Coding Discourse Structure in Dialogue (Version 1.0)

Christine Nakatani
Bell Laboratories, Lucent Technologies
chn@research.bell-labs.com

David Traum
University of Maryland
traum@cs.umd.edu

March 19, 1999

Abstract

This document is a manual for coding aspects of the discourse structure of dialogue. It was developed to serve as both as a starting point for discussion and a tool for coding exercises prior to the 3rd *Discourse Resource Initiative* (DRI) meeting, May 1998 in Chiba, Japan. The manual focuses on coding common ground units (CGUs) to get to a level of commonality between participants in dialogue, and then intentional and informational units (IUs) that represent the higher-level, hierarchical topic or purpose structure of dialogue.

University of Maryland Institute for Advanced Computer Studies Technical Report UMIACS-TR-99-03 and Computer Science Technical Report CS-TR-3980

1 Introduction

Discourse structure is many things to many researchers — attention, intentions, initiative, rhetorical structure, story trees, scripts, turn-taking behavior, etc. While there are many existing taxonomies of discourse structure, none are completely satisfactory as general purpose coding schemes for dialogue. Many of the most thorough schemes have been devised for single-speaker text, and thus are problematic to apply directly to spontaneous dialogue. Many schemes devised for dialogue are appropriate only for certain genres of dialogue (e.g., classroom instruction), or for particular domains. Others are intended for radically different purposes than those of the computational linguistics dialogue community, e.g., focusing on some of the social relationships of the participants.

It is obviously not possible at this point to devise a comprehensive coding scheme to cover all aspects of the discourse structure of dialogue. This manual therefore tries to focus in a principled way on two levels of discourse analysis for dialogues, with specific choices of content and form of representation at each level. [Traum, 1998] tries to examine some of the types of discourse structure that have been studied in the literature, proposing several dimensions by which to classify the type of structure. These dimensions include: *granularity*, *content*, and *structuring mechanisms*. The multi-dimensional space is then used to classify different extant coding schemes as to which aspects they are concerned with. We borrow some parts of that taxonomy to try to make clear what kinds of dialogue structure we will code in the present manual and which aspects we leave for future work. In particular, we adopt the partition of dialogue structure into three main ranges of granularity: *micro discourse structure*, concerning mainly those aspects within a single utterance (however that may best be defined), *meso* structure, concerned with a single sub-dialogue (involving speech by multiple partners that is locally coherent according to some organizing principle), and *macro* structure, concerning the larger, hierarchical structure of whole dialogues.

For this guide, we choose to focus on a single type of structure and content at each range of granularity. In particular, at the meso-level, we consider how subdialogues are used to synchronize participants' *mutual understanding* of what is being said. At the macro-level, we focus on hierarchical *informational* and *intentional* coherence. For now, we put aside some of the lower-level aspects of dialogue coordination, such as turn-taking and initiative, as well as aspects of individual agency, such as memory and individual agendas. We do this for several reasons. For one thing, it can be very tedious to code this kind of structure by hand, and moreover, we believe that analyses of interactive behavior, such as turn-taking and initiative, should be built on a solid understanding of how interaction facilitates or hinders the communication of information amongst participants and the achievement of their shared goals.

Further, this discourse structure coding guide is geared toward the genre of cooperative or collaborative problem-solving dialogue. Many computational systems, though not all, belong to this genre. To illustrate our guidelines, we take examples mainly from the two TRAINS-91 dialogues [Gross *et al.*, 1993] given in full in the appendix. We hope that our experience coding this genre will provide a foundation on which to develop possibly more sophisticated coding schemes for genres that may involve dialogue participants with competing goals (e.g. negotiation dialogues), or dialogue participants with complementary but not necessarily identical goals (e.g. "casual conversation" dialogues such as Switchboard).

We propose coding discourse structure in several ordered stages, corresponding to the granularity ranges mentioned above. First, a dialogue is *tokenized* into UTTERANCE-TOKENS that represent minimal analysis units, much as words are tokenized for part-of-speech tagging. Next, *micro-range* analysis of discourse relations within and between utterance-tokens may be carried out; we omit guidelines for this level of analysis at this time.[1] Third, for *meso-range* analysis, we introduce a new coding scheme that applies simplified principles of *grounding* theory [Clark and Schaefer, 1989]. In particular, meso-range analysis assigns utterance-tokens to COMMON GROUND UNITS (CGUs) that represent dialogue segments in which discourse participants add content to their common ground. Finally, we use CGUs as minimal units in the *macro-range* analysis. At this level, CGUs are grouped, based on *informational* and *intentional* relations, into I-UNITS (IUs). These IUs are hierarchically structured, like discourse segments in the linguistic structure of Grosz and Sidner [Grosz and Sidner, 1986]. However, the use of CGUs instead of the utterance-tokens themselves as the basic unit for macro-range analysis represents a significant departure from existing coding proposals.

[1]We anticipate micro-range analysis will overlap considerably with the coding of utterance-level forward-looking and backward-looking actions, e.g., as described in the DAMSL manual [Allen and Core, Draft 1997].

It emphasizes the role of low-level, informational constraints on discourse reasoning in collaborative, problem-solving dialogue, and insulates macro-level coding from several technical problems (i.e. temporal discontinuity and ambiguous utterances) which remain unsolved in alternative schemes.

The remainder of this manual is organized into four coding sections, covering (1) tokenization of a dialogue into minimal coding units (Section 2), (2) micro-range coding (Section 3), (3) meso-range coding (Section 4), and (4) macro-range coding (Section 5).

2 Preliminaries

2.1 Tokenization

Dialogues should previously have been split into utterance-tokens before you attempt to code the structure. The principles for splitting utterances into tokens are based on prosody and grammar, with the intuition that a token should correspond to a single intonational phrase [Pierrehumbert, 1980] or perhaps a single grammatical clause (i.e. tensed or untensed unit with predicate argument structure). Tokenization will be done mostly automatically, with some post-processing where necessary. No information about how these tokens are mapped to individual speech acts should be inferred, *a priori*. While it may often be the case that a single token corresponds to a single speech act at the forward/backward level, this is not guaranteed, nor is it a consideration in forming the tokens; it might require several tokens to comprise a single speech act, or a single token might contain multiple serial speech acts. However, we propose that cue phrases or discourse particles be set apart as independent tokens by convention.

By default, utterance tokens will be split into separate lines, and enumerated according to the conventions of [Gross *et al.*, 1993], that is, numbered `xx.yy`, where `xx` indicates the turn number (turns can be roughly defined as communication by one party without intervening communication by the other), and `yy`, the token within the turn. As an example, in the segment in (1), taken from dialogue 91-1.1, in [Gross *et al.*, 1993], the first token in this fragment, 7.5, is the fifth token in the seventh turn of the dialogue, the next is the sixth token in the seventh turn, and then, with the speaker change, the first token of the eighth turn.

(1) 7.5 M: okay
    7.6 : we have to get engines to the boxcars
    8.1 S: right

Also, for dyadic conversation, the identity of the speaker can be inferred simply from the turn number (odd turns for A, even turns for B). For multi-party conversation, a speaker identification code can be prepended directly onto the text line, as in (1). This speaker label will be optionally omitted from our dialogue coding since speaker identity can be automatically generated and immediately inferred from the token numbering.

For discourse structure coding, we expect tokenization will significantly increase efficiency. We also hypothesize that the coding of meso- and macro-level units will not split apart tokens, but one aspect of the homeworks will be to test this hypothesis.

3 Micro-range coding: Rhetorical relations

For the present time, we elect not to try coding at the micro-range level. Rhetorical or discourse relations at the utterance level would almost certainly need to also connect to material within an utterance token, and probably overlap syntactic and logical relations. While this kind of information is very important for many applications and research efforts, it is also not as clear how useful a domain/task-independent coding scheme might be for capturing useful information. This is certainly a topic which merits further work.

4 Meso-range coding: Common Ground Units (CGUs)

The first type of coding that you are asked to do is to cluster utterance-tokens together into units of *common ground* or mutual understanding. That is, agreement between the speakers about their understanding of what is being said (not necessarily agreement about the actual facts that are being discussed). These common ground units (CGUs) are very similar to the *Contributions* described in [Clark and Schaefer, 1989], or the *Discourse Units* (DUs) discussed in [Traum and Allen, 1992, Traum and Hinkelman, 1992, Traum, 1994]. The main difference is that we are not asking you to mark the sub-structures of these units (some of this information will be marked at other levels, such as the *understanding level* of backward act coding). Instead, we ask only that you identify distinct units of achieving common ground and mark which utterance-tokens contribute to that unit. Optionally, you can also encode the actual content that is added to common ground as a result of the unit. This will help to identify the distinctiveness of particular units, as well as to clarify how these units are used by higher level discourse structures. Even if it is too cumbersome to actually indicate the content, you should keep in mind while marking units what content is actually established into common ground as a result of the unit having been produced in the dialogue.

4.1 Basics of Common Ground Units (CGUs)

A Common Ground Unit (CGU) contains all and only the utterance tokens needed to *ground* (that is, make part of the common ground) some bit of content. This content will include the initial token of the unit, plus whatever additional content is added by subsequent tokens in the unit and added to the common ground at the same time as the initiating token. The main coherence principle for CGUs is thus not directly related to the coherence of the content itself (this kind of coherence is handled at the micro and macro levels), but whether the content is added to the common ground in the same manner (e.g., with the same acknowledgment utterance).

CGUs will require at least some initiating material by one conversational participant (the initiator), presenting the new content, as well as generally some *feedback* [Allwood *et al.*, 1992], or acknowledgment, by the other participant.[2] A fairly standard common ground unit is shown in tokens `7.6, 8.1` in example (1). Assuming that there are five previous units for the dialogue so far, we would start a new CGU, 6, and indicate that tokens `7.6 and 8.1` are components. We would end up with a table entry such as (2).[3]

[2]Depending on the modality of communication, some of this acknowledgment may be implicit or conveyed by non-linguistic action, e.g., by gaze or facial expression. Also, for some media, particularly those with some degree of persistence of messages, commonality may be assumed by the participants without any signal of understanding. In these cases, it is okay to have a CGU with contributions only by one party (assuming you actually believe that the material was assumed by the participants to have been added to the common ground).

[3]Details about how to record the code are going to necessarily be dependent on the

(2)

CGU	**Tokens**
...	
6	7.6,8.1
...	

However, a CGU can consist of more than just two tokens, one by each participant. For one thing, an initiator may take several tokens to express the initial content, as in example (3). In this case all of the first speaker's content is part of the same CGU, and the table entry would be as in (4).

(3)
5.5 M: I have to get a boxcar
5.6 : to Corning
5.7 : and then I have to load it with oranges and eventually I have to get that to Bath
5.8 : by 8 o'clock
6.1 S: right

(4)

CGU	**Tokens**
...	
5	5.5,5.6,5.7,5.8,6.1
...	

The included subsequent utterances by the same speaker can fulfill any particular type of grounding function, including just continuing to add more content, as in (3), or repairing some content, as in (5).[4]

(5)
36.2 S: well we / it just needs to get there
36.3 : by 1 PM
36.4 : I mean / sorry
36.5 : it needs to get there by 3 PM
37.1 M: +by +
37.2 : +3+ PM
37.3 : okay

tools used. The description here assumes just pencil and paper or the equivalent, forming a simple 2-column table. It would be helpful to eventually have some simple interface that lets people start a new CGU, select a *current* CGU, and then click on utterance-tokens to add or delete them from the CGU. It would also be nice to highlight in some manner all tokens belonging to a CGU, or perhaps make them the same color.

[4] From d91-4.2 in [Gross *et al.*, 1993]. The + signs indicate speech overlapping with the text just above in the previous utterance.

Here, token `36.5` corrects the mistaken information in `36.3`. Example (5) also shows a different style of acknowledgment, repeating (some of) the content, and also illustrates that an acknowledgment may also take multiple tokens: all three tokens of M's turn 37 should be added as part of this CGU. In general, tokens which convey no part of the final content expressed in the CGU may be omitted from the list of tokens (see Section 4.2), as in 15.2 in (9), or 3.2,.3.3, in (12), below. The exception is tokens containing *editing terms* specifically related to indicating the status of previous material (as in `36.4`), and tokens which express content that is changed later in the same CGU, as in `36.2, 36.3`. A borderline case is a *restart-continue*, such as 10.3 in Appendix A.2, in which a speaker starts an expression without getting very far, and then starts again from the beginning. In general, if there is no content expressed, then leave it out of the CGU. If one is unsure as to the content expressed, it is also okay to include it, if that seems more comfortable.

As well as containing a simple presentation and acknowledgment, a CGU may also contain repairs to the understanding that are "embedded" between the first initiation and final acknowledgment, involving one or more exchanges of turns. Thus a CGU may include tokens from more than two turns. A simple example is given in (6)[5], in which all of the utterances would be part of a single CGU. While some might be tempted to see example (6) as two units, one hierarchically a part of the second, for our current coding practice, we will mark all the tokens contributing to the common ground of the initiation as part of a single unit.

(6) 20.2 S: it doesn't take any time to
    21.1 M: to couple?
    22.1 S: to couple
    23.1 M: okay

In addition, extra acknowledgments by the initiating speaker are allowed as part of the CGU. Sometimes it takes several acknowledgments, back and forth to establish common ground sufficient for the current purposes [Clark and Schaefer, 1989].

While much of the structure of CGUs corresponds to *initiative-response* pairs, as in the LINDA coding scheme [Dahlbäck and Jönsson, 1998], or

[5]From d91-8.1 in [Gross *et al.*, 1993].

dialogue games [Kowtko *et al.*, 1991, Carletta *et al.*, 1997], there are some differences. Those kinds of coding schemes attempt to encode *all* of the types of exchange behavior in dialogue, whereas CGUs are attempting to capture only those parts relating to mutual understanding. As [Allwood *et al.*, 1992, Clark, 1994, Dillenbourg *et al.*, 1996] describe, there are multiple levels of coordination in dialogue. Grounding (which is what CGUs capture) is mainly concerned with the understanding level (and also the perception of messages), while there is a large part of the notion of *response* that is concerned with attitudinal reaction, which is not strictly a part of the grounding process. Except for very short reactions which are expressed in the same locution with the feedback signal of understanding, the grounding of the reaction itself will also constitute a separate CGU. Thus, a single token can be part of multiple CGUs. A good example is a question followed by an answer. While the answer does indicate understanding (and is thus part of the prior CGU which expresses the question), it also contributes new material (the content of the answer itself) that must be added to the common ground, and thus the answer also initiates a new CGU.

An example is the subdialogue in (7). As shown in (8), this contains one CGU which grounds the question, containing tokens 3.4, and 4.1, and another which grounds the answer, containing (again) 4.1, as well as the continuation in 4.2 and acknowledgment in 5.1. 4.2 is not in CGU 2, because the question is already grounded by the first part of the answer. *How* S answers the question is not important for grounding the question, merely S's demonstration that the question was understood. Thus, since 4.2 does not offer additional evidence of having understood the question (unlike in (5), the multiple acknowledgments in turn 37), it is not included in CGU 2.

(7)

3.4	M:	where are there oranges
4.1	S:	the oranges are in the warehouse
4.2	:	at Corning
5.1	M:	oh okay

(8)

CGU	**Tokens**
...	
2	3.4,4.1,
3	4.1,4.2,5.1
...	

While technically, according to [Clark and Schaefer, 1989], even simple attitudinal reactions (such as the acceptance indicated by a reply of "okay" to a proposal or statement) would also need to be grounded, usually such grounding of the simple reaction is done implicitly, by introduction of the next relevant content. This kind of implicit acknowledgment does not indicate any direct relationship to the acceptance itself, and thus it can be very difficult to determine whether the acceptance is actually acknowledged (or even needs acknowledgment) so we do not form a separate CGU to indicate the acceptance.[6] Instead one word replies (or similar multi-word phrases that do not add additional information) are seen as belonging wholly to the CGU in which they provide acknowledgment. Thus, token 5.1 from (7), is only included in CGU 3, in (8), and does not initiate a new CGU.

It is also important to realize that, unlike dialogue games, CGUs do not have to correspond to a single main dialogue act. Remember, the main organizing principle of a CGU concerns material that gets grounded together, not how that material is clustered into speech acts (or rhetorical relations, for that matter). There may be more than one speech act as part of a CGU. Example (3) shows one way this can happen, as a compound directive or statement. Another possibility is that the CGU could contain different types of actions, as in (9), where the first part of turn 15 contains a directive or statement, while the latter part indicates a question about the feasibility of this plan.

(9)	15.1	M:	and from Corning we'll pick up the oranges
	15.2	:	and um
	15.3	:	take them to Bath
	15.4	:	will it / that get m / me
	15.5	:	do you think that I can get .. this all over to Bath by 8 o'clock
	16.1	S:	yeah

In addition, some CGUs may be completed (with their content grounded), before the participants even complete a speech act: the CGU may merely serve to focus attention or identify a referent which is later used in a more

[6] For similar reasons, [Traum and Heeman, 1997] used a special category `-e` to indicate turn transitions following cue word acknowledgments.

complete action. In (10)[7], while it requires 11.3 and 13.1 to complete the suggestion of moving E2 to Corning, 11.3 and 12.1 form one CGU, while 13.1 is part of another, along with the question action in 13.2 and the acknowledgment implicit in the answer in 14.1. 14.1 also starts a new CGU for the content of the answer, which in turn is acknowledged by token 15.1, which also initiates a new CGU for the directive, acknowledged in 16.1. Since 16.1 is a simple acknowledgment, it does not start another CGU, even though it also signals acceptance of as well as understanding of the directive. The CGUs for this fragment are shown in (11).

(10)

11.3	M: wh / why don't we take E2
12.1	S: okay
13.1	M: and uh move it to Corning
13.2	: ... how long will that take
14.1	S: that'll take two hours
15.1	M: okay go ahead and do that
16.1	S: okay

(11)

CGU	**Tokens**
...	
9	11.3,12.1
10	13.1,13.2,14.1
11	14.1,15.1
12	15.1,16.1
...	

Again, the guideline for when to split a CGU is as follows: if the previous content has already been acknowledged, and the next token provides new material (however intimately linked with the prior content), then start a new CGU. If, however, the prior material has not been grounded, and a single acknowledgment (e.g., "ok") would ground both the prior and current tokens, then continue the current CGU. If the new material seems unrelated in such a way that a simple acknowledgment would *not* ground both this and the prior material, then you may start a new CGU even if the prior one is not grounded.

[7]From d91-8.1 in [Gross *et al.*, 1993].

4.2 What not to code

Not all tokens need be part of any CGU. If a token does not contain content that makes its way to the common ground (or is instrumental in other content being added to the common ground, as with editing terms), it does not need to be part of any CGU. The simplest case of this is tokens concerned wholly with what [Allwood *et al.*, 1989] call *own communication management*, other than those which correct content. Any tokens which are concerned solely with a speaker holding onto the floor or deciding what to say do not need to be part of any CGU. For example, tokens 3.1, 3.2, and 3.3 in (12) can be left out of any CGU.

(12) 3.1 M: um
3.2 : so
3.3 : let's see
3.4 : where are there oranges

A more difficult case is where the speaker actually starts to express relevant content, but then changes her mind and drops any commitment to the (tentative) expression. An example is 5.2,5.3 in (13). Since this material about the tanker car is never entered into the common ground, it need not be part of any CGU. However, one can not tell this until the actual cancellation in 5.3. Thus, it may be desirable to start a CGU with 5.2, and then later mark this CGU as *cancelled*, when 5.3 comes along. Common ground will thus correspond only to *uncanceled* CGUs. Cancelled CGUs can be represented by putting a ∗ symbol before the CGU identifier, as in (14).

(13) 5.2 M: and I see that there's a tanker car there
5.3 : oh we don't want a tanker car do we
5.4 : um
5.5 : I have to get a boxcar
5.6 : to Corning

(14)

CGU	**Tokens**
...	
∗4	5.2,5.3
...	

Cancellation here refers to cancelling the entire CGU. In cases where only *some* of the content is changed and not part of the eventual common ground, no special markings are required. Thus, in (5), above, the "1pm" contributed by token 36.3 does not make its way into the eventual common ground, having been replaced by the content of "3pm" in token 36.5. However, we do not mark this in any special way; all of these tokens are part of an uncanceled CGU. This policy is to be contrasted with later repair of grounded material in subsequent CGUs, which is handled at the macro-level, by specially marking the IU of which it is a part (see Section 5.3).

4.3 Difficult cases

Sometimes it will be very difficult to know the precise relationships of common ground. This is okay, because sometimes the participants in a conversation have the same problem! Sometimes it is difficult to tell what function (e.g., acknowledging, ignoring, or repairing) a particular token has in relation to an ongoing CGU, or which CGU a token functions within. Things get particularly difficult in cases of overlapping speech, where the speakers are working on bringing different content to common ground, or are pursuing parallel paths towards the same objective. For these difficult cases, just make the best guess you can as to which CGU(s) a token belongs to, and make notes if you think another possibility also seems plausible.

An example of a difficult case is shown in (15). 24.1 is ambiguous (especially without hearing the prosody) as to whether it is acknowledging 23.2, beginning to repair it, or starting something unrelated. Choosing the middle possibility, then 25.1 continues the same CGU, repairing "immediately" to "just after it arrives". 26.1 indicates some understanding of the plan, but a lack of acceptance. 26.2, can be left out of the CGUs (as an apology for speaking out of turn), and 26.3 starts a new CGU, with a reason not to send immediately. 26.4 can again be ignored, and 26.3 is acknowledged with 27.1. This is summarized in (16).

(15)

23.2	M:	then send it immediately
24.1	S:	so
25.1	M:	soon as it gets there
26.1	S:	+although well+
26.2	:	sorry
26.3	:	it / it takes an hour to load
26.4	:	so
27.1	M:	okay

(16)

CGU	**Tokens**
...	
16	23.2,24.1,25.1,26.1
17	26.3,27.1
...	

Another issue concerns tokens that don't seem to receive acknowledgments, but are not clearly cancelled either. Utterance 11.3 from (17) is a case in point. One could argue on one hand, that the content is so obvious from the prior context that it doesn't *need* acknowledging, and is arguably already part of the common ground. On the other hand, one could argue that 11.4 explicitly cancels the potential contribution to common ground (note that this is "okay" said by the *same* speaker, and not an acknowledgment). A third possibility is that it was never meant as a communicative action at all, but was just the speaker talking to herself, in the planning process. Finally, this content might not have been explicitly cancelled, but still might not make it to the common ground, in virtue of not having been acknowledged. We will take the position that whether it is explicitly cancelled or not, this utterance does not form part of a grounded CGU, and thus mark it as a CGU with an asterisk. We could also have just left it out of the record, like the contentless 11.2 and 11.4,5.

(17) 10.1 S: there's boxcars at Bath Danville and Elmira
11.1 M: oh okay
11.2 : ... um
[4sec]
11.3 : and we only need one boxcar
11.4 : okay
11.5 : so
[2sec]
11.6 : aha
11.7 : I see an engine and a boxcar both at Elmira
12.1 S: right

A further difficult issue is that of how long CGUs can remain *open* and receive new utterance-token components. Sometimes, after some content is ostensibly grounded, the conversants go back and talk about it again, either confirming or perhaps correcting some material. While some of this extended confirming behavior has to do with grounding, e.g., as modeled by some of the larger *Contributions* in [Clark and Schaefer, 1989], much of this behavior can be attributed to other resource limitations, such as memory and attention [Walker, 1993]. While we consider the type of grounding captured by CGUs to be a local phenomenon, we also do not wish to go the other extreme of forbidding any more inclusions to a CGU, once it has been acknowledged. For one thing, sometimes participants perform multiple acknowledgments. Also, immediate repairs should not be excluded even when occurring after acknowledgments (or sometimes *because of* incorrect acknowledgments). While the precise formulation of how long a CGU can remain open is still a research issue, for coding purposes, assume the following policy:

(18) Do not insert any utterance tokens into a CGU after you have started three subsequent CGUs no matter how relevant the new token is to the grounding of the content of that CGU.

Thus, tokens which relate to an old (closed) CGU must be part of some later CGU (perhaps starting a new one). Any relation to the common ground will need to be captured by the macro-level analysis (see, in particular Section 5.3, for corrections).

Note also, that rule (18) allows limited amounts of "crossing" of CGU token sequences, as happens quite frequently in typed (e.g., IRC or MOO) dialogues. An example of this phenomenon is shown in (19), with the CGU coding in (20).[8]

(19)

1	H	But what did you mean when you said the gun was at the bar? Who brought it there from the Colonels room?
2	S	Marie just admitted that she kenew something was wrong with the insurance
3	H	What's wrong with the insurance. The painting is a fake. Do you mean that ?
4	S	i was wrong when i said it was at the bar. it was in colonel's room
5	H	Ok
6	H	Maybe we should exchange our notebooks again.
7	H	`[Give dn1 to sherlock]`
8	S	i mean: the painting is a fake and it is insured for a huge amount of money. ...
9	S	`[Give dn1 to herc]`

(20)

CGU	**Tokens**
1	1,4
2	2,3
3	3,8
4	4,5
5	6,7,9

4.4 Review of coding principles

To summarize, you are to go through the dialogue, utterance-token by utterance-token, creating CGUs when necessary, and adding tokens to all CGUs to which they belong. The following principles summarize your decision procedures:

[8]This episode is taken from a longer MOO interaction, 5T2 in the BOOTNAP corpus `http://tecfa.unige.ch/tecfa/research/cscps/bootnap.html`. In this medium, messages are sent when the participant hits `<return>`. No attempt has been made here to divide these utterances into smaller units, using grammatical principles or punctuation, or group them into turns. Note also that 7 and 9 are non-linguistic actions (of giving notebooks) in the virtual MOO world rather than utterances said to each other. Utterance 8 is actually much longer but has been truncated for exposition here.

(21)

1. **If** the token contains *new* content, and there is no accessible ungrounded CGU, the contents of which could be acknowledged together with the current token
   then add a new CGU label, and add this token to it.
2. **if** there is an accessible CGU (according to rule (18)) for which the current token:
   (a) acknowledges the content
   (b) repairs the content
   (c) cancels the CGU (in this case, also put a * before the CGU marker, to indicate that it is canceled).
   (d) continues the content, in such a fashion that all content could be grounded together (with the same acknowledgment)

   then add this token to the CGU
3. **otherwise**, do not add this token to the CGU

Note that these rules are not exclusive: more than one may apply, so that a token can be added to more than one CGU.

4.5 Extended examples

The CGU coding for two whole dialogues are shown below. (22) has the coding for dialogue 91-1.1, while (23) has that for 91-8.1. The text of these dialogues is presented in Appendix A.

(22) **Dialogue 91-1.1**

CGU	**Tokens**	**Content**
1	1.2,1.3,2.1	state goal
2	3.4,4.1	ask oranges location
3	4.1,4.2,5.1	answer oranges location
$*4$	5.2,5.3	(locate tanker car)
5	5.5,5.6,5.7,5.8,6.1	plan decomposition
6	7.6,8.1	first step:get engines
7	9.2,10.1	ask boxcars
8	10.1,11.1	answer boxcars
$*9$	11.3	(focus search)
10	11.7,12.1	choose engine, boxcar
11	13.1,13.2,13.3,13.4,13.5,14.1,14.2	go to corning
12	15.1,15.3,15.4,15.5,16.1	request eval
13	16.1,16.2,16.3,17.1	eval ok
14	17.4,17.5,18.2,18.3	what next?

(23) **Dialogue 91-8.1**

CGU	Tokens	Content
1	1.3,1.4,1.5,1.6,1.7,1.9,2.1	specify goal
2	3.2,3.3,3.4,4.1	ask oranges
3	4.1,4.2,5.1	answer oranges
4	5.2,6.1	ask engine corning
5	6.1,7.1	answer engine
6	7.4,7.6,8.1	suggest engine
7	9.2,9.3,10.1	followup both engines
8	10.1,10.2,10.3,11.1	answer both engines
9	11.3,12.1	pick E2
10	13.1,13.2,14.1,15.1	finish suggestion, ask time
11	14.1,15.1	answer time
12	15.1,16.1	commit e2 $\rightarrow$ corning
13	17.1,17.2,17.3,17.4,17.5,17.6,17.7,18.1	use tanker car
14	19.1,19.2,19.3,20.1	send to bath
15	20.2,21.1,22.1,23.1	clarify coupling time
16	23.2,24.1,25.1,26.1	suggest send immediately/arrival
17	26.3,27.1	clarify loading
18	28.1,29.1	schedule loading
19	29.2,30.1	ask time
20	30.1,31.1	answer time
21	31.2,32.1	commit to plan
22	32.2,32.3,32.4,33.1,34.1,35.1,36.1	summarize and check
23	33.1,33.2,34.2	confirm completion

5 Macro-range coding: I-units (IUs)

At the macro-level of discourse structure coding, you are asked to reason about the relationships amongst the pieces of information that have been established as common ground. This is achieved by performing a *topic-structure* or *planning-based* analysis of the content of the CGUs, to produce a hierarchy of CGUs in a well-formed tree data structure. Such analysis proceeds in similar fashion to the intention-based methodology outlined in [Nakatani *et al.*, 1995], but there exist some crucial differences. While CGU analysis concentrates on establishing *what* is being said at the level of information exchange, macro-level analysis goes beyond this to establish relationships at a higher-level, namely relationships amongst CGUs (instead of utterance-tokens) and relationships amongst groups of CGUs. These relationships may be both informational and intentional. Thus, we refer to groupings of CGUs at the lowest level of macro-structure as I-UNITS (IUs), where "I" stands for either informational or intentional.[9]

There is wide agreement that reasoning about discourse can proceed both "top-down", with planning-based or intentional analysis, and "bottom-up", with topic-oriented or informational analysis [Hobbs, 1996, Grosz and Sidner, 1986, Moore and Pollack, 1992], *inter alia*. By "bottoming-out" our macro-level analyses with common ground units, we simply establish low-level informational constraints on both informational and intentional reasoning about the hierarchical discourse structure. The interpretation of higher-level relationships amongst CGUs and lower-level IUs, whether intentional or informational, must be consistent with the established mutual beliefs modeled by CGU analysis. Finally, while it has been argued that intentional structure and informational structure are not necessarily isomorphic and thus must be separately represented in discourse analysis [Moore and Pollack, 1992], in using CGUs as the building blocks for macro-level analysis, we rely in a practical sense on the observation made by Hobbs that more often than not, the "intentional account" of a dialogue is that the speaker has the goal that the other participant believe the speaker's own "informational account" [Hobbs, 1996]. It is beyond the scope of the current guidelines to model the informa-

[9] We do not opt for the prevalent term, *discourse segment*, to emphasize the abstraction from utterance-tokens to CGUs as the basic unit, some of whose consequences for discourse analysis are discussed in Section 5.1.

tional or intentional structures of uncooperative dialogue participants who a priori cannot share joint purposes, or intend that their intentions not be recognized.

5.1 Comparative notes

While based fundamentally on the methodology of [Nakatani *et al.*, 1995], IU analysis differs from [Nakatani *et al.*, 1995] in three important ways: (1) IU analysis uses CGUs as the minimal unit of analysis instead of utterances themselves; (2) IU analysis explicitly relies on informational constraints in intentional analysis; and (3) IU analysis extends the monologue-oriented methodology of [Nakatani *et al.*, 1995] to dialogue, while preserving most of the theoretical motivations of that work. It remains to be seen whether other single-speaker oriented schemes for macro-level discourse analysis, such as relational discourse analysis, can be generalized to handle dialogue coding.[10] It is also unclear whether higher-level discourse structure coding schemes that allow only for the linear grouping of meso-level structures, such as the chunking of moves into games, can be extended to account for hierarchical or recursive discourse relations. Hierarchical relations naturally reflect the complexity of the communicative goals of human speakers in extended discourse.

Also, as noted in the Introduction to this coding manual, there are many aspects of discourse structure that remain unanalyzed at the macro-level. Issues of individual agency and interaction management are but two areas that remain unaddressed. While we envision many possible extensions to this guide to capture additional dimensions of macro-structure, we would also like to highlight what is new in our approach toward informational and intentional analysis.

The use of CGUs as the basic units for organizing the macro-level structure is not significant for the form of the analysis. From the point of view of interpreting the content, however, using CGUs instead of the utterance-tokens themselves actually solves some outstanding problems with the use of strictly tree-based linguistic structures that organize textual units themselves, as is prevalent in current methods of macro-level analysis.

[10] At the first discourse tagging workshop held at the University of Pennsylvania in 1996, the higher-level discourse structure subgroup focused on coding monologues using [Nakatani *et al.*, 1995] and [Moser *et al.*, 1996].

The first issue that is solved is that of temporal discontinuities of utterance-tokens that share or continue the same topic or purpose. The temporally unrestricted assignment of utterance-tokens to CGU "sets" allows for the recording of such discontinuities, yet makes them invisible at the macro-level. These discontinuities, if important to understand, can be retrieved from the meso-level CGU coding and reasoned about. Discourse phenomena involving discontinuities include discourse interruptions [Grosz and Sidner, 1985], or "out-of-order" and "out-of-the-blue" segments [Nakatani *et al.*, 1995], and even crossing dependencies which arise not infrequently when parallel topics are interleaved in discussion (See e.g. CGUs 21 and 22 (23)), encoding the Trains dialogue given in Appendix A.2.

A second significant issue addressed by using CGUs rather than utterance-tokens as the basic units is that of ambiguous utterance-tokens. As shown in the example meso-level analyses, it is quite common to have an utterance-token belong to more than one CGU (e.g. when it affirms understanding of a question and then adds new information). These meaningful ambiguities, or multiple discourse roles, of certain kinds of utterance-tokens are one source of the problem of "fuzzy boundaries" in text-based discourse segmentation [Passonneau and Litman, 1997]. Another source of fuzzy boundaries is the mandatory assignment to discourse segments of rather meaningless ambiguous utterance tokens, such as strings of filled pauses, that serve only to hold the turn, if anything. For meso-level coding, such "meaningless" tokens are not assigned to CGUs and again are invisible to macro-level analysis. Of course, the role of such tokens in turn-taking or analyzing initiative is meaningful; the treatment of these tokens in the proposed scheme reveals its bias toward directly representing *informational* and *intentional* discourse relations, and not directly representing *initiative* or *turn-taking* phenomena.

All of the described properties together combine to abstract away from the "messy" aspects of dialogue that do not bear directly on intentional and informational discourse analysis. Within a CGU, there may be complex exchanges involving repairs and other performance errors, interruptions, overlapping speech, and any one of a number of types of acknowledgment or confirmation strategies, none of which may alter the higher-level intended goals and actions of the dialogue participants. By using CGUs as minimal, information-based units of analysis, IU analysis is insulated or isolated from these dialogic phenomena. We believe these properties will allow the human (or machine) analyzing the hierarchical topic or planning-based structure of a dialogue to focus most clearly on the relevant informational and intentional content and relations.

5.2 Basics of I-units (IUs)

Specific guidelines for performing macro-level segmentation of CGUs into I-units are presented below. In general, IU analysis organizes CGUs and IUs into a well-formed tree data structure, in which CGUs are the leaves and IUs are the internal nodes. The hierarchical structure is determined by discourse relations that hold between IUs.

5.2.1 Discourse relations: a review

IU trees are created by identifying certain kinds of discourse relations. Following [Grosz and Sidner, 1986], macro-level analysis captures two fundamental intentional relations between I-units, those of *domination* (or parent-child) and *satisfaction-precedence* (or sibling) relations. The corresponding informational relations are *supports* and *generates* [Pollack, 1986, Goldman, 1970]. More concretely, the domination relation can be elaborated in a planning-based framework as holding between a *subsidiary* plan and its parent, in which the completion of one plan contributes to the completion of its parent plan; the satisfaction-precedence relation can be elaborated as the temporal dependency between two plans [Lochbaum, 1994]. As is often the case, when a temporal dependency cannot be strictly established, two IUs will be placed in a sibling relationship by virtue of their each being in a subsidiary relationship with the same dominating IU.

5.2.2 Minimal units

Before beginning macro-level coding, we assume the meso-level analysis has identified CGUs, and the CGUs are uniquely identified by integers 1...n; the two levels of analysis, meso and macro, must proceed in that order. CGUs represent minimal segmentation units. You may include as many units as you like in a given IU using the notation described below.

5.2.3 Forming IUs

IU analysis consists of identifying the higher-level intentional/informational structure of the dialogue, where each IU in the macro structure achieves a joint (sub)goal or conveys information necessary to achieve a joint (sub)goal. Indicate IU boundaries and the relationship between IUs as described below.

IU heading: Precede each new IU segment with its IU segment number, according to the Gorn numbering scheme for numbering nodes and leaves in a tree data structure, as illustrated below

```
(24) iu.1 (first top-level IU)
          iu.1.1
          iu.1.2
                iu.1.2.1
                iu.1.2.2
                iu.1.2.3
          iu.1.3
                iu.1.3.1
          iu.1.4
                iu.1.4.1
                      iu.1.4.1.1
                      iu.1.4.1.2
                      iu.1.4.1.3
                      iu.1.4.1.4
                iu.1.4.2
                      iu.1.4.2.1
                      iu.1.4.2.2
     iu.2 (second top-level IU)
     .
     .
```

The top-level node or nodes (i.e. nodes that are not dominated by any other node) are assigned identifiers 1...n, in order of linear occurrence. The children of any top-level node are identified as x.1 through x.n, where x is the number assigned to the dominating node and n is the total number of children. The next level nodes are assigned nodes x.y.1 through x.y.n, where x is the top-level dominating node, and y is the identifier of the immediately dominating node, and so on.

Optionally, an IU heading may also contain a short description of the intentional or informational content of this segment; this would appear immediately following the Gorn number for the IU, i.e. on the same line.

Finally, the integer labels for the CGUs assigned to the IU are placed immediately below the IU heading. An example is shown in (25).

```
(25) iu.1: give recipe for making apple pie
     1
     2
     .
     .
```

Grouping notation:

The initiation of a new IU can be notated by placing a new IU label at the appropriate position in the file. Note that the numbering for IUs should follow their linear order of occurrence. (If, however, you need to "squeeze" in an IU, you may use alphabetic tags: e.g. iu.2a, iu.2b. This situation may arise when you have completed further labeling and then decide to add a new IU, for instance).

Use indentation (indicated by tabbing in textfiles) to distinguish between sister IUs and embedded IUs. Place all CGUs and embedded IUs in a given IU at the same indentation level as the IU label for the dominating IU, e.g.:

```
(26) iu.1: give recipe for making apple pie
     1
     2
           iu.1.1: describe selection and preparation of apples
           3
           4
           iu.1.2: describe preparation of crust
           5
           6
     .
     .
```

In (26), iu.1.1 and iu.1.2 are daughter IUs of iu.1; iu.1.1 and iu.1.2 are sister IUs. In effect, the IUs form a tree structure, with iu.1 as the root of the tree. In the default cases modeled above, the intention of an embedded IU, iu.1.1, is related to that of its embedding IU, iu.1, by an intentional domination relationship [Grosz and Sidner, 1986]. That is, the achievement of iu.1.1 partially satisfies iu.1. For example, describing the preparation of the apples partially satisfies the goal of giving a recipe for making an apple pie. In the informational realm, it can be reasoned that if iu.1 dominates iu.1.1, iu.1.1 supports iu.1. In contrast, iu.1.1 and iu.1.2 convey different subtasks in making apple pie. Neither of the two IUs generates nor enables the other, but since both support the overall goal of making apple pie, they are placed in a sibling relationship in the IU tree.

The multiple embedding of IUs is allowed. That is, a token may initiate both its immediately containing unit as well as dominating IUs, e.g.:

(27) iu.1: give recipe for making apple pie
        iu.1.1: motivate making of apple pie
        1
        2
  .
  .

The closing of IUs in most cases is not explicitly notated, but is implicit in the indentation levels. That is, if there is an IU closing and resumption of the immediately embedding IU, there will be no preceding line with an IU label, as in (28). CGU 5 resumes iu.1, in which iu.1.1. is embedded. This is inferred from the lack of an IU heading between CGU 4 and CGU 5.

(28) iu.1
  1
  2
        iu.1.1
        3
        4
  5
  6

In contrast, in (29) below, CGU 5 begins a new IU, iu.1.2, after the completion of an embedded IU, iu.1.1. In this case, iu.1.2 is not subordinate to iu.1.

(29) iu.1
    1
    2
        iu.1.1
        3
        4
    iu.2
    5
    6

The final possibility for a segment ending is starting a new segment at the same level as the one which just ended, as in iu.1.2 in (26).

5.3 Replanning and common ground corrections

Analogous to *cancelled* CGUs at the meso-level, we posit the notion of *modified* IUs at the macro-level. A modified IU is defined as any IU whose grounded content is replaced or corrected, in part or in whole, by the content introduced into common ground by later IUs. These later IUs will be referred to as *correcting* or *modifying* IUs. Modifying IUs may typically concern either changes in beliefs or changes in goals or plans. For example, a change in belief may be expressed when a mistake in reference is discovered, such as confusing engine E1 with engine E2. Or it may occur when a genuine discrepancy in understanding arises, as when an automatic spoken dialogue agent (or human!) misrecognizes *Newark* as *New York* and proceeds to provide the wrong itinerary information. A change in goal may occur when dialogue participants discover a plan constraint cannot be satisfied or enablement conditions do not hold and cannot be brought about.

For the sake of efficiency, only the most specific IUs that are modified by a correcting IU will be marked by placing the symbol # before the IU heading. In certain cases, multiple IUs must be marked as modified, as in (30).

(30) iu.1 plan weekend activity
    #iu.1.1 plan to drive in A's car
    #iu.1.2 plan to picnic at lakeside
.
.
.
iu.5 plan to go to seashore instead of lake
    iu.5.1 plan to take train because A's car will be in car shop

In other cases, an IU whose content is only partially modified by later correcting IUs will be marked, as in (31).

(31) #iu.1 plan to drive in A's car to lakehouse for the weekend
.
.
iu.5 plan to meet to catch the train instead
    iu.5.1 plan to take train because A's car will be in car shop

In general, if multiple IUs at lower-levels of the IU tree contain all occurrences of modified content, mark the multiple lower IUs individually. More general IUs whose content is only partially modified should be labeled with the symbol # only when no other marking of lower-level IUs containing the equivalent content is possible.

5.4 Examples from TRAINS

The IU coding for two whole dialogues are shown below. The IU coding assumes the CGU analyses presented in (22) (for dialogue 91-1.1) and (23) (for dialogue 91-8.1). The text of these dialogues is presented in Appendix A.

An important rule for IU analysis is that *cancelled* CGUs (marked by the symbol $*$), such as CGUs 4 and 9 in (22), are excluded from IU trees. This reflects the belief that ungrounded information does not play a large role in discourse reasoning at the informational and intentional levels. For other types of discourse structure, however, such information may prove critical to annotate. Leaving out cancelled CGUs in IU analysis is analogous to not assigning contentless tokens (such as filled pauses) to any CGU during meso-analysis.

(32) **Dialogue 91-1.1**

```
iu.1: Establish joint purpose of shipping boxcar of oranges to Bath
by 8pm
1
   iu.1.1: locate oranges
   2
   3
   iu.1.2: plan to get boxcar to corning, load with oranges, move to bath
   5
     iu.1.2.1: plan to get engines to boxcar
     6
         iu.1.2.1.1: locate boxcars
         7
         8
     10
     iu.1.2.2: plan to get boxcar/engine to corning
     11
     iu.1.2.3: plan to pick up oranges and move them to bath
     12
         iu.1.2.3.1: check time constraint is satisfied
         13
   iu.1.3: check joint purpose is satisfied
   14
```

(33) **Dialogue 91-8.1**

```
iu.1: establish joint purpose of shipping boxcar of oranges to Bath by 8pm
1
   iu.1.1: locate oranges
   2
   3
   iu.1.2: select engines
   4
   5
   6
   7
   8
   9
   iu.1.3: plan to move engine E2 to corning
   10
        iu.1.3.1: check time constraint can still be satisfied
        11
   12
   iu.1.4: plan to attach tanker car and engine and send to bath
   13
   14
      iu.1.4.1: check time used to attach car and engine
      15
      16
      iu.1.4.2: check time needed to load before sending
      17
   18
      iu.1.4.3: check time constraint is satisfied
      19
      20
      21
      22
   iu.1.5: check joint purpose is satisfied
   23
```

5.5 Macro-level analysis for collaborative problem-solving dialogues

The examples above exemplify the following macro-level structural schema for collaborative problem-solving dialogue:

(34)
- Establish problem to be collaboratively solved, or *joint goal*.
- Negotiate how to achieve joint goal.
  This may involve:
  1. Deciding which (of possibly several) recipe(s) for action to use,
  2. Deciding how to implement a recipe in the participants' domain by instantiating or identifying constraints and parameters of the recipe (e.g. deciding which of two engines to move to the orange warehouse),
  3. Breaking the plan down into subplans, whose own achievements can be similarly negotiated at the subtask level.
- Confirm achievement of (or failure to achieve) joint goal.

This schema explicitly accommodates the inferential interface between the intentional and informational levels of analysis. For example, intentional and informational relations blend as siblings at the level of choosing and implementing a recipe and breaking down a plan into subplans. This reflects the simple fact that achieving a goal via action requires knowledge of the world (e.g. identification of objects), knowledge of how to act in the world (i.e. knowledge of recipes), and knowledge of how to reason about complex relations among actions (i.e. the ability to plan and replan).

The hierarchical structure of the schema represents typical domination and satisfaction-precedence relationships.[11] Generally, it must be decided

[11] Although developed independently, this schema bears a resemblance to the problem-solving action coding scheme proposed in [Sikorski and Allen, 1997]. In particular, Sikorski and Allen's action labels, "Establish Goal" and "Evaluate Solution", map to the first and last steps in the template. Nevertheless, two crucial differences arise: first, Sikorski and Allen do not address how problem-solving action codings can be hierarchically structured;

which recipe to use to achieve a discourse purpose before the implementation of the recipe itself is discussed. So, the IU for deciding a recipe generally comes before the IU(s) on implementing the recipe. In some cases, however, the recipe itself will be built from scratch, and elaborated in parallel with implementation of the plan. In addition, the recursive step of breaking down a plan into subplans is not strictly temporally ordered with respect to choosing a recipe and implementing it. Often, once a joint purpose is established, participants may iterate through various parts of the negotiation process such as considering and rejecting several recipes for achieving the joint purpose, or may even encounter the failure of a subplan, which then necessitates replanning (see Section 5.3).

Finally, the last step in the schema, in which dialogue participants explicitly confirm their achievement of (or failure to achieve) their joint purpose, seems to frequently arise in the genre of cooperative problem-solving dialogues. These exchanges are also called pre-closings [Schegloff and Sacks, 1973]. In macro-level analysis, there is the opportunity for goal confirmation at various levels of planning. For example, in the TRAINS dialogues, participants often explicitly confirm that a substep in their joint plan meets a temporal constraint, and then later confirm that the entire plan has been solved. At the lowest level of confirmation, e.g. tokens realizing the speech act of acknowledgment, however, no macro-level coding is required since these types of utterance-token exchanges are already grouped together into a single CGU. We suggest the convention that confirmation subdialogues be embedded in the IU that concerns the most recent action(s) contributing to the outcome being evaluated. For example, in the IU analysis in (33), in iu.1.3 the partners plan to move engine E2 to Corning. In iu.1.3.1, they check that the overall time constraint can still be satisfied, if engine E2 is moved according to their plan. In this example, iu.1.3.1 is embedded relative to iu.1.3,

and second, the remaining labels of relevance, "Assess Situation" and "Establish Solution", seem to conflate the distinction between decomposing a plan into subplans and choosing and implementing recipes to achieve a given plan. For example, segments in which dialogue participants propose either components to a solution (i.e. subplans) or resources to accomplish a task are both labeled as "Establish Solution" segments. Following our proposed schema, the former would be part of step 3 in the process, "Negotiate how to achieve joint goal", while the latter would be part of step 2 in that same process. The recursion in the proposed schema falls out naturally from this distinction. In this regard, the schema maps more closely to the *discourse script* for the TRAINS domain used in [Traum, 1994], since both explicitly address goal decomposition.

since the time constraint checked in iu.1.3.1 is evaluated with respect to the action recently planned out in iu.1.3.

It can be seen from the schema presented that the informational-intentional interface relies heavily on the distinct notions of plan (a mental state to intend to achieve a goal) versus recipe (a data structure description for action to achieve a specified goal) [Pollack, 1986, Bratman, 1987]. When dialogue participants are explicitly discussing their goals or subgoals, the IU takes on a more intentional flavor. When participants are identifying parameters for executing recipes, the IU takes on a more informational flavor.

We will have to discover whether the application of macro-level analysis to additional genres of corpora, such as conversational and tutoring dialogues, can be handled by extensions to this schema or will require the formulation of entirely new schemata.

6 Summary

In this document, we have presented a two-level coding scheme for discourse structure in dialogue. At the meso-level of discourse structure, we defined guidelines for coding Common Ground Units (CGUs), based on analysis of how the hearer and speaker establish mutual beliefs through dialogue. At the macro-level, we used CGUs as primitive units of analysis and defined guidelines for carrying out I-unit (IU) analysis. IU codings reflect the intentional or informational structure of dialogue. This two-level scheme is a natural complement to schemes intended to cover other aspects of dialogue, especially micro-level schemes developed for dialogue acts (e.g., [Allen and Core, Draft 1997]) or co-reference (e.g., [Passonneau, 1997]).

The coding scheme presented here is more a first attempt than a comprehensive product. We expect further development in several directions. First, there may be additional coding schemes along other dimensions to address other dialogue phenomena (e.g. initiative); the current scheme considers only grounding at the meso-range, and I-structure at the macro-range. Secondly, we expect fine-tuning on the specific coding principles to facilitate both greater reliability and utility of the two levels we do cover. We hope other researchers will explore whether a more productive synergy can be found between the two levels, both in theory and in practice. The relation we hypothesize between the two levels, and our supposition that important relations may be found between micro-level schemes and the two-level scheme posited here, lay the groundwork for more focused investigations of coding schemes for discourse structure in dialogue than have previously existed within the DRI initiative.

Acknowledgments

We would like to thank Masato Ishizaki for helpful comments on previous drafts of this document.

References

[Allen and Core, Draft 1997] James Allen and Mark Core. Draft of damsl: Dialog act markup in several layers. available through the WWW at: http://www.cs.rochester.edu/research/trains/annotation, Draft, 1997.

[Allwood *et al.*, 1989] Jens Allwood, Joakim Nivre, and Elisabeth Ahlsen. Speech management: On the non-written life of speech. Technical Report (GPTL) 58, Gothenburg Papers in Theoretical Linguistics, University of Göteborg, 1989.

[Allwood *et al.*, 1992] Jens Allwood, Joakim Nivre, and Elisabeth Ahlsen. On the semantics and pragmatics of linguistic feedback. *Journal of Semantics*, 9, 1992.

[Bratman, 1987] Michael E. Bratman. *Intention, Plans, and Practical Reason*. Harvard University Press, 1987.

[Carletta *et al.*, 1997] Jean Carletta, Amy Isard, Stephen Isard, Jacqueline C. Kowtko, Gwynewth Doherty-Sneddon, and Anne H. Anderson. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23(1):13–31, 1997.

[Clark and Schaefer, 1989] Herbert H. Clark and Edward F. Schaefer. Contributing to discourse. *Cognitive Science*, 13:259–294, 1989. Also appears as Chapter 5 in [Clark, 1992].

[Clark, 1992] Herbert H. Clark. *Arenas of Language Use.* University of Chicago Press, 1992.

[Clark, 1994] Herbert H. Clark. Managing problems in speaking. *Speech Communication*, 15:243 – 250, 1994.

[Dahlbäck and Jönsson, 1998] Nils Dahlbäck and Arne Jönsson. A coding manual for the linköping dialogue model. unpublished manuscript, 1998.

[Dillenbourg *et al.*, 1996] Pierre Dillenbourg, David Traum, and Daniel Schneider. Grounding in multi-modal task-oriented collaboration. In *Proceedings of the European Conference on AI in Education*, 1996.

[Goldman, 1970] A. I. Goldman. *A Theory of Human Action.* Princeton University Press, Princeton, NJ, 1970.

[Gross *et al.*, 1993] Derek Gross, James Allen, and David Traum. The TRAINS 91 dialogues. TRAINS Technical Note 92-1, Department of Computer Science, University of Rochester, July 1993.

[Grosz and Sidner, 1985] B. Grosz and C. Sidner. Discourse structure and the proper treatment of interruptions. In *Proceedings of IJCAI-85*, Los Angeles, Ca., August 1985.

[Grosz and Sidner, 1986] Barbara Grosz and Candace Sidner. Attention, intentions, and the structure of discourse. *Computational Linguistics*, 12(3):175–204, 1986.

[Hobbs, 1996] Jerry R. Hobbs. On the relation between the informational and intentional perspectives on discourse. In Eduard H. Hovy and Donia R. Scott, editors, *Computational and conversational discourse*, pages pages 139–157. Springer Verlag, 1996.

[Kowtko *et al.*, 1991] Jacqueline C. Kowtko, S. Isard, and G. Doherty. Conversational games within dialogue. In *Proceedings of the ESPRIT Workshop on Discourse Coherence*, 1991.

[Lochbaum, 1994] Karen Lochbaum. *Using Collaborative Plans to Model the Intentional Structure of Discourse.* PhD thesis, Harvard University, 1994. Available as Technical Report 25-94.

[Moore and Pollack, 1992] J. Moore and M. Pollack. A problem for rst: the need for multi-level discourse structure. *Computational Linguistics*, 18(4):537–544, 1992.

[Moser *et al.*, 1996] Megan G. Moser, Johanna D. Moore, and Erin Glendening. Instructions for coding explanations: Identifying segments, relations and minimal units. Technical Report 96-1, University of Pittsburgh, Department of Computer Science, 1996.

[Nakatani *et al.*, 1995] Christine H. Nakatani, Barbara Grosz, David Ahn, and Julia Hirschberg. Instructions for annotating discourse. Technical Report 21-95, Center for Research in Computing Technology, Harvard University, Cambridge, MA, September 1995.

[Passonneau and Litman, 1997] Rebecca Passonneau and Diane Litman. Discourse segmentation by human and automated means. *Computational Linguistics*, 23(1):103–140, 1997.

[Passonneau, 1997] R. Passonneau. Instructions for applying discourse reference annotation for multiple applications (DRAMA). Unpublished manuscript., December 1997.

[Pierrehumbert, 1980] J. B. Pierrehumbert. The phonology and phonetics of english intonation. Doctoral dissertation, Massachusetts Institute of Technology, 1980.

[Pollack, 1986] Martha E. Pollack. *Inferring Domain Plans in Question-Answering*. PhD thesis, University of Pennsylvania, 1986.

[Schegloff and Sacks, 1973] Emmanuel A. Schegloff and H. Sacks. Opening up closings. *Semiotica*, 7:289–327, 1973.

[Sikorski and Allen, 1997] Teresa Sikorski and James F. Allen. A scheme for annotating problem solving actions in dialogue. In *Working Notes of the AAAI Fall Symposium on Communicative Action in Humans and Machines*, 1997.

[Traum and Allen, 1992] David R. Traum and James F. Allen. A speech acts approach to grounding in conversation. In *Proceedings 2nd International*

Conference on Spoken Language Processing (ICSLP-92), pages 137–40, October 1992.

[Traum and Heeman, 1997] David R. Traum and Peter Heeman. Utterance units in spoken dialogue. In Elisabeth Maier, Marion Mast, and Susann Luperfoy, editors, *Dialogue Processing in Spoken Language Systems — ECAI-96 Workshop*, Lecture Notes in Artificial Intelligence, pages 125–140. Springer-Verlag, Heidelberg, 1997.

[Traum and Hinkelman, 1992] David R. Traum and Elizabeth A. Hinkelman. Conversation acts in task-oriented spoken dialogue. *Computational Intelligence*, 8(3):575–599, 1992. Special Issue on Non-literal language.

[Traum, 1994] David R. Traum. *A Computational Theory of Grounding in Natural Language Conversation.* PhD thesis, Department of Computer Science, University of Rochester, 1994. Also available as TR 545, Department of Computer Science, University of Rochester.

[Traum, 1998] David R. Traum. Notes on dialogue structure. Unpublished manuscript, 1998.

[Walker, 1993] Marilyn A. Walker. *Informational Redundancy and Resource Bounds in Dialogue.* PhD thesis, University of Pennsylvania, 1993.

A Full Dialogue Examples

A.1 TRAINS Dialogue 91-1.1

```
Total Time: 1'38''      Total Turns: 20      Total Utterances: 63

UU#  Speaker: Utterance
1.1 M: okay
1.2  : I have to
1.3  : ship a boxcar of oranges to Bath by 8 o'clock today
...
2.1 S: okay
3.1 M: um
3.2  : so
3.3  : let's see
3.4  : where are there _oranges_
4.1 S: the oranges are in the warehouse
4.2  : at Corning
5.1 M: oh okay
5.2  : and I see that there's a tanker car there
5.3  : oh we don't want a tanker car do we
5.4  : um
5.5  : I have to get a boxcar
5.6  : to Corning
5.7  : and then I have to load it with oranges and eventually I
  have to get that to Bath
5.8  : by 8 o'clock
6.1 S: right
7.1 M: um
7.2  : ts
7.3  : so
7.4  : what's the cl  /
7.5  : w okay
7.6  : we have to get engines to the boxcars
8.1 S: right
9.1 M:  +um+
9.2  : are there boxcars anywhere except at Bath
```

```
10.1 S: there's boxcars at Bath Danville and Elmira
11.1 M: oh okay
11.2  : ... um
[4sec]
11.3  : and we only need one boxcar
11.4  : okay
11.5  : so
[2sec]
11.6  : aha
11.7  : I see an engine and a boxcar both at Elmira
12.1 S: right
13.1 M: this looks like the best thing to do
13.2  : so we should get
13.3  : ... the eng / engine to picks up the boxcar
13.4  : and head for Corning
13.5  : 's that sound reasonable
14.1 S: sure
14.2  : that sounds good
15.1 M: and from Corning we'll pick up the oranges
15.2  : and um
15.3  : take them to Bath
15.4  : will it / that get m / me
15.5  : do you think that I can get  .. this all over to Bath by 8 o'clock
16.1 S: yeah
16.2  : that gets us to Bath at f / 5 AM
16.3  : so it's plenty of time
17.1 M: okay
17.2  : so
17.3  : um
17.4  : do I tell you what to do at this point
17.5  : [chuckle]
18.1 S: +um+
18.2 S: well
18.3 S: we're done
19.1 M: [chuckle]
20.1 S: +we can now sh ship+
20.2  : we're done with the warmup problem
```

A.2 TRAINS Dialogue 91-8.1

```
Total Time:  1'26''  Total Turns: 36      Total Utterances: 81

UU# Speaker:Utterance

1.1 M: all right
1.2  : so it's uh
1.3  : it's 12 midnight
1.4  : and we need to get uh
1.5  : a boxcar of oranges to Bath
1.6  : by
1.7  : 8 AM
1.8  : uh
1.9  : so in 8 hours
2.1 S: okay
3.1 M: and uh
3.2  : so are there any oranges
3.3  : at / uh
3.4  : Corning
4.1 S: yes
4.2  : there are
5.1 M: okay
5.2  : is there an _engine_ at Corning
6.1 S: no
7.1 M: all right
7.2  : 's see
7.3  : so
7.4  : why don't we take
7.5  : uh
7.6  : an engine from Elmira
8.1 S: okay
9.1 M: uh
9.2  : are both available
9.3  : there're two engines there I
```

```
10.1 S: yeah
10.2  : they're both
10.3  : they're both there
11.1 M: okay
11.2  : so
11.3  : wh / why don't we take E2
12.1 S: okay
13.1 M: and uh move it to Corning
13.2  : ... how long will that take
14.1 S: that'll take two hours
15.1 M: okay go ahead and do that
16.1 S: okay
17.1 M: and then um
17.2  : we can take the tanker car
17.3  : in Corning
17.4  : and / uh
17.5  : attach it to the uh
17.6  : the engine E / the engine / engine that's coming over
17.7  : E2
18.1 S: okay
19.1 M: and then send them off to Bath
19.2  : at 2 AM
19.3  : err 2:30 AM
20.1 S: okay
20.2  : it doesn't take any time to
21.1 M: to couple
22.1 S: to couple
23.1 M: okay
23.2  : then send it immediately
24.1 S: so
25.1 M: soon as it gets there
26.1 S:        +although well+
26.2  : sorry
26.3  : it / it takes an hour to load
26.4  : so
27.1 M: okay
28.1 S: we have to do that
```

```
29.1 M: so by th / at 3
29.2  : then how long does it take to get from Corning to Bath
30.1 S: two hours
31.1 M: so fine
31.2  : go ahead
32.1 S:  +so+
32.2  : get to bath
32.3  : at 5 AM
32.4  : so
33.1 M: and then
33.2  : the task is .. finished
34.1 S:             +that should be+  fine
34.2  : yeah
35.1 M: +yeah+
36.1 S: okay
```