The Reliability of A Dialogue Structure Coding Scheme

Jean Carletta, University of Edinburgh*
Amy Isard, University of Edinburgh*
Stephen Isard, University of Edinburgh†
Jacqueline C. Kowtko, University of Edinburgh†
Gwyneth Doherty-Sneddon, University of Stirling‡
Anne H. Anderson, University of Glasgow§

This paper describes the reliability of a dialogue structure coding scheme which is based on utterance function, game structure, and higher level transaction structure, and which has been applied to a corpus of spontaneous task-oriented spoken dialogues.

1. Introduction

Dialogue work, like the rest of linguistics, has traditionally used isolated examples, either constructed or real. Now many researchers are beginning to try to code large dialogue corpora for higher level dialogue structure in the hope of giving their findings a firmer basis. The purpose of this paper is to introduce and describe the reliability of a scheme of dialogue coding distinctions which have been developed for use on the Map Task Corpus (Anderson et al., 1991). These dialogue structure distinctions were developed within a larger 'vertical analysis' of dialogue encompassing a range of phenomena beginning with speech characteristics, and therefore are intended to be useful whenever an expression of dialogue structure is required.

2. Other dialogue structure coding schemes

A number of alternative ways of coding dialogue are mentioned in the recent literature. Walker and Whittaker (1990) mark utterances as *assertions*, *commands*, *questions*, or *prompts* (utterances which do not express proposition content) in an investigation of mixed initiative in dialogue. Sutton et al. (1995) classify the possible responses to a question in terms of whether or not they answer the question and how complete and concise the answer is, as part of designing an automated spoken questionnaire. Alexandersson et al. (1995) devise a set of seventeen 'speech acts' which occur in dialogues between people setting the date for a business meeting; some of these speech acts are task-specific. They use these speech acts to derive statistical predictions about what speech act will come next within VERBMOBIL, a speech-to-speech dialogue translation system which operates on demand for limited stretches of dialogue. Nagata and Morimoto (1993) use a set of nine more task-independent illocutionary force distinctions for a similar purpose. Ahrenberg et al. (1995) divide moves in 'Wizard of Oz' information-seeking dialogues into initiations and responses and then further classify them according to the function

* Human Communication Research Centre, 2 Buccleuch Place, Edinburgh EH8 9LW, Scotland.
† Centre for Speech Technology Research, 80 South Bridge, Edinburgh EH1 1HN, Scotland.
‡ Psychology Department, Stirling FK9 4LA, Scotland.
§ Human Communication Research Centre, 56 Hillhead Street, Glasgow G12 9YR, Scotland.

which they serve in the information transfer, in order to show how this relates to the focus structure of the dialogues. Condon and Cech (1995), while investigating the difference between face-to-face and computer-mediated communication, classify utterances according to the role they take in the decision-making.

The coding described in this paper differs from all of these coding schemes in three important ways. First, although the move categories are informed by computational models of dialogue, the categories themselves are more independent of the task than the schemes which are devised with particular machine dialogue types in mind. Second, although other coding schemes may distinguish many categories for utterances segmented according to the discourse goals which they serve, by showing game and transaction structures this coding scheme attempts to classify dialogue structure at higher levels as well. Finally, although the other coding schemes appear to have been devised primarily with one purpose in mind, this coding scheme is intended to represent dialogue structure generically so that it can be used in conjunction with codings of many other dialogue phenomena.

3. The Dialogue Structure Coding

The coding distinguishes three levels of dialogue structure, similar to the three middle levels in Sinclair and Coulthard's (1975) analysis of classroom discourse. At the highest level, dialogues are divided into *transactions*, which are subdialogues that accomplish one major step in the participants' plan for achieving the task. The size and shape of transactions is largely dependent on the task. In the Map Task, two participants have slightly different versions of a simple map with approximately fifteen landmarks on it. One participant's map has a route printed on it; the task is for the other participant to duplicate the route. A typical transaction is a subdialogue which gets the route follower to draw one route segment on the map.

Transactions are made up of *conversational games*, which are often also called dialogue games (Carlson, 1983; Power, 1979), interactions (Houghton, 1986), or exchanges (Sinclair and Coulthard, 1975), and show the same structure as Grosz and Sidner's discourse segments (1986) when applied to task-oriented dialogue. All forms of conversational games embody the observation that, by and large, questions are followed by answers, statements by acceptance or denial, and so on. Game analysis makes use of this regularity to differentiate between *initiations* which set up a discourse expectation about what will follow, and *responses* which fulfill those expectations. In addition, games are often differentiated by the kind of discourse purpose which they have — for example, getting information from the partner or providing information. A conversational game is a set of utterances starting with an initiation and encompassing all utterances up until the purpose of the game has been either fulfilled (e.g., the requested information has been transferred) or abandoned. Games can nest within each other if one game is initiated to serve the larger goal of a game which has already been initiated (for instance, if a question is on the floor but the hearer needs to ask for clarification before answering). Games are themselves made up of *conversational moves*, which are simply different kinds of initiations and responses classified according to their purposes.

All levels of the dialogue coding are described in detail in (Carletta et al., 1996).

3.1 The Move Coding Scheme

The move coding analysis is the most substantial level. It was developed by extending the moves which make up Houghton's (1986) interaction frames to fit the kinds of interactions found in the Map Task dialogues. In any categorisation there is a tradeoff between usefulness and ease or consistency of coding. Too many semantic distinctions

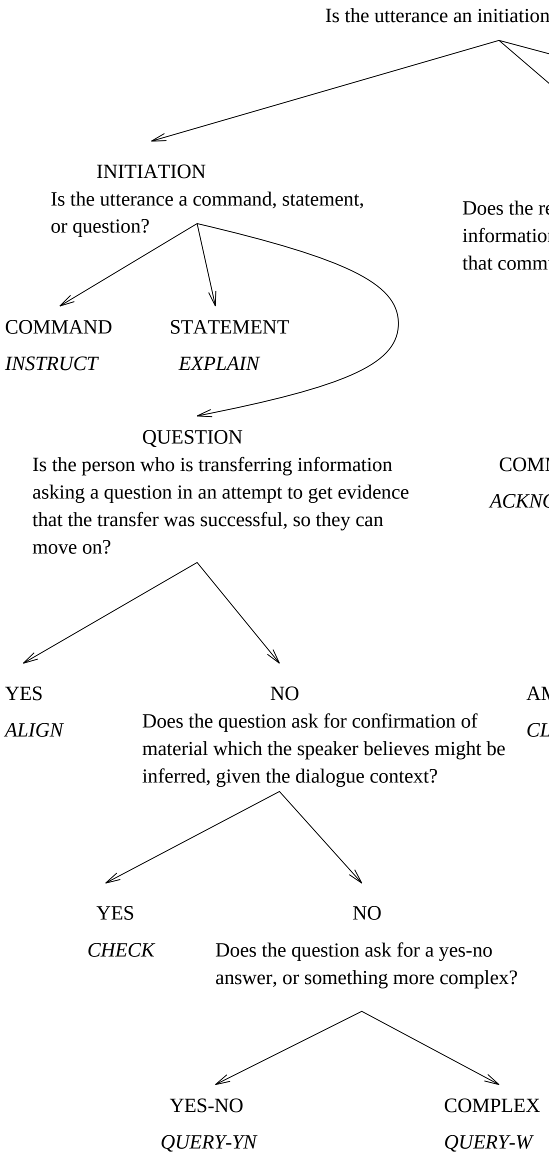

Figure 1
Conversational move categories.

make coding difficult. These categories were chosen to be useful for a range of purposes but still be reliable. The distinctions used to classify moves are summarised in figure 1.

3.1.1 The INSTRUCT Move. An INSTRUCT move commands the partner to carry out an action. Where actions are observable, the expected response could be performance of the action. The instruction can be quite indirect, as in example 3 below, as long as there is a specific action which the instructor intends to elicit (in this case, focusing on the start point). In the Map Task, this usually involves the route giver telling the route follower how to navigate part of the route. Participants can also give other INSTRUCT moves, such as telling the partner to go through something again but more slowly. In these and later examples, 'G' denotes the instruction giver, the participant who knows the route, and 'F', the instruction follower, the one who is being told the route. Editorial comments which help to establish the dialogue context are given in square brackets.

Example 1
G: Go right round, ehm, until you get to just above them.

Example 2
G: If you come in a wee bit so that you're about an inch away from both edges.

Example 3
G: We're going to start above th... directly above the telephone kiosk.

Example 4
F: Say it... start again.

Example 5
F: Go. [as first move of dialogue; poor quality but still an instruction]

3.1.2 The EXPLAIN Move. An EXPLAIN states information which has not been directly elicited by the partner. (If the information were elicited, the move would be a response, such as a reply to a question.) The information can be some fact about either the domain or the state of the plan or task, including facts which help establish what is mutually known.

Example 6
G: Where the dead tree is on the other side of the stream there's farmed land.

Example 7
G: I've got a great viewpoint away up in the top left-hand corner.

Example 8
F: I have to jump a stream.

Example 9
F: I'm in between the remote village and the pyramid.

Example 10
F: Yeah, that's what I thought you were talking about.

3.1.3 The CHECK Move. A CHECK move requests the partner to confirm information that the speaker has some reason to believe, but is not entirely sure about. Typically the information to be confirmed is something which the partner has tried to convey explicitly or something which the speaker believes was meant to be inferred from what the partner has said. In principle, CHECK moves could cover past dialogue events (e.g., "I told you about the land mine, didn't I?") or any other information that the partner is in a position to confirm. However, CHECK moves are almost always about some information which the speaker has been told. One exception in the Map Task occurs when a participant is explaining a route for the second time to a different route follower, and asks for confirmation that a feature occurs on the partner's map even though it has not yet been mentioned in the current dialogue.

Example 11

G: ... you go up to the top left-hand corner of the stile, but you're only, say about a centimetre from the edge, so that's your line.
F: OK, up to the top of the stile?

Example 12

G: Ehm, curve round slightly to your right.
F: To my right?
G: Yes.
F: As I look at it?

Example 13

G: Right, em, go to your right towards the carpenter's house.
F: Alright well I'll need to go below, I've got a blacksmith marked.
G: Right, well you do that.
F: Do you want it to go below the carpenter? [*]
G: No, I want you to go up the left hand side of it towards green bay and make it a slightly diagonal line, towards, em sloping to the right.
F: So you want me to go above the carpenter? [**]
G: Uh-huh.
F: Right.

Note that in example 13, the move marked * is not a CHECK because it asks for new information — F has only stated that he'll have to go below the blacksmith — but the move marked ** is a CHECK because F has inferred this information from G's prior contributions and wishes to have confirmation.

3.1.4 The ALIGN Move. An ALIGN move checks the attention or agreement of the partner, or his readiness for the next move. At most points in task-oriented dialogue, there is some piece of information which one of the participants is trying to transfer to the other participant. The purpose of the most common type of ALIGN move is for the transferer to know that the information has been successfully transferred, so that they can close that part of the dialogue and move on. If the transferee has acknowledged the information clearly enough, an ALIGN move may not be necessary. If the transferer needs more evidence of success, then alignment can be achieved in two ways. If the transferer is sufficiently confident that the transfer has been successful, a question such as "OK?" suffices. Some participants ask for this kind of confirmation immediately after issuing an instruction, probably to force more explicit responses to what they say. Less confident transferers can ask for confirmation of some fact which the transferee should be able to infer from the transferred information, since this provides stronger evidence of success. Although ALIGN moves usually occur in the context of an unconfirmed information transfer, participants also use them at hiatuses in the dialogue to check that "everything is OK" (i.e., that the partner is ready to move on) without asking about anything in particular.

Example 14

G: OK? [after an instruction and an acknowledgement]

Example 15

G: You should be skipping the edge of the page by about half an inch, OK?

Example 16

G: Then move that point up half an inch so you've got a kind of diagonal line again.
F: Right.
G: This is the left-hand edge of the page, yeah? [where the query is asked very generally about a large stretch of dialogue, 'just in case']

3.1.5 The QUERY-YN Move. A QUERY-YN asks the partner any question which takes a 'yes' or 'no' answer and does not count as a CHECK or an ALIGN. In the Map Task, these questions are most often about what the partner has on the map. They are also quite often questions which serve to focus the attention of the partner on a particular part of the map or which ask for domain or task information where the speaker does not think that information can be inferred from the dialogue context.

Example 17

G: Do you have a stone circle at the bottom?

Example 18

G: I've mucked this up completely have I?

Example 19

F: I've got Dutch Elm.
G: Dutch Elm. Is it written underneath the tree?

Example 20

G: Have you got a haystack on your map?
F: Yeah
G: Right just move straight down from there, then,
F: Past the blacksmith? [with no previous mention of blacksmith or any distance straight down, so that F can't guess the answer]

3.1.6 The QUERY-W Move. A QUERY-W is any query which is not covered by the other categories. Although most moves classified as QUERY-W are wh-questions, otherwise unclassifiable queries also go in this category. This includes questions which ask the partner to choose one alternative from a set, as long as the set is not 'yes' and 'no'. Although technically the tree of coding distinctions allows for a CHECK or an ALIGN to take the form of a wh-question, this is unusual in English. In both ALIGN and CHECK moves, the speaker tends to have an answer in mind, and it is more natural to formulate them as yes-no questions. Therefore in English all wh-questions tend to be categorised as QUERY-W. It might be possible to subdivide QUERY-W into theoretically interesting categories rather than using it as a 'wastebasket', but in the Map Task such queries are rare enough that subdivision is not worthwhile.

Example 21

G: Towards the chapel and then you've
F: Towards what?

Example 22

G: Right, okay. Just move round the crashed spaceship so that you've ... you reach the finish, which should be left ... just left of the ... the chestnut tree.

F: Left of the bottom or left of the top of the chestnut tree?

Example 23

F: No I've got a... I've got a trout farm over to the right underneath Indian Country here.

G: Mmhmm.

I want you to go three inches past that going south, in other words just to the level of that, I mean, not the trout farm.

F: To the level of what?

3.2 Response moves

The following moves are used within games after an initiation, and serve to fulfill the expectations set up within the game.

3.2.1 The Acknowledge Move. An ACKNOWLEDGE move is a verbal response which minimally shows that the speaker has heard the move to which it responds, and often also demonstrates that the move was understood and accepted. Verbal acknowledgements do not have to appear even after substantial explanations and instructions, since acknowledgement can be given non-verbally, especially in face-to-face settings, and because the partner may not wait for one to occur. Clark and Schaefer (1989) give five kinds of evidence that an utterance has been accepted: continued attention, initiating a relevant utterance, verbally acknowledging the utterance, demonstrating an understanding of the utterance by paraphrasing it, and repeating part or all of the utterance verbatim. Of these kinds of evidence, only the last three count as ACKNOWLEDGE moves in this coding scheme; the first kind leaves no trace in a dialogue transcript to be coded, and the second involves making some other, more substantial dialogue move.

Example 24

G: Ehm, if you... you're heading southwards.

F: Mmhmm.

Example 25

G: Do you have a stone circle at the bottom?

F: No.

G: No, you don't.

3.2.2 The Reply-y Move. A REPLY-Y is any reply to any query with a yes-no surface form which means 'yes', however that is expressed. Since REPLY-Y moves are elicited responses, they normally only appear after QUERY-YN, ALIGN, and CHECK moves.

Example 26

G: See the third seagull along?

F: Yeah.

Example 27

G: Do you have seven beeches?

F: I do.

Example 28
F: Green Bay?
G: Uh-huh.

Example 29
G: Do you want me to run by that one again?
F: Yeah, if you could.

3.2.3 The Reply-n Move. Similar to reply-y, a reply to a a query with a yes/no surface form which means 'no' is a reply-n.

Example 30
G: Do you have the west lake, down to your left?
F: No.

Example 31
G: So you're at a point that's probably two or three inches away from both the top edge, and the left-hand side edge. Is that correct?
F: No, no at the moment.

One caveat about the meaning of the difference between reply-y and reply-n: rarely, queries include negation (e.g., "You don't have a swamp?"; "You're not anywhere near the coast?"). As for the other replies, whether the answer is coded as a reply-y or a reply-n depends on the surface form of the answer, even though in this case "yes" and "no" can mean the same thing.

3.2.4 The Reply-w Move. A reply-w is any reply to any type of query which doesn't simply mean 'yes' or 'no'.

Example 32
G: And then below that, what've you got?
F: A forest stream.

Example 33
G: No, but right, first, before you come to the bakery do another wee lump
F: Why?
G: Because I say.

Example 34
F: Is this before or after the backward s?
G: This is before it.

3.2.5 The Clarify Move. A clarify move is a reply to some kind of question in which the speaker tells the partner something over and above what was strictly asked. If the information is substantial enough, then the utterance is coded as a reply followed by an explain, but in many cases, the actual change in meaning is so small that coders are reluctant to mark the addition as truly informative. Route givers tend to make CLARIFY moves when the route follower seems unsure of what to do, but there isn't a specific problem on the agenda (such as a landmark now known not to be shared).

Example 35

G: And then, have you got the pirate ship?
F: Mmhmm.
G: Just curve from the point, go right ... go down and curve into the right til you reach the tip of the pirate ship
F: So across the bay?
G: Yeah, through the water.
F: So I just go straight down?
G: Straight down, and curve to the right, til you're in line with the pirate ship.

Example 36

[... instructions which keep them on land...]
F: So I'm going over the bay?
G: Mm, no, you're still on land.

3.2.6 Other possible responses. All of these response moves help to fulfill the goals proposed by the initiating moves which they follow. It is also theoretically possible at any point in the dialogue to refuse to take on the proposed goal, either because the responder feels that there are better ways to serve some shared higher level dialogue goal or because the responder does not share the same goals as the initiator. Often refusal takes the form of ignoring the initiation and simply initiating some other move. However, it is also possible to make such refusals explicit; for instance, a participant could rebuff a question with "No, let's talk about...", an initiation with "What do you mean — that won't work!", or an explanation about the location of a landmark with "Is it?", said with an appropriately unbelieving intonation. One might consider these cases akin to ACKNOWLEDGE moves, but with a negative slant. These cases were sufficiently rare in the corpora used to develop the coding scheme that it was impractical to include a category for them. However, it is possible that in other languages or communicative settings, this behaviour will be more prevalent. Grice and Savino (1995) found that such a category was necessary when coding Italian Map Task dialogues where speakers were very familiar with each other. They called the category OBJECT.

3.3 The READY Move

In addition to the initiation and response moves, the coding scheme identifies READY moves as moves which occur after the close of a dialogue game and prepare the conversation for a new game to be initiated. Speakers often use utterances such as "OK" and "right" to serve this purpose. It is a moot point whether READY moves should form a distinct move class or should be treated as discourse markers attached to the subsequent moves, but the distinction is not a critical one, since either interpretation can be placed on the coding. It is sometimes appropriate to consider READY moves as distinct, complete moves in order to emphasise the comparison with ACKNOWLEDGE moves, which are often just as short and even contain the same words as READY moves.

Example 37

G: Okay. Now go straight down.

Example 38

G: Now I have banana tree instead.

Example 39

G: Right, if you move up very slightly to the right along to the right.

3.4 The Game Coding Scheme

Moves are the building blocks for conversational game structure, which reflects the goal structure of the dialogue. In the move coding, a set of initiating moves are differentiated, all of which signal some kind of purpose in the dialogue. For instance, instructions signal that the speaker intends the hearer to follow the command, queries signal that the speaker intends to acquire the information requested, and statements signal that the speaker intends the hearer to acquire the information given. A conversational game is a sequence of moves starting with an initiation and encompassing all moves up until that initiation's purpose is either fulfilled or abandoned.

There are two important components of any game coding scheme. The first is an identification of the game's purpose; in this case, the purpose is identified simply by the name of the game's initiating move. The second is some explanation of how games are related to each other. The simplest, paradigmatic relationships are implemented in computer-computer dialogue simulations, such as those of Power (1979) and Houghton (1986). In these simulations, once a game has been opened, the participants work on the goal of the game until they both believe that it has been achieved or that it should be abandoned. This may involve embedding new games with subservient purposes to the top level one being played (for instance, clarification subdialogues about some crucial missing information), but the embedding structure is always clear and mutually understood. Although some natural dialogue is this orderly, much of it is not; participants are free to initiate new games at any time (even while the partner is speaking), and these new games can introduce new purposes rather than serving some purpose which is already present in the dialogue. In addition, natural dialogue participants often fail to make clear to their partners what their goals are. This makes it very difficult to develop a reliable coding scheme for complete game structure.

The game coding scheme simply records those aspects of embedded structure which are of the most interest. First, the beginning of new games is coded, naming the game's purpose according to the game's initiating move. Although all games begin with an initiating move (possibly with a READY move prepended to it), not all initiating moves begin games, since some of the initiating moves serve to continue existing games or remind the partner of the main purpose of the current game again. Second, the place where games end or are abandoned is marked. Finally, games are marked as either occurring at top level or being embedded (at some unspecified depth) in the game structure, and thus being subservient to some top level purpose. The goal of these definitions is to give enough information to study relationships between game structure and other aspects of dialogue whilst keeping those relationships simple enough to code.

3.5 The Transaction Coding Scheme

Transaction coding gives the subdialogue structure of complete task-oriented dialogues, with each transaction being built up of several dialogue games and corresponding to one step of the task. In most Map Task dialogues, the participants break the route into manageable segments and deal with them one by one. Because transaction structure for Map Task dialogues is so closely linked to what the participants do with the maps, the maps are included in the analysis. The coding system has two components: (1) how route givers divide conveying the route into subtasks and what parts of the dialogue serve each of the subtasks, and (2) what actions the route follower takes and when.

The basic route giver coding identifies the start and end of each segment and the subdialogue which conveys that route segment. However, Map Task participants do not always proceed along the route in an orderly fashion; as confusions arise, they often have to return to parts of the route which have already been discussed and which one or both of them thought had been successfully completed. In addition, participants occasionally overview an upcoming segment in order to provide a basic context for their partners, without the expectation that their partners will be able to act upon their descriptions (for instance, describing the complete route as "a bit like a diamond shape ... but ... a lot more wavy than that ..."). They also sometimes engage in subdialogues which are not relevant to any segment of the route, sometimes about the experimental setup but often nothing at all to do with the task. This gives four transaction types: NORMAL, REVIEW, OVERVIEW, and IRRELEVANT.

Other types of subdialogues are possible (such as checking the placement of all map landmarks before describing any of the route, or concluding the dialogue by reviewing the entire route), but are not included in the coding scheme because of their rarity.

Coding involves marking where in the dialogue transcripts a transaction starts and which of the four types it is, and for all but IRRELEVANT transactions, indicating the start and end point of the relevant route section using numbered crosses on a copy of the route giver's map. The ends of transactions are not explicitly coded because, generally speaking, transactions do not appear to nest; for instance, if a transaction is interrupted to review a previous route segment, participants by and large restart the goal of the interrupted transaction afterwards. It is possible that transactions are simply too large for the participants to remember how to pick up where they left off. Note that it is possible for several transactions (even of the same type) to have the same starting point on the route.

The basic route follower coding identifies whether the follower action was drawing a segment of the route or crossing out a previously drawn segment, and the start and end points of the relevant segment, indexed using numbered crosses on a copy of the route follower's map.

4. Reliability of Coding Schemes

It is important to show that subjective coding distinctions can be understood and applied by people other than the coding developers, both to make the coding credible in its own right and to establish that it is suitable for testing empirical hypotheses. Krippendorff (1980), working within the field of content analysis, describes a way of establishing reliability which applies here.

4.1 Tests of reliability

Krippendorff argues that there are three different tests of reliability with increasing strength. The first is *stability*, also sometimes called test-rest reliability, or inter-test variance; a coder's judgments should not change over time. The second is *reproducibility*, or inter-coder variance, which requires different coders to code in the same way. The third is *accuracy*, which requires coders to code in the same way as some known standard. Stability can be tested by having a single coder code the same data at different times. Reproducibility can be tested by training several coders and comparing their results. Accuracy can be tested by comparing the codings produced in these same coders to the standard, if such a standard exists. Where the standard is the coding of the scheme's 'expert' developer, the test simply shows how well the coding instructions fit the developer's intention.

Whichever type of reliability is being assessed, most coding schemes involve placing units into one of n mutually exclusive categories. This is clearly true for the dialogue structure coding schemes described here, once the dialogues have been segmented into appropriately sized units. Less obviously, segmentation also often fits this description. If there is a natural set of possible segment boundaries which can be treated as units, one can recast segmentation as classifying possible segment boundaries as either actual segment boundaries or non-boundaries. Thus for both classification and segmentation, the basic question is what level of agreement coders reach under the reliability tests.

4.2 Interpreting reliability results

It has been argued elsewhere (Carletta, 1996) that since the amount of agreement one would expect by chance depends on the number and relative frequencies of the categories under test, reliability for category classifications should be measured using the kappa coefficient.[1] Even with a good yardstick, however, care is needed to determine from such figures whether or not the exhibited agreement is acceptable, as Krippendorff (1980) explains. Reliability in essence measures the amount of noise in the data; whether or not that will interfere with results depends on where the noise is and the strength of the relationship being measured. As a result, Krippendorff warns against taking overall reliability figures too seriously in favour of always calculating reliability with respect to the particular hypothesis under test. Using α, a generalised version of kappa which also works for ordinal, interval, and ratio-scaled data, he remarks that a reasonable rule of thumb for associations between two variables which both rely on subjective distinctions is to require $\alpha > .8$, with $.67 < \alpha < .8$ allowing tentative conclusions to be drawn. Krippendorff also describes an experiment by Brouwer in which English-speaking coders reached $\alpha = .44$ on the task of assigning television characters to categories with complicated Dutch names which did not resemble English words! It is interesting to note that medical researchers have agreed on much less strict guidelines, first drawn up by Landis and Koch (1977), who call $K < 0$ "poor" agreement, 0 to .2 "slight", .21 to .40 "fair", .41 to .60 "moderate", .61 - .80 "substantial", and .81 to 1 "near perfect". Landis and Koch describe these ratings as "clearly arbitrary, but useful benchmarks".

Krippendorff also points out that where one coding distinction relies on the results of another, the second distinction cannot be reasonable unless the first also is. For instance, it would be odd to consider a classification scheme acceptable if coders were unable to agree on how to identify units in the first place. In addition, when assessing segmentation, it is important to choose the class of possible boundaries sensibly. Although kappa corrects for chance expected agreement, it is still susceptible to order of magnitude differences in the number of units being classified, when the absolute number of units placed in one of the categories remains the same. For instance, one would obtain different values for kappa on agreement for move segment boundaries using transcribed word boundaries and transcribed letter boundaries, simply because there are so many extra agreed non-boundaries in the transcribed letter case. Despite these warnings, kappa has clear advantages over simpler metrics and can be interpreted as long as appropriate care is used.

1 The kappa coefficient (K) (Siegel and Castellan, 1988) measures pairwise agreement among a set of coders making category judgments, correcting for chance expected agreement.
$K = (P(A) - P(E))/(1 - P(E))$ where $P(A)$ is the proportion of times that the coders agree and $P(E)$ is the proportion of times that one would expect them to agree by chance.

4.3 Reliability of Move Coding

The main move and game cross-coding study involved four coders, all of whom had already coded substantial portions of the Map Task Corpus. For this study, they simply segmented and coded four dialogues using their normal working procedures, which included access to the speech as well as the transcripts. All of the coders interacted verbally with the coding developers, making it harder to say what they agree upon than if they had worked solely from written instructions. On the other hand, this is a common failing of coding schemes, and in some circumstances it can be more important to get the ideas of the coding scheme across than to control how it is done tightly.

4.3.1 Reliability of Move Segmentation. First, the move coders agree on how to segment a dialogue into moves. Two different measures of agreement are useful. In the first, kappa is used to assess agreement on whether or not transcribed word boundaries are also move segment boundaries. On average, the coders marked move boundaries roughly every 5.7 words, so that there were roughly 4.7 times as many word boundaries which were not marked as move boundaries as word boundaries which were. The second measure, similar to information retrieval metrics, is the actual agreement reached measuring pairwise over all locations where any coder marked a boundary. That is, the measure considers each place where any coder marked a boundary and averages the ratio of the number of pairs of coders who agreed about that location over the total number of coder pairs. Note that it would not be possible to define 'unit' in the same way for use in kappa because then it would not be possible for the coders to agree on a non-boundary classification. Pairwise percent agreement is the best measure to use in assessing segmentation tasks when there is no reasonable independent definition of units to use as the basis of kappa. It is provided for readers who are skeptical about our use of transcribed word boundaries.

The move coders reached $K = .92$ using word boundaries as units ($N = 4079$ [the number of units], $k = 4$ [the number of coders]); pairwise percent agreement on locations where any coder had marked a move boundary was 89% ($N = 796$). Most of the disagreement fell into one of two categories. First, some coders marked a READY move but the others included the same material in the move which followed. One coder in particular was more likely to mark READY moves, indicating either greater vigilance or a less restrictive definition. Second, some coders marked a reply, while others split the reply into a reply plus some sort of move which conveys further information not strictly elicited by the opening question (i.e., an EXPLAIN, CLARIFY, or INSTRUCT). This confusion was general, suggesting that it might be useful to think more carefully about the difference between answering a question and providing further information. It also suggests possible problems with the CLARIFY category, since unlike EXPLAIN and INSTRUCT moves, most CLARIFY moves follow replies, and since CLARIFY moves are intended to contain unelicited information. However, in general the agreement on segmentation reached was very good and certainly provides a solid enough foundation for move classification.

4.3.2 Reliability of Move Classification. The argument that move classification is reliable uses the kappa coefficient; units in this case are moves for which all move coders agreed on the boundaries surrounding the move. Note that it is only possible to measure reliability of move classification over move segments where the boundaries were agreed. The more unreliable the segmentation, the more data must be omitted. Classification results can only be interpreted if the underlying segmentation is reasonably robust.

Overall agreement on the entire coding scheme was good ($K = .83$, $N = 563$, $k = 4$), with the largest confusions between (1) CHECK and QUERY-YN, (2) INSTRUCT and CLARIFY, and (3) ACKNOWLEDGE, READY, and REPLY-Y. Combining categories, agreement was also very good ($K = .89$) for whether a move was an initiation type or a response or ready type. For agreed initiations themselves, agreement was very high ($K = .95$, $N = 243$, $k = 4$) on whether the initiation was a command (the INSTRUCT move), a statement (the EXPLAIN move), or one of the question types (QUERY-YN, QUERY-W, CHECK, or ALIGN). Coders were also able to agree on the subclass of question ($K = .82$, $N = 98$, $k = 4$). Coders could also reliably classify agreed responses as ACKNOWLEDGE, CLARIFY, or one of the reply categories ($K = .86$, $N = 236$, $k = 4$). However, coders had a little more difficulty ($K = .75$, $N = 132$, $k = 4$) distinguishing between different types of moves which all contribute new, unelicited information (INSTRUCT, EXPLAIN, and CLARIFY).

4.3.3 Reliability of Move Classification from Written Instructions. For a workshop sponsored by the University of Pennsylvania, three non-HCRC computational linguists and one of the original coding developers, who had not done much coding, move coded a Map Task dialogue from written instructions only, using just the transcript and not the speech source. Agreement on move classification was K=.69 (N=139, k=4). Leaving the coding developer out of the coder pool did not change the results ($K = .67, k = 3$), suggesting that the instructions conveyed his intentions fairly well. The coding developer matched the official Map Task coding almost entirely. One coder never used the CHECK move; when that coder was removed from the pool, $K = .73$ ($k = 3$). When CHECK and QUERY-YN were conflated, agreement was $K = .77$ ($k = 4$). Agreement on whether a move was an initiation, response, or ready type was good ($K = .84$). Surprisingly, non-HCRC coders appeared to be able to distinguish the CLARIFY move better than in-house coders. This amount of agreement seems acceptable given that this was a first coding attempt for most of these coders and was probably done quickly. Coders generally become more consistent with experience.

4.3.4 Reliability of Move Coding in Another Domain. Move coding is perhaps the level of coding most useful for work in other domains. To test how well the scheme would transfer, it was applied by two of the coders from the main move reliability study to a transcribed conversation between a hi-fi sales assistant and a married couple intending to purchase an amplifier. Dialogue openings and closings were omitted since they are well understood but do not correspond to categories in the classification scheme. The coders reached $K = .95$ ($N = 819, k = 2$) on the move segmentation task, using word boundaries as possible move boundaries, and $K = .81$ ($N = 80, k = 2$) for move classification. These results are in line with those from the main trial. The coders recommended adding a new move category specifically for when one conversant completes or echoes an utterance begun by another conversant. Neither of the coders used INSTRUCT, READY, or CHECK moves for this dialogue.

4.4 Reliability of Game Coding

The game coding results come from the same study as the results for the expert move cross-coding results. Since games nest, it is not possible to analyse game segmentation in the same way as was done for moves. Moreover, it is possible for a set of coders to agree on where the game begins and not where it ends, but still believe that the game has the same goal, since the game's goal is largely defined by its initiating utterance. Therefore the best analysis considers how well coders agree on where games start and, for agreed starts, where they end. Since game beginnings are rare compared to word boundaries, pairwise percent agreement is used.

Calculating as described, coders reached promising but not entirely reassuring agreement on where games began (70%, $N = 203$). Although one coder tended to have longer games (and therefore fewer beginnings) than the others, there was no striking pattern of disagreement. Where the coders managed to agree on the beginning of a game (i.e., for the most orderly parts of the dialogues), they also tended to agree on what type of game it was (INSTRUCT, EXPLAIN, QUERY-W, QUERY-YN, ALIGN, or CHECK) ($K = .86$, $N = 154$, $k = 4$). Although this is not the same as agreeing on the category of an initiating move because not all initiating moves begin games, disagreement stems from the same move naming confusions (notably, the distinction between QUERY-YN and CHECK). There was also confusion about whether a game with an agreed beginning was embedded or not ($K = .46$). The question of where a game ends is related to the embedding subcode, since games end after other games which are embedded within them. Using just the games for which all four coders agreed a beginning, the coders reached 65% pairwise percent agreement on where the game ended. The abandoned game subcode turned out to be so scarce in the crosscoding study that it was not possible to calculate agreement for it, but agreement is probably poor. Some coders have commented that the coding practice was unstructured enough that it was easy to forget to use the subcode.

To determine stability, the most experienced coder completed the same dialogue twice, two months and many dialogues apart. She reached better agreement (90%; $N = 49$) on where games began, suggesting that one way to improve the coding would be to formalise more clearly the distinctions which she believes herself to use. When she agreed with herself on where a game began, she also agreed well with herself about what game it was ($K = .88$, $N = 44$, the only disagreements being confusions between CHECK and QUERY-YN), whether or not games were embedded ($K = .95$), and where the games ended (89%). There were not enough instances of abandoned games marked to test formally, but she did not appear to use the coding consistently.

In general, the results of the game crosscoding show that the coders usually agree, especially on what game category to use, but when the dialogue participants begin to overlap their utterances or fail to address each other's concerns clearly, the game coders have some difficulty agreeing on where to place game boundaries. However, individual coders can develop a stable sense of game structure, and therefore if necessary, it should be possible to improve the coding scheme.

4.5 Reliability of Transaction Coding

Unlike the other coding schemes, transaction coding was designed from the beginning to be done solely from written instructions. Since it is possible to tell from the video what the route follower drew and when they drew it uncontroversially, reliability has only been tested for the other parts of the transaction coding scheme.

The replication involved four naive coders and the 'expert' developer of the coding instructions. All four coders were postgraduate students at the University of Edinburgh; none of them had prior experience of the Map Task or of dialogue or discourse analysis. All four dialogues used different maps and differently shaped routes.

To simplify the task, coders worked from maps and transcripts. Since intonational cues can be necessary for disambiguating whether some phrases such as "OK" and "right" close a transaction or open a new one, coders were instructed to place boundaries only at particular sites in the transcripts, which were marked with blank lines. These sites were all conversational move boundaries except those between READY moves and the moves which followed them. Note that such move boundaries form a set of independently derived units which can be used to calculate agreement on transaction segmentation. The transcripts did not name the moves or indicate why the potential transaction boundaries were placed where they were.

Each subject was given the coding instructions and a sample dialogue extract and pair of maps to take away and examine at leisure. The coders were asked to return with the dialogue extract coded. When they returned they were given a chance to ask questions. They were then given the four complete dialogues and maps to take away and code in their own time. The four coders did not speak to each other about the exercise. Three of the four coders asked for clarification of the OVERVIEW distinction, which turned out to be a major source of unreliability; there were no other queries.

4.5.1 Measures. Overall, each coder marked roughly a tenth of move boundaries as transaction boundaries. When all coders were taken together as a group, the agreement reached on whether or not conversational move boundaries are transaction boundaries was $K = .59$ ($N = 657$, $k = 5$). The same level of agreement ($K = .59$) was reached when the expert was left out of the pool. This suggests the disagreement is general rather than arising from problems with the written instructions. Kappa for different pairings of naive coders with the expert were .68, .65, .53, and .43, showing considerable variation from subject to subject. Note that the expert interacted minimally with the coders, and therefore differences were not due to training.

Agreement on the placement of map reference points was good; where the coders agreed that a boundary existed, they almost invariably placed the begin and end points of their segments within the same four centimeter segment of the route, and often much closer, as measured on the original A3 (296 x 420 mm.) maps. In contrast, the closest points which did not refer to the same boundary were usually five centimeters apart, and often much further. The study was too small for formal results about transaction category. For 64 out of 78 boundaries marked by at least two coders, the category was agreed.

4.5.2 Diagnostics. Because this study was relatively small, problems were diagnosed by looking at coding mismatches directly rather than by using statistical techniques. Coders disagreed on where to place boundaries with respect to introductory questions about a route segment (such as "Do you have the swamp?", when the route giver intends to describe the route using the swamp) and attempts by the route follower to move on (such as "Where do I go now?"). Both of these confusions can be corrected by clarifying the instructions. In addition, there were a few cases where coders were allowed to place a boundary on either side of a discourse marker, but the coders did not agree. Using the speech would probably help, but most uses of transaction coding would not require boundary placement this precise. OVERVIEW transactions were too rare to be reliable or useful and should be dropped from future coding systems.

Finally, coders had a problem with 'grain size'; one coder had many fewer transactions than the other coders, with each transaction covering a segment of the route which other coders split into two or more transactions, indicating that he thought the route givers were planning ahead much further than the other coders did. This is a general problem for discourse and dialogue segmentation. Greene and Cappella (1986) show very good reliability for a monologue segmentation task based on the 'idea' structure of the monologue, but they explicitly tell the coders that most segments are made up of two or three clauses. Describing a typical size may improve agreement, but might also weaken the influence of the real segmentation criteria. In addition, higher level segments such as transactions vary in size considerably. More discussion between the expert and the novices might also improve agreement on segmentation, but would make it more difficult for others to apply the coding systems.

5. Conclusions

Subjective coding has been described for three different levels of task-oriented dialogue structure, called conversational moves, games, and transactions, and the reliability of all three kinds of coding discussed. The codings were devised for use with the HCRC Map Task Corpus. The move coding divides the dialogue up into segments corresponding to the different discourse goals of the participants and classifies the segments into one of twelve different categories, some of which initiate a discourse expectation and some of which respond to an existing expectation. The coders were able to reproduce the most important aspects of the coding reliably, such as move segmentation, classifying moves as initiations or responses, and subclassifying initiation and response types. The game coding shows how moves are related to each other by placing into one game all moves which contribute to the same discourse goal, including the possibility of embedded games, such as those corresponding to clarification questions. The game coding was somewhat less reproducible but still reasonable. Individual coders can come to internally stable views of game structure. Finally, the transaction coding divides the entire dialogue into subdialogues which correspond to major steps in the participants' plan for completing the task. Although transaction coding has some problems, the coding can be improved by correcting a few common confusions. Game and move coding have been completed on the entire 128 dialogue Map Task Corpus; transaction coding is still experimental.

Game and move coding are currently being used to study intonation both in one-word English utterances (Kowtko, 1995) and in longer utterances across languages (Grice et al., 1995), the differences between audio-only, face-to-face, text-based, and video-mediated communication (Doherty-Sneddon et al., forthcoming; Newlands, Anderson, and Mullin, 1996), and the characteristics of dialogue where one of the participants is a non-fluent Broca-type aphasic (Merrison, Anderson, and Doherty-Sneddon, 1994). In addition, the move coded corpus has been used to train a program to spot the dialogue move category based on typical word patterns, in aid of speech recognition (Bird et al., 1995). The move categories themselves have been incorporated into a computational model of move goals within a spoken dialogue system in order to help the system predict what move the user is making (Lewin et al., 1993).

Acknowledgments
This work was completed within the Dialogue Group of the Human Communication Research Centre. It was funded by an Interdisciplinary Research Centre Grant from the Economic and Social Research Council (U.K.) to the Universities of Edinburgh and Glasgow and grant number G9111013 of the Joint Councils Initiative. Authors JC and AI are responsible for developing the transaction coding scheme and for carrying out the reliability studies; all authors contributed to the development of the move and game coding schemes. We would like to thank our anonymous reviewers for their comments on the draft manuscript.

References

Ahrenberg, Lars, Nils Dahlback, and Arne Jonsson. 1995. Coding schemes for studies of natural language dialogue. In *Proceedings of the AAAI 1995 Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 8–13, March.

Alexandersson, Jan, Elisabeth Maier, and Norbert Reithinger. 1995. A robust and efficient three-layered dialogue component for a speech-to-speech translation system. In *Proceedings of the Seventh European Meeting of the ACL*, pages 188–193.

Anderson, Anne H., Miles Bader, Ellen Gurman Bard, Elizabeth Boyle, Gwyneth Doherty, Simon Garrod, Stephen Isard, Jacqueline Kowtko, Jan McAllister, Jim Miller, Catherine Sotillo, Henry Thompson, and Regina Weinert. 1991. The HCRC Map Task Corpus. *Language and Speech*, 34(4):351–366.

Bird, Stuart, Sue Browning, Roger Moore, and Martin Russell. 1995. Dialogue move recognition using topic spotting techniques. In *ESCA Workshop on Spoken Dialogue Systems - Theories and Applications*, pages 45–48, May.

Carletta, Jean. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254, June.

Carletta, Jean, Amy Isard, Stephen Isard, Jacqueline Kowtko, Gwyneth

Doherty-Sneddon, and Anne Anderson. 1996. HCRC dialogue structure coding manual. Technical Report HCRC/TR-82, Human Communication Research Centre, University of Edinburgh, June.

Carlson, Lauri. 1983. *Dialogue Games: An Approach to Discourse Analysis.* D. Reidel.

Clark, Herbert and Edward Schaefer. 1989. Contributing to discourse. *Cognitive Science*, 13:259–294.

Condon, Sherri L. and Claude G. Cech. 1995. Functional comparison of face-to-face and computer-mediated decision-making interactions. In S. Herring, editor, *Computer-Mediated Conversation.* John Benjamins.

Doherty-Sneddon, Gwyneth, Anne H. Anderson, Claire O'Malley, Steve Langton, Simon Garrod, and Vicki Bruce. Forthcoming. Face-to-face and video mediated communication: A comparison of dialogue structure and task performance. Human Communication Research Centre.

Greene, John O. and Joseph N. Cappella. 1986. Cognition and talk: The relationship of semantic units to temporal patterns of fluency in spontaneous speech. *Language and Speech*, 29(2):141–157.

Grice, Martine, Ralf Benzmuller, Michelina Savino, and Bistra Andreeva. 1995. The intonation of queries and checks across languages: data from Map Task dialogues. In *Proceedings of the Thirteenth International Congress of Phonetic Sciences*, volume 3, pages 648–651, August.

Grice, Martine and Michelina Savino. 1995. Intonation and communicative function in a regional variety of Italian. In *Phonus 1*, pages 19–32. Institute of Phonetics, University of the Saarland.

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

Houghton, George. 1986. *The Production of Language in Dialogue: A Computational Model.* Ph.D. thesis, University of Sussex, April.

Kowtko, Jacqueline C. 1995. The function of intonation in spontaneous and read dialogue. In *Proceedings of the XIIIth International Congress of Phonetic Sciences*, volume 2, pages 286–289.

Krippendorff, Klaus. 1980. *Content Analysis: An introduction to its methodology.* Sage Publications.

Landis, J. R. and G. G. Koch. 1977. The measurement of observer agreement for categorical data. *Biometrics*, 33:159–174.

Lewin, Ian, Martin Russell, David Carter, Sue Browning, Keith Ponting, and Stephen Pulman. 1993. A speech based route enquiry system built from general purpose components. In *Proceedings of the 3rd European Conference on Speech Communication and Technology (EUROSPEECH-93)*, September. Also SRI Cambridge Technical Report CRC-033.

Merrison, Andrew John, Anne H. Anderson, and Gwyneth Doherty-Sneddon. 1994. An investigation into the communicative abilities of aphasic subjects in task oriented dialogue. Technical Report RP-50, Human Communication Research Centre, June.

Nagata, Masaaki and Tsuyoshi Morimoto. 1993. An experimental statistical dialogue model to predict the speech act type of the next utterance. In Katsuhiko Shirai, Tetsunori Kobayashi, and Yasunari Harada, editors, *Proceedings of the International Symposium on Spoken Dialogue*, pages 83–86.

Newlands, Alison, Anne H. Anderson, and Jim Mullin. 1996. Dialog structure and cooperative task performance in two CSCW environments. In J. Connolly, editor, *Linguistic Concepts and Methods in CSCW.* Springer-Verlag, chapter 5, pages 41–60.

Power, Richard J. D. 1979. The organisation of purposeful dialogues. *Linguistics*, 17:107–152.

Siegel, Sidney and N. J. Castellan, Junior. 1988. *Nonparametric Statistics for the Behavioral Sciences.* McGraw-Hill, second edition.

Sinclair, John M. and R. Malcolm Coulthard. 1975. *Towards an Analysis of Discourse: The English used by teachers and pupils.* Oxford University Press.

Sutton, Stephen, Brian Hansen, Terri Lander, David G. Novick, and Ronald Cole. 1995. Evaluating the effectiveness of dialogue for an automated spoken questionnaire. In *Proceedings of the AAAI 1995 Spring Symposium on Empirical Methods in Discourse Interpretation and Generation*, pages 156–161, March.

Walker, Marilyn and Steve Whittaker. 1990. Mixed initiative in dialogue: An investigation into discourse segmentation. In *Proceedings of the 28th Meeting of the ACL*, pages 70–78.