
<td>Item in [4] which is unclear</td>
<td>Item in [2] or [3]</td>
</tr>
<tr>
<td>ReqRepair(^R)</td>
<td>Item in [4] which is unclear, or item in [5] which doesn't seem right</td>
<td>Item in [2] or [3]</td>
</tr>
<tr>
<td>Ack(^I), Ack(^R)</td>
<td>Item in [5]</td>
<td>Item in [3]</td>
</tr>
</tbody>
</table>

Table 3.5: Constraints on Grounding Acts

Using this information, we can now account for the constraints on the distribution of grounding acts in a discourse unit shown in Table 3.1. In state S, there is nothing of the current Discourse Unit in any of the boxes (other than perhaps Box 1), so according to Table 3.5, the only act possible is an initiate. Also, an initiate act is not possible in any other state, because this DU has already been initiated (though, of course, either party may begin a new DU with a subsequent initiate act).

State 1 corresponds to there being something in Box 3 if X is the initiator, or Box 5 if Y is the initiator. From State 1, Ack\(^I\) is disallowed, because there is nothing in the appropriate box (Box 5 if X is initiator, Box 3 if Y is the initiator) for the act to acknowledge. Similarly, there is nothing for the initiator to request repair of. Continuations and repairs by the initiator will just add more to Box 3 if X is initiator or Box 5 if Y is the initiator. An acknowledgement by the responder will move items into Box 6.

State 2 corresponds to a point after a repair request by the responder. If X is the
initiator, then there is something in Box 3, and the repair request in Box 4. Also, X has a discourse obligation to make a repair. A continuation is precluded because, given the obligation, it would be seen as somehow addressing the request, and therefore a repair. If, somehow, the initiator's next utterance were seen as a continuation, it would be a signal that the initiator did not process the request. As in State 1, the initiator cannot acknowledge or confirm, because there is nothing in the proper box. The expected operation from State 2 is that the initiator will perform the requested repair, but there are a few other possibilities. The initiator might not be able to interpret the request, and may request a repair of the reqRepair, shifting the discourse obligation over to the responder, and putting the DU in State 4. The responder may realize that his request might not be interpreted correctly, and may repair it, remaining in State 2. He might make a different repair request, also remaining in State 2. The final possibility is that, on further reflection, the responder realizes the answer without the initiator having repaired. In this case the responder may acknowledge the original contribution, just as though the request had never been made. This takes us directly to State F, and removes the obligation to repair.

State 3 is reached when the responder has directly repaired the initiator's utterance. Here the responder is shifting from what the initiator intended for the DU to what the responder intends. In making the repair, an item has been placed in Box 3, when X is responder, or Box 5, when Y is the responder. The responder can be seen as shifting the roles and seizing the initiative. This state is thus a sort of mirror of State 1. The initiator can repair in return, seizing back the initiative and moving back to State 1. Also the responder can make a follow-up repair, adding more items to the appropriate box, but remaining in State 3. The initiator might not understand the repair, and may make a repair request, moving to State 4. The responder might have a problem with something else that the initiator said, and can "release the initiative" with a repair request, moving back to State 2. Also, the initiator can acknowledge the repair, moving the items to Box 6. The responder may no longer acknowledge his own repair, though he may request an acknowledgement, or even rescind the repair (e.g., "oh, sorry, you're right.").

State 4 is perhaps the most complicated state. It corresponds to the responder having a discourse obligation to repair. If we trace back the conditions on the acts, we see this can only happen after an original initiate by the initiator, some response by the responder, and then a repair request by the initiator. Thus there is something in each of Boxes 2–5 and the responder has an obligation to make a repair. From this state, the initiator may make a further repair request, or repair her previous request, remaining in State 4, the responder may repair, moving on to State 3, or the responder may request repair of the repair request, moving to State 2.

State F occurs when items have moved on to Box 6. Ideally, things are now grounded, and no further action is necessary. It is still possible, however, to reopen the discourse unit, as shown in the last column of table 3.1. A repair will put the new item in Box 3 if performed by X, and in Box 5 if performed by Y. A reqRepair or reqAck will produce

\[5\]Or more precisely what he thinks that the initiator should intend.
the appropriate discourse obligation. In addition, a follow-up acknowledgement will keep the DU in State F.

### 3.5 Deficiencies of the Finite Automaton Grounding Model

While the model given in Section 3.2 has the virtue of simplicity, a question arises as to how accurate this model is: can it cover a sufficient proportion of the grounding phenomena present in natural language conversations? Will the predictions made by the model as to the groundedness of content conform to the intuitions of participants? Will an agent using the model be able to participate naturally in conversation, or will the adherence to the model cause extra confusion or unnatural constraints?

There are a number of phenomena which, at first glance, this simple model does not seem to handle correctly. This section will explore some of those phenomena and consider whether some more complex model might be more appropriate.

**Multiple acknowledgements** While the finite-state model does not preclude multiple acknowledgements as in (8)^6^, a fragment from the example conversation in Section 4.3, it does not provide any reason for their occurrence.

<table>
<thead>
<tr>
<th>UU Act</th>
<th>UU#</th>
<th>Speaker: Utterance</th>
</tr>
</thead>
<tbody>
<tr>
<td>cont₉</td>
<td>9.2</td>
<td>M: move the engine</td>
</tr>
<tr>
<td>cont₉</td>
<td>9.3</td>
<td>at Avon</td>
</tr>
<tr>
<td>repair₉</td>
<td>9.4</td>
<td>engine E</td>
</tr>
<tr>
<td>cont₉</td>
<td>9.5</td>
<td>to</td>
</tr>
<tr>
<td>repair₉</td>
<td>10.1</td>
<td>S: engine E₁</td>
</tr>
<tr>
<td>ack₉</td>
<td>11.1</td>
<td>M: E₁</td>
</tr>
<tr>
<td>ack₉</td>
<td>12.1</td>
<td>S: okay</td>
</tr>
</tbody>
</table>

DU #9 is considered grounded after utterance 11.1. Utterance 12.1 thus does not do any additional grounding work according to this model. So why was it uttered? One possibility is that these extra acknowledgements have to do with Clark and Schaefer’s *Grounding Criterion* and *Strength of Evidence Principle*: S felt that the DU was not grounded to a degree sufficient for the current purposes, and so a further acknowledgement was added. Unfortunately, making use of this principle will significantly complicate the analysis, since it requires a notion of degrees of groundedness. If we equate grounding with mutual belief, then we have to have a notion of a degree of confidence in mutual belief rather than a boolean belief operator. Immediately, the question arises as to how many degrees there should be and which degrees will result from particular utterance types. While this is a very interesting area for further research, it is beyond the scope of the present work.

---

^6^The notation used here is explained in Section 4.3.2. The left column lists the grounding act type, subscripted with the DU it is part of. The middle column shows the utterance number, and the right column, the speaker and the utterance.
**center embedding** In actual conversation, it seems that we can acknowledge a part of the ungrounded content, whereas the discourse unit model in Section 3.2 only allows the whole discourse unit to be acknowledged at once. Some of the simple cases seem to follow a “stack-based” structure, in which an *insertion sequence* is grounded as part of the grounding of the top-level, as in Clark and Shaefer’s contribution model. For these utterances, we’d thus have a structure such as (9),

\[
(9) \begin{align*}
A &: \text{utterance 1} \\
B &: \text{utterance 2} \\
A &: \text{ack utterance 2} \\
B &: \text{ack utterance 1}
\end{align*}
\]

Some of these sequences can easily be accommodated if utterance 2 starts a new discourse unit. However, if utterance 2 is a repair, and clearly seems to be part of the same unit as 1, things are more problematic. It might seem that a push-down automaton such as that presented in Section 3.1 might be a more appropriate computational model, reflecting this nesting structure. In fact, the example repair fragment presented in example (8), above might well be analyzed as having this structure. Such a model would have second-person repairs and repair requests entering a new *sub-DU* on the stack, which would have to be grounded before the main unit could be.

**partial acknowledgement** There are also other sequences in which an utterance seems to acknowledge part but not all of the current content, but there is no clear prior separation within the DU of the parts that become grounded from the parts that remain ungrounded. The sequence in (10) illustrates this.

\[
(10) \begin{align*}
56.2 & \ S: \text{and then which route do you wanna take} \\
57.1 & \ M: \text{took to Avon} \\
58.1 & \ S: \text{yeah} \\
59.1 & \ M: \text{um} \\
59.2 & \ : \text{the shortest route}
\end{align*}
\]

The repair in 57.1 shows an understanding of part of the previous question, but not quite enough to provide an answer. Even more problematic is the sequence in (11).

---

7Taken from Dialogue 91-6.2 in [Gross et al., 1993].

8Taken from Dialogue 91-8.2 in [Gross et al., 1993].
(11) 87.1  M: but if we do that
  87.2  : then we won't be able to m /
  87.3  : we won't make it
  87.4  : the other/ the other constraint won't work either
  88.1  S: well
  88.2  : you mean
  88.3  : t /
  88.4  : if / if we do which

The “well” in 88.1 signals acknowledgement (though lack of acceptance of the claim), then 88.4 is a repair request, asking for a more explicit referent for “that” in 87.1.

These sorts of repairs indicate an understanding of the explicit material, but a lack of sufficient certainty about the some of the implicit contents, such as the destination of the route in example (10), and the precise plan that is the referent of “that” in 87.1 in example (11). While we might be inclined to just treat these clarifications as separate DUs expressing the more detailed information, this would be somewhat problematic, because in the absence of such a clarification, the initiator of the main DU would assume the implicit content was understood after a simple acknowledgement. It would appear, then, that the content of a DU would be sensitive to the type of acknowledgement performed.

3.5.1 Alternative Grounding Models

As a first alternative, consider the model in Section 3.1, in which other-repair, instead of moving to a new state in the DU, actually pushes a new sub-DU onto the stack. Moving to such a push-down repair model will have several implications on the general grounding model.

First off, what does one say about the groundedness of a popped sub-unit? For example, for the sequence in (12), what would the push-down model claim about whether the material in the repair is grounded? For the finite-state model in Section 3.2, the entire DU will be considered grounded at this point. For a push-down model, there are two options: that the whole DU is ungrounded, or that the repair sub-DU is grounded, but other parts of the utterance are ungrounded.

(12)  A init
       B repair
       A ack

An empirical question is whether sequences like example (12) occur by themselves and are considered grounded. This is a difficult question to settle, since direct second-person repair is fairly infrequent, and even when one does occur without an explicit
second acknowledgement by the repair producer, the next utterance by the repair producer can usually be seen as an implicit acknowledgement. Even if this question were settled on the side of sequences like example (12) being ungrounded, this would not invalidate a finite-state approach, but would merely mean, for example, that the transition from State 3 given an Ack should be to state 1, rather than State F. If on the other hand, sequences like example (12) are grounded, then this would be hard to account for in the push-down model.

The big question is whether one actually needs the full flexibility and constraints of a context-free language, or whether a regular language will be sufficient. This question is made more difficult by the fact that human processing limitations will make it unlikely that more than a couple of levels of nesting could occur even with a push-down model – beyond this, the participants would get confused and restart again at the top level. This seems to suggest that a finite-state model will be adequate, though perhaps one might want a few extra states to represent another level or two of nesting.

A push-down model with popped sub-DUs seen as grounded makes an even stronger case that some sort of partial acknowledgement must be happening: often the repaired part does not make much sense isolated from the context of the main DU. This means that if the repaired part is grounded, at least some of the rest of the content must be grounded, perhaps by the initial other repair itself. If we can, in fact, accomplish this partial acknowledgement, then perhaps we can also do it within a more restrictive model as well.

Another way to try to fix the grounding model to accommodate these cases of partial acknowledgement would be to add a notion of composite acts, acts which, given an initial DU would split it into multiple DUs, each consisting of a part of the content of the original, and thus the new utterance will function as different acts for the different resulting DUs. Thus, an utterance might acknowledge part of an open DU while repairing another part. For instance, we might crudely represent the relevant state of the conversation in example (11) after utterance 87.4 as (13).

\[
\begin{array}{c|c|c}
\text{(13) DU 1:} & \text{state:} & 1 \\
& \text{content:} & \ldots \\
& \text{Claim(M,S, IF DO({M,S},THAT)} & \\
& \quad \neg \text{MakeIt({M,S})} & \\
& \text{THAT=} & (\text{the x (Plan x) } \wedge \ldots) \\
& & \ldots \\
\end{array}
\]

After 88.4, though we might have something more like (14), where M now has a discourse obligation to repair the referent of THAT. Thus, S’s actions result in splitting DU 1 here into two parts, acting as an acknowledgement for one part, and as a repair request for the other.
A grounding theory such as this is more complex, since it now requires one to treat DUs as non-atomic, but it allows a very fine-grained analysis of the grounding process. For each bit of content represented in a particular DU, as each utterance is processed, one must decide how the utterance affects the particular bit of content. If an utterance affects different parts of the content differently, then one must split the new DU appropriately. If multiple bits are affected similarly, and the same future action might also affect these bits similarly, then there is good reason to keep them in the same unit. This issue is taken up further at the end of Chapter 7, in which a theory of grounding based on plan execution is presented, allowing formalization of both the grounding acts presented earlier in this chapter as well as the fine-grained “splitting” acts considered in this section.

One question relevant to the decision on whether to use a push-down model is whether some parts cannot be grounded until other parts are. If there are such constraints, then this would be a good reason to have a pushdown model. If there are not such constraints, then there really isn’t a good reason for such a model.

3.6 Chapter Summary

In this chapter, we have presented a protocol for grounding in conversation that is based on performing grounding acts to build up grounded content in the form of discourse units. This protocol solves the problems with Clark and Shaefer’s Contribution model, discussed in Section 2.6. The infinite regression of acknowledgements is avoided by allowing some autonomous acknowledgement acts that do not require further acknowledgement. The protocol allows an agent involved in the conversation to judge whether a unit is grounded and if not what else is required. In addition, the model is finite-state rather than recursive, allowing for a simpler theory and easier on-line processing. A mentalistic model of grounding was also presented that shows the effects on the mental state of the participating agents of processing grounding acts and provides reasons for producing grounding acts, based on linguistic goals. This model is also compared to the protocol, and helps explain why the details are as they are. Finally, some potential problem areas for the finite-state grounding protocol are considered.

Chapter 4 shows how grounding fits in as just one of several levels of action that occur in a conversation. Chapter 5 extends the mentalistic model in this chapter to
a fuller theory of dialogue management, showing how grounding can be naturally fit into a task-oriented natural language conversation system. Chapter 6 describes an implementation of the ideas in the previous chapters as part of the TRAINS system.
4 An Overview of Conversation Acts

Chapter 3 presented the foundation for a computational theory of grounding, deciding for sequences of grounding acts which material was grounded. A cognitive model of grounding was also presented, which showed the rationale for performing grounding acts as well as their effects. There are still several pieces missing from a complete grounding theory, however. First, while the content of grounding acts was mentioned, this content, which is the aim of grounding, was left unspecified. In addition, while the cognitive model laid out the rationale for performing one act instead of another depending on the context and goals, it had nothing to say about when to perform an act and when to wait for the other conversant to act. Finally, something must be said about how to recognize the performance of particular acts in a given context. Examination of these issues must involve issues beyond just the grounding phenomena, including issues of turn-taking and utterance content.

This chapter describes the theory of Conversation Acts\(^1\), a multi-stratal theory of action in conversation, which includes a level of Grounding Acts. Section 4.1 argues for the need of a multi-stratal theory of action. Section 4.2 describes a taxonomy of conversation acts. Section 4.3 presents a short conversation in the TRAINS domain with annotated conversation acts. Section 4.4 describes some of the methods for recognizing conversation acts. Finally, Section 4.5 compares the theory of conversation acts to other multi-stratal action theories.

4.1 The Need for Multiple Levels of Action in Conversation

The models of speech acts described in Section 2.1 are very useful tools for analyzing the relationships between sentential utterances and the mental states of the speaker and hearer of such utterances. We wish to extend this work to be able to model more of the types of coordinated activity that takes place between agents in a conversation than can be modelled using just these sentential-level speech acts.

In order to accomplish this, we have to relax the following assumptions which have usually been made within those theories of speech acts:

\(^1\) Most of this chapter appeared previously in [Traum and Hinkelmann, 1992].
1. Utterances are heard and understood correctly by the listener as they are uttered. Moreover, it is mutually expected by both participants that this will be the case.

2. Speech acts are single-agent plans executed by the speaker. The listener is only passively present.

3. Each utterance encodes a single speech act.

Each of these assumptions is too strong to be able to handle many of the types of conversations people actually have, as described in Sections 2.3 and 2.5:

1. Not only are utterances often misunderstood, conversation is structured in such a way as to take account of this phenomenon and allow for repair. Rather than just assuming that an utterance has been understood as soon as it has been said, there is a grounding process which builds up to this mutual understanding by including actions by both conversants. The assumption of mutual understanding is not made until some positive evidence is given by the listener (an acknowledgement) that the listener has understood. Some acknowledgements are made with explicit utterances (so-called backchannel responses such as “okay”, “right”, “uh huh”), some by continuing with a next relevant response (e.g. a second part of an adjacency pair such as an answer to a question), and some by visual cues, such as head nodding or continued eye contact. If some sort of evidence is not given, however, the speaker will assume that he has not made himself clear, and either try to repair, or request some kind of acknowledgement (e.g. “did you get that?”).

2. Since the traditional speech acts (e.g. Inform, Request) have mutual belief as an effect, they require at least an initial presentation by one agent and some form of acknowledgement by another agent. They are thus are inherently multi-agent actions. Rather than being formalized in a classical single-agent logic, they must be part of a framework which includes multiple agents.

3. Each utterance can encode parts of several different acts. It can be a presentation part of one act as well as the acknowledgement part of another act. It can also contain turn-taking acts, and be a part of other relationships relating to larger-scale discourse structures.

Additionally, there are several different types of action being performed in carrying on a conversation. In addition to performing the traditional propositional-level speech acts, conversants are structuring their utterances into coherent segments and also engaging in the turn-taking and grounding processes. While it would be possible to construct individual descriptions of the totality of action being performed, this would miss the generalization that action at these different levels are largely orthogonal to each other. By separating out the action at the different levels, the action descriptions at each individual level can remain more simple.
4.2 Levels of Conversation Acts

We distinguish four levels of action necessary for expressing the content and maintaining the coherence of conversation, as shown in Table 4.1. Action attempts at any of these levels can be signaled directly by surface features of the discourse, although usually a combination of surface features and context will be necessary to disambiguate particular acts. In Table 4.1 the levels from top to bottom are typically realized by larger and larger chunks of conversation, from turn-taking acts, usually realized sub-lexically to grounding acts which are realized within a single utterance unit (UU), to core speech acts which are only completed at the level of a completed discourse unit (DU), to argumentation acts which can span whole conversations. The table also shows some representative acts for each class.

<table>
<thead>
<tr>
<th>Discourse Level</th>
<th>Act Type</th>
<th>Sample Acts</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub UU</td>
<td>Turn-taking</td>
<td>take-turn, keep-turn, release-turn, assign-turn</td>
</tr>
<tr>
<td>UU</td>
<td>Grounding</td>
<td>Initiate, Continue, Ack, Repair, ReqRepair, ReqAck, Cancel</td>
</tr>
<tr>
<td>DU</td>
<td>Core Speech Acts</td>
<td>Inform, YNQ, Check, Eval Suggest, Request, Accept, Reject,</td>
</tr>
<tr>
<td>Multiple DUs</td>
<td>Argumentation</td>
<td>Elaborate, Summarize, Clarify Q&amp;A Convince Find-Plan</td>
</tr>
</tbody>
</table>

Table 4.1: Conversation Act Types

It is important to note that according to the terminology of Halliday [1961] these classes are levels of language description, and not ranks. That is, the distinction between these classes is more like that between phonology and syntax rather than that between a word and a phrase. For example, there is no grammar which will build up a grounding act as an ordered collection of turn-taking acts.

4.2.1 The Core Speech Acts: DU Acts

In adapting speech act to spoken conversation, we maintain the traditional speech acts (described in Section 2.1) such as Inform, Request, and Promise, referring to them as core speech acts. To model the multi-utterance exchanges necessary for mutual understanding of core speech acts, we posit a level of structure called a Discourse Unit (DU). As described in Chapter 3, a DU consists of an initial presentation, and as many subsequent utterances by each party as are needed to make the act mutually understood, or grounded. The initial presentation is best considered a core speech act attempt, which is not fully realized until its DU is grounded. A minimal DU contains an initial presentation and an acknowledgement (which may be implicit in the next presentation by another speaker). However, it may also include any repairs or continuations that
are needed to realize the act. A discourse unit corresponds more or less to a top-level *Contribution*, in the terminology of [Clark and Schaefer, 1989] (see Section 2.5). The structure of discourse units is described more fully in Chapter 3.

Some of the core speech acts occurring in the example conversation in Section 4.3 and their approximate meanings are given below. This is not meant as an exhaustive list of core speech act types. As in Chapter 3, *initiator* is the agent who makes the initial presentation (often called *speaker* in traditional single agent accounts, e.g., [Allen, 1983b]), and *responder* is the other agent (often called *hearer* in previous accounts) who must acknowledge the presentation for the act to have its expected effect.

**inform** Initiator presents responder with new information in an attempt to add a new mutual belief.

**ynq** Initiator asks responder to provide information that initiator is missing but suspects that responder may know; imposes a discourse obligation on responder to respond.

**check** Like a **ynq**, but initiator already suspects the answer; initiator wants to move the proposition in question from individual to mutual belief, or bring the belief into working memory.

**eval** An evaluation by the initiator of the "value" of some (physical or intentional) object.

**suggest** Initiator proposes a new item as part of a plan.

**request** Like a **suggest**, but also imposes a discourse obligation to respond.

**accept** Initiator agrees to a proposal by responder.

**reject** Initiator rejects a proposal by responder.

### 4.2.2 Argumentation Acts

Higher-level discourse acts may be built out of combinations of core speech acts. For instance, an **inform** act may be used in order to summarize, clarify, or elaborate prior conversation. A very common argumentation act is the Q&A pair, used for gaining information. Also, a combination of informs and questions can be used to convince another agent of something. An agent may even use a whole series of acts in order to build a plan, such as the top-level goal for the conversations in the TRAINS domain [Allen and Schubert, 1991]. The kinds of actions generally referred to as *rhetorical relations* [Mann and Thompson, 1987] take place at this level, as do *adjacency pairs* [Schegloff and Sacks, 1973] and many of the actions signaled by cue phrases.
4.2.3 Grounding Acts: UU Acts

An utterance unit (UU) is defined as continuous speech by the same speaker, punctuated by prosodic boundaries (including pauses of significant length and boundary tones). Each utterance corresponds to one grounding act for each DU it is a part of. An utterance unit may also contain one or more turn-taking acts (see below). The grounding acts (introduced in Chapter 3) are defined informally below.

**Initiate** An initial utterance component of a Discourse unit – traditionally this utterance alone has been considered sufficient to accomplish the core speech act. An initiate usually corresponds to the (first utterance in the) presentation phase of a top level Contribution in [Clark and Schaefer, 1989].

**Continue** A continuation of a previous act performed by the same speaker. A continue is expressed in a separate utterance unit, but is syntactically and conceptually part of the same act.

**Ack** An acknowledgement claiming or demonstrating understanding of a previous utterance. It may be either a repetition or paraphrase of all or part of the utterance, an explicit signal of comprehension such as “ok” or “uh huh”, or an implicit signaling of understanding, such as by proceeding with the initiation of a new DU which would naturally follow the current one in the lowest level argumentation act. Typical cases of implicit acknowledgement are answers to questions. Acknowledgements are also referred to by some as confirmations (e.g. [Cohen and Levesque, 1991a]) or acceptances (e.g. [Clark and Schaefer, 1989]). We prefer the term acknowledgement as unambiguously signaling understanding, reserving the term acceptance for a core speech act signaling agreement with a proposed domain plan.

**Repair** Changes the content of the current DU. This may be either a correction of previously uttered material, or the addition of omitted material which will change the interpretation of the speaker’s intention. A repair can change either the content or core speech act type of acts in the current DU (e.g. a tag question can change an Inform to a YNQ). Repair actions should not be confused with domain clarifications, e.g. CORRECT-PLAN and other members of the Clarification Class of Discourse Plans from [Litman and Allen, 1990]. Repairs are concerned merely with the grounding of content. Domain clarifications, which modify grounded content, are argumentation acts.

**ReqRepair** A request for a repair by the other party. This is roughly equivalent to a next turn repair initiator [Schegloff et al., 1977]. Often a reqRepair can be distinguished from a repair or ack only by intonation. A reqRepair invokes a discourse obligation on the listener to respond with either the requested repair, or an explicit refusal or postponement (e.g. a followup request).

**ReqAck** Attempt to get the other agent to acknowledge the previous utterance. This invokes a discourse obligation on the listener to respond with either the requested
acknowledgement, or an explicit refusal or postponement (e.g., a followup repair or repair request).

**Cancel** Closes off the current DU as ungrounded. Rather than repairing the current DU, a cancel abandons it; the underlying intention, if it is still held, must be expressed in a new DU.

Chapter 3 shows in detail how sequences of grounding acts can be used to build complete DUs, as well as giving motivations for performing particular acts, and their effects on the conversational state and mental states of the participants.

### 4.2.4 Turn-taking Acts: Sub UU Acts

We posit a series of low-level acts to model the turn-taking process [Sacks *et al.*, 1974; Ostrom, 1983]. The basic acts are **keep-turn**, **release-turn** (with a sub-variant, **assign-turn**) and **take-turn**.

There may be several turn-taking acts in a single utterance. The start of an utterance can be a take-turn action (if another party initially had the turn), the main part of the utterance generally keeps the turn, and the end might release it (or keep it to follow up with another utterance). Conversants can attempt these acts by any of several common speech patterns, ranging from propositional (e.g., “let me say something”) to lexical to sub-lexical. Many turn-taking acts are signaled with different intonation patterns and pauses. Although a conversant can attempt a turn-taking action at any time, it will be a matter of negotiation as to whether the attempt succeeds. Conversational participants may engage in a “floor battle” where one tries to keep the turn while another tries to take it. Participants may also use plan recognition on seeing certain kinds of behavior to determine that the other party is attempting to perform a particular act and, if cooperative, may then facilitate it (e.g., refraining from taking a turn when signaled that another wants to keep it, or releasing when another wants to take the turn).

Any instance of starting to talk can be seen as a take-turn attempt. We say that this attempt has succeeded when no one else talks at the same time (and attention is given to the speaker). It may be the case that someone else has the turn when the take-turn attempt is made. In this case, if the other party stops speaking, the attempt has been successful. If the new speaker stops shortly after starting while the other party continues, we say that the take-turn action has failed and a keep-turn action by the other party has succeeded. If both parties continue to talk, then neither has the turn, and both actions fail.

Similarly, any instance of continuing to talk can be seen as keeping the turn. Any signals that are part of the speech stream may be seen as keep-turn actions; certain sound patterns, such as “uhh”, seem to carry no semantic content beyond keeping the turn. Silent pauses are opportunities for anyone to take the turn. “Filling” pauses with such utterances as “uhh” can signal desire to keep the turn through what might otherwise be seen as a release-turn. Certain pauses are marked by context (e.g., a previous topic introduction or request) as to who has the turn. Even here, an excessive pause can open up the possibility of a take-turn action by another conversant.