Abstract
In this paper I will describe some work towards developing a reliable coding scheme for the purpose of representing relationships between moves in dialogues. The purpose of this coding scheme is to represent the interactional structure of negotiation discourse, particularly negotiations in which multiple points of negotiation are managed at once, so that it is possible to study correspondences between this structure and patterns of anaphora and referring expressions. I will discuss the results of an inter-coder reliability test which indicates that within one week totally inexperienced coders can make considerable progress towards learning how to apply this coding scheme consistently.

Introduction
In this paper I will describe some work toward developing a reliable coding scheme to provide a framework in which theories about discourse structure can be tested. This is a difficult task because even the most precisely stated theories about discourse structure are built on top of largely subjective intuitions. I will discuss my coding scheme with a built in mechanism for evaluating the correctness of an analysis. I will also discuss the results of an inter-coder reliability test carried out in order to determine the extent to which this coding scheme can be learned and applied consistently by a group of coders.

Bakeman and Gottman(86) discuss the advantages to developing reliable coding schemes for conducting analyses of sequential behavior. They explain that reliable coding results indicate that others observing the same data that we do would tend to see the same sorts of phenomena that we see and that our observations are not just products of our desire to see our theory validated. This makes it possible for our results to be reproduced and for other researchers to carry on our research, testing our predictions in different domains or on different types of discourse. Developing a reliable coding scheme insulates us to some extent from our own biases. However, it is also true that in developing a coding scheme we build in a particular bias which filters what we observe and therefore limits what conclusions we can draw from our analyses. This limitation must always be kept in mind. But it is also true that without any bias we would have no basis for drawing any conclusions at all.

With this coding scheme I have carved out a finite set of discrete relationships which I intend to identify. The types of relationships which I am looking for are moves in a negotiation. My goal is to use this coding scheme to analyze the structure of negotiation dialogues in which multiple points of negotiation are being managed in parallel. Below is an example of the type of dialogue I am analyzing.

S1: 1. When can you meet next week?
S2: 2. Tuesday afternoon looks good.
     3. I could do it Wednesday morning too.
S1: 4. Tuesday I have class from 12:00-1:30.
     5. But the other day looks good.

In this scheduling dialogue, two possible meeting times are negotiated in parallel. By representing the relationships between individual moves in the dialogue, it is possible to represent each of the parallel negotiations as a separate thread made up of a sequence of moves, each one contributing the result of the previous one. In this way, I characterize negotiation discourse as a tapestry of multiple interwoven threads. The question I would like to answer is whether the parallelism which exists on the intentional level in these dialogues exists also on the attentional level which makes predictions about types of referring expression which are likely to occur in particular contexts.

My coding scheme was inspired by Edmondson's(81) dissertation research in which he develops a method for analyzing conflict resolution dialogues. What is different between Edmondson's coding scheme and others who have studied sequences of speech acts (Searle 1969; Searle 1975) and how speech acts relate to one another (Searle 1992; Klammer 1973; Labov & Fanshel 1977; Schegloff & Sacks 1973; Sinclair & Coulthard 1975) is that he has separated the notion of illocutions from that of interactional structure which others have left
conflated. He argues that the concept of a Promise, for example, embodies something of the sequential placement of a speech act in a dialogue in addition the the intentions which are expressed. In his formulation, the illocution is an action which communicates the beliefs, desires, and intentions communicated by a sentence without reference to what function this action plays in the discourse. The interactional component represents the relationship between this action and the surrounding discourse, how this sentence works together with surrounding sentences in order to produce conversational outcomes which represent what has been accomplished by the dialogue. Making this distinction allows us to make generalizations about how moves in a dialogue interact with one another without having to try to sequence the illocutions which make up those moves. For a fuller discussion of why this distinction is important, see (Edmondson 1981).

The Coding Scheme

S1: 1. We should meet soon to discuss the plans for our paper.
    2. When can you meet next week?
S2: 3. Monday afternoon looks open.
    4. Or I could certainly do it on Thursday morning also.
    5. I'm busy on all of the other days.
S1: 6. Monday I have a class from 12:00-1:30.
    7. But Thursday sounds acceptable.
    8. How about 9:30?

Figure 1: Example Analysis

In Edmondson's coding scheme, conversational outcomes were said to be derived from the interactional structure, but these outcomes were never formalized, nor was the process of deriving them from the interactional structure. In my coding scheme, I have defined what these outcomes are and how they are derived in order to make it possible to determine whether any given analysis for a dialogue is correct for a given interpretation. Also, rather than assign each sentence a label indicating how it fits into the interactional structure as Edmondson did, I assign these labels to relationships between conversational outcomes, making it possible for a sentence to participate in multiple relations in the interactional structure.

Because scheduling dialogues are what I am applying my coding scheme to, I consider conversational outcomes to be partial plans for a meeting which are either positive, indicating that the speaker is in favor of this partial plan, or negative, indicating that the speaker is not in favor of this partial plan. In other task-related domains, conversational outcomes could be partial plans for other types of actions. At any given point in the analysis process, the coder can check the validity of the analysis by making sure the outcomes which are open correspond to the smallest full set of mutually exclusive partial plans open for negotiation at that point in the dialogue according to the coder's understanding of the dialogue.

The unit of analysis in my coding scheme is called a pseudo-sentence. Because I am working with spoken discourse, it is difficult to segment the discourse into units by purely syntactic means. Although pseudo-sentences are not always syntactic sentences strictly speaking, pseudo-sentence breaks roughly correspond to clause boundaries. Each pseudo-sentence can potentially be labeled with a conversation act and produce a local conversational outcome. A local outcome may have implications regarding the status of one or more previously produced conversational outcomes, giving rise to the interactional structure. When a conversational outcome interacts with another conversational outcome, producing a new conversational outcome, a conversational move is said to have taken place.

Each pseudo-sentence with propositional content expressing an attitude towards some partial plan for scheduling a meeting expresses a conversation act which produces a local conversational outcome. For example, the pseudo-sentence "The twenty eighth is bad for me," produces a negative local conversational outcome about the partial plan for meeting on the twenty eighth. In my coding scheme, the conversation act which produces a negative local conversational outcome is called prohibit whereas the one which produces a positive one is called license. Sentences such as "No," which have no propositional content produce no local outcome and are assigned the conversation act neutral. They are purely interactional and produce an outcome only through the interactional structure.

Input/Output

Support/Supported
Primed/Prime
Restrict/Restricted
Specify/Specified
Satisfy/Satisfied
Reject

Figure 2: Interactional Relations

The procedure followed by the coders contains four

1In more abstract domains, conversational outcomes may represent positions on issues being discussed.
2I am calling illocutions in Edmondson's sense conversation acts.
steps. The first step is to segment the dialogue into pseudo-sentence units and number the units. Then for each unit, the coders mark which of the possible local outcomes, if any, is produced from this unit in isolation. Next they label this unit with the conversation act which this unit expresses. Finally, they derive the interactional structure produced by this unit in relation to its context. These last three steps are repeated for each unit. See Figure 1 for an example.

The full set of moves I have included in my coding scheme are included in Figure 2 and explained below. I will be making reference to the example in Figure 1.

The Support Move

Supporting units clarify or otherwise elaborate on what has already been said by the same speaker within the same turn as part of a move in the interactional structure. Support is not so much a move as it is a way of linking parts of a move together. In other words, if a single move is expressed over multiple pseudo-sentence units, the outcomes from each of these units are linked together in a chain of Support relations. The purpose of this coding scheme is to represent the interactions between moves, so this part is left very general. The content of the supporting outcome is added to the content of the supported outcome, producing a new outcome. Generally, if two outcomes produced by pseudo-sentence units in the same turn relate to one another, they relate to one another in a Support relationship. The only exception is in the case of repairs where a speaker retracts part of what was said because it was not intended.

Figure 3: The Support Relation

Figure 3 displays the interactional structure after sentence 2. Sentences 1 and 2 together constitute a move for proposing a meeting in the next week to discuss a paper. The outcomes are displayed in Figure 4. Outcome 1-a is the local outcome of sentence 1. It specifies that the speaker is expressing a positive attitude towards meeting to discuss plans for a paper. This outcome is supported by outcome 2-a which is the local outcome of sentence 2. The result of this Support relationship is outcome 2-b. This outcome specifies that the speaker has expressed a positive attitude towards a meeting with the combined features from 1-a and 2-a.

The Prime Move

A Prime is a move which requests a move of a particular type. For example, “When are you free next month?” primes suggestions for potential meeting times. And “What do you think?” after a list of suggestions primes responses to those suggestions. If a unit produces an outcome which is understood as having been primed by a previous unit, a new outcome is produced with the restrictions from the priming unit included in the new outcome along with the restrictions from the primed one.

Figure 4: Outcomes 1-a, 2-a, and 2-b

Figure 5 displays the interactional structure for the whole dialogue in Figure 1. Figure 6 displays outcomes
3-a, 3-b, 4-a, and 4-b. Sentence 2 acts as a *Prime* for suggestions for meeting times. Sentences 3 and 4 are both suggestions for meeting times which were primed. Since Outcome 2-b is the most specific outcome resulting from this prime, this will be the outcome which will play this priming role in the interactional structure. Outcome 3-a is the local outcome from sentence 3 which specifies that the speaker is expressing a positive attitude towards a meeting on Monday afternoon. The result of this outcome participating in the *Prime* move with outcome 2-b is outcome 3-b which specifies that as a result of the interaction between the priming outcome and the primed outcome, the speaker is expressing a positive attitude towards a meeting next week, specifically on Monday afternoon, to discuss the plans for the paper. In a very similar manner, outcome 4-a participates in this relationship with outcome 2-b producing outcome 4-b. Outcomes 3-b and 4-b are now the most specific current outcomes in the interactional structure. They constitute the beginnings of two parallel threads in the dialogue, one a negotiation over meeting on Monday afternoon, and the other about meeting on Thursday.

**The Restrict Move**

A *Restrict* is a move which attempts to modify a

**Figure 5: Interactional Structure**

**Figure 7: Outcomes 5-a and 5-b**

previous outcome by removing some part or all of what was specified there. If the speaker produces a pseudo-sentence which expresses a *prohibit* and therefore attempts to retract part of an open partial plan, this would act as a *Restrict* in relation to that outcome. The resulting outcome is identical to the outcome which is restricted except for the removal of the specified portion and the addition of a restriction which expresses the proposition “not specified part”.

The local outcome which is produced by sentence 5 specifies that the speaker has expressed a negative atti-
OUTCOME: 6-a
START TIME: Monday from 12:00-1:30
ATTITUDE: negative

OUTCOME: 6-b
START TIME: soon, next week,
Monday afternoon, not from 12:00-1:30
TOPICS: plan for the paper
ATTITUDE: positive

Figure 8: Outcomes 6-a and 6-b

The Specify Move

OUTCOME: 7-a
START TIME: Thursday
ATTITUDE: positive

Figure 9: Outcome 7-a

A Specify is an move which attempts to append something to a previous outcome. So if Monday has been suggested and the other speaker responds with “Monday I can meet you in the morning”, it would act as a Specify. It accepts part of what has been suggested, but it adds a new restriction. The resulting outcome is identical to the previous outcome which is made more specific except for having been modified in the specified way. Note that this is very similar to the previously discussed Prime move. The difference is that a Prime occurs as a first move in a sequence and can be seen as initiating that sequence where a Specify occurs later in the sequence and can be seen as a response to what has already been contributed to that sequence.

Sentences 7 and 8 fit together into one move similarly to sentences 1 and 2. Outcome 8-a (see Figure 10) participates in a Support relation with outcome 7-a (see Figure 9) producing 8-b (see figure 10) which expresses a positive attitude towards meeting on Thursday at 9:30. This accepts part of outcome 4-b which expressed a positive attitude towards meeting on Thursday morning. So outcome 8-b participates in a Specify move with outcome 4-b producing outcome 8-c (see figure 10), expressing a positive attitude towards meeting next week, specifically on Thursday morning, more specifically at 9:30, in order to discuss the plans for the paper. This continues the second parallel thread.

The Satisfy Move

OUTCOME: 9-a
ATTITUDE: neutral

OUTCOME: 9-b
START TIME: soon, next week,
Thursday morning, 9:30
TOPICS: plan for the paper
ATTITUDE: positive

Figure 11: Outcomes 9-a, 9-b

A Satisfy is an action which brings about a possible closing point in part of a negotiation, or a transition relevance point (Sacks, Schegloff, & Jefferson 1974). Essentially, something which acts as a Satisfy in relation to another outcome accepts it without modification. If a suggestion for Monday is open and someone says, “Monday I’m free.” This would act as a Satisfy.
The resulting outcome is the same as the current unit's local outcome with addition of the restrictions from the previous outcome which was satisfied by it.

Outcome 9-a (see figure 11) is produced by sentence 9. Since sentence 9 has no propositional content, outcome 9-a is completely neutral. But it acts as a Satisfy in relation to the most recent proposal, that specified in outcome 8-c. So outcome 9-a participates in a Satisfy move with outcome 8-c producing outcome 9-b (see Figure 11), expressing a positive attitude towards meeting next week, specifically on Thursday morning, more specifically at 9:30, in order to discuss the plans for the paper.

The Reject Move
A final move is Reject. A Reject is a move which challenges the appropriateness of a previous move. It produces no outcome since if it is not retracted, it makes the outcome which it is in relation to as if it had never occurred. An example of a Reject is something like, "Who asked you for a suggestion?"

Inter-Coder Reliability Test
An inter-coder reliability test was carried out in order to determine the extent to which this coding scheme can be learned and used consistently across coders. Four coders participated in the experiment for one week. One of the coders was me, and the other three had no previous experience with the coding scheme, two with no previous experience in discourse analysis whatsoever. On the first day, the coders familiarized themselves with the coding scheme and worked through the analysis of two sample dialogues. The next day the coders worked through dialogue 3 individually, asking technical questions as needed. Next the coders worked through dialogue 4 with no assistance. The coders met 3 days later at which time they turned in their analysis and also discussed dialogues 3 and 4. Finally, the coders worked through dialogue 5 individually and turned it in two days later. Considering that three of the coders had no previous experience with the coding scheme whatsoever, and two had no experience with discourse analysis, their results within one week's training are surprisingly good.

Table 1 displays a summary of my results. For pseudo-sentence segmentation and conversation act assignment, I used Cohen's Kappa (Cohen 1960) as a measure of agreement. PA indicates the probability of agreement calculated by dividing the number of assignments agreed upon by the total number of assignments. K indicates the result of calculating Cohen's Kappa.

Cohen's Kappa is an agreement statistic which corrects for chance. It is defined:

\[ k = \frac{P_a - P_e}{1 - P_e} \]

where

- \( P_a \) is the observed proportion of agreement, and \( P_e \) is the proportion of agreement expected by chance. This measurement of agreement is recommended for analyses of sequential behavior in (Bakeman & Gottman 1986). Because it is not possible to apply the standard Kappa measure when there are more than two coders, it was necessary to average the Kappas obtained for each pair of coders in the set (Bishop, Fienberg, & Holland 1991). In each case a \( z \) score was computed as a test for statistical significance, indicating that the results were significantly better than they would have been had the coders simply been guessing.\(^4\)

The results for pseudo-sentence segmentation were calculated by averaging pairwise kappas with each speaker classifying the end of each line broken at intonation based punctuation marks as a pseudo-sentence break or not. Conversation acts were evaluated the same way as pseudo-sentence breaks by taking the average of pairwise kappas on classification of pseudo-sentences as license, prohibit, or neutral. Because pseudo-sentence breaks were not always consistently marked across coders, the segmentation in my own coding was used as the standard, and the classification of the pseudo-sentence unit from each of the other dialogues with the most overlap with my own was selected for evaluation along with my own.

Interactional structure was difficult to evaluate since there are an arbitrary number of different relations which can be added with each additional pseudo-sentence unit. The results were evaluated by comparing the results from the other coders to my own. Coders were evaluated on each of the two tasks involved in creating the interactional structure. The first task is deciding which previous outcomes the current outcome interacts with. For each pair of outcomes I selected to interact with one another in my own analysis, I assigned 1 point. After a pair of outcomes are determined to interact with one another, the coder must choose one of the six relations discussed above. 1 point was assigned to the correct relation for each pair of outcomes. So for each pair of outcomes which I decided interacted with one another in my analysis, if the coder also decided the equivalent of those two outcomes in their own analysis related to one another and also selected the correct relation, they received 2 points. If they decided the pair of outcomes related to one another but did not select the correct relation, they received 1 point. Otherwise they did not receive any points. No penalty was assigned for identifying extraneous relations between outcomes. For each dialogue in Table 1, PA1 indicates the number of points assigned to coder 1 divided by the total number of points possible based on my own analysis. PA2 was

\(^4\)\( z \) scores above 2.85 are generally considered to be large enough to indicate statistical significance. My \( z \) scores for pseudo-sentence segmentation for each pair of coders varied between 3.7 and 10. And My \( z \) scores for conversation act assignment varied between 6.1 and 8.3.
the same for coder 2, and likewise PA3 for coder 3.

The results for building the interactional structure were lower than they could have been for two main reasons. The first reason is that although the scores for pseudo-sentence segmentation came out fairly high, there were enough differences to cause a large variation at the interactional level when individual relations are taken into consideration as in my evaluation. If two pseudo-sentence units are considered to be a single pseudo-sentence unit in another analysis, it could mean that a Support relation is lost in the other analysis in the case that those two units were part of the same move. Or it could mean that what might have been separated into two threads at the interactional level would be conflated into a single thread, in which case not only one relation would be missing, but some number of subsequent relations which would have made up the missing thread would also have been lost. Coders also differed in their approach to attaching moves to the previous outcomes in the interactional structure. If a move was composed of several pseudo-sentence units with outcomes attached to one another with Support relations, some coders attached the first outcome to previous outcomes where other coders first joined the outcomes with Support relations and then attached the final resulting outcome to the previous outcomes.

Despite these differences, the coders did in most cases identify the same threads in their analyses. These encouraging results indicate that within one week, totally inexperienced coders can make considerable progress towards learning how to apply this coding scheme.

Future Directions

I am currently working on simplifying my coding scheme to eliminate some of the difficulties discussed above. My plan is to apply this coding scheme to a large number of spontaneous scheduling dialogues to study the correspondence between the interactional structure and patterns of anaphora and definite expressions to determine whether the multiple threads identified with this coding scheme exist on the attentional level as well as on the intentional level.

### Acknowledgements

I would like to thank Project Enthusiast for providing coders for my inter-coder reliability test, especially Lori Levin for tirelessly offering her assistance with related administrative tasks and for valuable feedback on the design of the test. I would also like to thank Ariel Cohen for carefully reading and commenting on an earlier draft of this paper.

### References