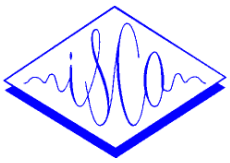

ETRW on Dialogue and Prosody
Veldhoven, The Netherlands
September 1-3, 1999

RELATING DIALOGUE GAMES TO INFORMATION STATE

Stephen Pulman

University of Cambridge Computer Laboratory
New Museums Site, Cambridge CB2 3QG;
and SRI International, Cambridge.
sgp@cl.cam.ac.uk

ABSTRACT

This paper[1] discusses the use of 'conversational' or 'dialogue games' as a basis for building dialogue systems. We give a tutorial overview of some recent attempts to relate the notion of a dialogue act to changes of information state of the participants in a dialogue. These attempts all distinguish some notion of 'grounded' or 'common' propositions. We raise the question as to whether these attempts might make the notion of dialogue game redundant, reducing it to an epiphenomenon arising out of the manipulation of information states. The answer to the question is no, not quite, yet.

1. DIALOGUE GAMES

The use of dialogue games and moves as a level of linguistic description in characterising what is happening in a dialogue has become ubiquitous, at least in computational approaches. A somewhat coarse-grained characterisation of what currently happens if you want to build a dialogue system for some application is as follows: firstly, collect a corpus of dialogues (by some means or other) which you think are characteristic of the kind of dialogues that will be entered into by the system you want to build. Secondly, decide on a repertoire of dialogue moves that you can use to annotate the dialogues so as to describe the types of linguistic transaction that will take place. This repertoire might be taken directly from an existing annotated corpus, like the Edinburgh Map Task ([15]); from an annotation scheme which is partly intended to serve as a standard resource for this purpose, like the DAMSL or similar schemes ([1]); or a special purpose one might be developed in order to exploit certain regularities in the domain of application, as for example in the Verbmobil or other similar projects ([22]). Thirdly, exploit a pool of cheap labour at your local university linguistics department to annotate as many of the dialogues as you can afford to. Finally, use the corpus of annotated dialogues to develop or to train a dialogue move recognition engine, and perhaps also some component to plan dialogue moves (although in most currently implemented systems this is usually hard-wired in a way defined by the dialogue task). The dialogue move handling components will be one of the most important in the resulting dialogue system for they have to figure out what stage of the dialogue we are at, and what to do next.

Along the way there are several choices and decisions to be made. Firstly, the set of move and game descriptors chosen must be capable of being applied to an utterance by virtue of some objective feature of it, or the context in which it is uttered. It is notoriously the case that annotating a dialogue is a task for which experience and training are required if there is to be sufficient agreement between annotators for the resulting corpus to be reliable. Some coding schemes, like the Edinburgh one, come with a coder's manual which gives a simple decision tree for the basic move categories, along with discussion and legislation for difficult cases. Even so, when you are actually in the position of annotating a dialogue, it is typically not long before questions arise about particular cases which are not covered by the documentation. For example, in annotating the Autoroute dialogues ([16, 17]) we came across several cases where silence was clearly being interpreted as assent, as in the following situation:

```
1. W     Where would you like to go?
2. C     From uh, I'd like to go to
         Southampton, from Swansea.
3. W     You want to go to Southampton
         from Swansea?
4. W     Please wait.
```

Using the Map Task move set, we would probably want to annotate this section as a Wh-question game with an embedded Checking game:

```
[WHQ
 WH-query 1. W     Where would you like to go?
 WH-reply 2. C     From uh, I'd like to go to
                   Southampton, from Swansea.
 [CHK
   CHECK    3. W   You want to go to Southampton
                   from Swansea?
   CLARIFY  ....
  ]
 ALIGN    4. W     Please wait.
]
```

But this would not be possible if we observed the letter of the law in applying the MapTask coding manual, since, although well aware of this problem, it has to assume that all relevant evidence will be available in the transcript. (NB Clark and Schaefer [4] count 'continued attention' as signalling acceptance. Since the Autoroute dialogues are telephone-based, continued attention is a matter of inference.)

Many other problems that can arise in applying a particular move scheme to a new domain are discussed in Larsson (1998). There

[1] Preparation of this paper was partly supported at SRI by the EU Trindi project, LE4-8314. The Autoroute dialogues are made available by permission of the Speech Research Unit, DERA Malvern, UK

is not merely the question of what label to use when describing a particular utterance, but also the perhaps prior question of how to chop utterances up into several units; whether one utterance can realise more than one move, and so on and so forth.

Notice that in deciding how to treat the Checking game, and the WHQ game in which it is embedded, we have already departed from the requirement that we should be guided by observable features of the utterance or the context. We have decided that the WHQ game is not over after the WH-query is met with a WH-reply, although this would be a simple and relatively reliable and observable criterion to apply. Rather, we have decided that the game is not properly over until the Checking sub-game has been completed. The reason we have decided this is that our intuition is that the information requested in the initial question, although it has been offered, has not really been accepted or 'grounded' ([4]) until the checking game is completed. The information that the caller wants to go from Southampton to Swansea is in the air but only becomes a fully fledged belief of the wizard at this point. In other words, we are making our decision based on what we know about the information states of the participants in the dialogue. The same is true of our assumption about the implicit clarification given by the caller's failure to issue a negative when the checking move is made. We reason that the wizard is expecting an answer which either confirms or rejects the check. It is a convention that silence can be construed as assent, or, to be more precise, that in the right circumstances not saying no can be taken as a yes. (There may not have been a long silence: just failure to take up a turn.)

Actually, the same thing is largely true of the decisions we make guided by the coding decision trees. Overt linguistic properties like syntactic form or intonation are important, but other factors like whether the information expressed can be assumed to be already known to a participant are also invoked.

Observations like these suggest that the notion of information state is primary when trying to determine dialogue moves. Dialogue moves serve to change information states. This is not really surprising, of course: the point of (most) dialogues is to express and alter the information states of the participants in them. Nor is it a novel observation: Hamblin ([12]), Isard ([13]) Stalnaker ([23]) and many others up to and including the current industry in dynamic semantic approaches to language have made the same point.

However, there have been very few attempts to systematically relate the notion of dialogue game to information state. The speech act derived 'Belief, Desire Intention' model of Cohen, Perrault, Allen and others ([5, 6]) attempted to do this indirectly, by relating the felicity conditions of particular speech acts to the mental states of the participants in a dialogue. But as many people have pointed out, speech acts are not the same as dialogue acts, and some extra machinery has to be added to achieve the updating function, as well as to achieve the link between separate but linked utterances (e.g. question-answer pairs) that is missing from the speech act tradition ([21]).

In the following sections I will describe and illustrate two current theories which do make an attempt to elaborate the connection between dialogue move and information state. I will then conclude by discussing the question of whether a sufficiently elaborated theory of information state change would make the postulation of dialogue acts as distinct theoretical entities unnecessary. In other words, if we have a way of describing structured information states, and if we have an understanding of the semantic and pragmatic properties of the relevant linguistic constructs (e.g. wh-questions, fragment ellipsis, focus, etc.) do we need anything else to characterise what goes on in dialogue?

2. POESIO AND TRAUM

Poesio and Traum ([19, 20]), building on work done by James Allen's group in the TRAINS project, have elaborated a theoretical and descriptive framework with which to characterise dialogues. They go on to try to use their framework to provide an axiomatisation of some dialogue acts. The basic components of their approach are as follows.

The set of dialogue acts used are those advocated in the Discourse Resource Initiative ([1]), deriving originally from the speech act tradition but extended so as to remedy some of the deficiencies of speech act theory as an empirical account of what happens in real dialogue.

In the DRI approach, dialogue acts are classified as forward- or backward looking, depending on whether they are initiating or responding acts. Acts are hierarchically arranged, inheriting some of their properties. The forward looking acts include:

```
Statement
  Assert
  Reassert
  Other-Statement
Influencing-addressee-future-action
  Open-option
  Directive
    Action-Directive
    Info-Request
Committing-speaker-future-action
  Offer
  Commit
Conventional
  Opening
  Closing
Explicit-performative
Exclamation
```

Backward-looking acts include:

```
Agreement
  Accept
  Accept-part
  Maybe
  Reject
  Reject-part
  Hold
Answer
```

Poesio and Traum also adopt the view that 'grounding' - the process by which information becomes accepted by both parties - is carried out by special backward-looking dialogue acts, including:

```
Understanding-act
 Signal-non-understanding
```

```
Signal-understanding
  Acknowledge
  Repeat-rephrase
  Completion
Correct-missspeaking
```

In addition to the DRI move set, Poesio and Traum postulate a set of 'turn-taking' acts, which are to do with management and control of the dialogue: `take-turn`, `keep-turn`, `release-turn`, and `assign-turn`.

Poesio and Traum model each conversational participant using the notation of Discourse Representation Theory ([14]). At any given time a conversational participant will usually have some 'discourse units' that have been grounded (in G), and some that are currently pending, awaiting grounding in UDUS (ungrounded DUs). The current discourse init (CDU) is the first member of this latter list.

G, DU1, DU2, DU3, UDUS, CDU
G = … DU1= … DU2= … DU3= … UDUS=⟨ DU3,DU1⟩ (CDU=first(UDUS)=DU3)

All new information gets first added to a DU, which triggers obligations of various types on the part of the addressee, and updates concerning the intentions of the initiator. After clarifications and acknowledgements grounded information moves into G.

Here is a (simplified) example of the state of G for a participant after a dialogue like:

A: There is an engine at Avon.
B: Okay.
A: It is hooked to a boxcar.

In the following, utterance discourse referents are u1, u2 etc, and 'conversational event' referents are ce1, ce2, etc.

u1,u2,u3, ce1, ce2, ce3, K1, K2
u1: utter(A,'There is an engine at Avon') ce1: assert(A,B,K1) K1 = x,w,e | engine(x), Avon(w), at(x,w) generate(u1,ce1) u2: utter(B,'Okay') ce2: accept(B,ce1) generate(u2,ce2) u3: utter(A,'It is hooked to a boxcar') c3: assert(A,B,K2) K2= y,u | boxcar(y), hooked-to(y,u), u = x generate(u3,ce3)

Notice that the representation includes information about what was said ('utter'), how it was interpreted in terms of dialogue moves ('assert', 'accept'), and which utterance corresponds to which move ('generate'). This enables agents to reason about the occurrence of dialogue acts of different types as well as about the content of the utterances that gave rise to them. (Note that 'rhetorical relations', and conditions like those described by Grosz and Sidner ([11]) are also represented by Poesio and Traum in this manner although we have omitted them here.)

Poesio and Traum are now able to characterise various different dialogue acts in terms of their effects on different components of the information state of a conversational participant. In stating these effects they use various intentional predicates which are reminiscent of the Cohen, Perrault and Allen treatment of speech acts:

Try(A,P)	an event of A trying to do P
Achieve(A,P)	A brings about P
Address(A,P)	A considers and responds to P
Bel(A, K)	A believes the proposition expressed by DRS K
Int(A,P)	A intends to do or bring about P
Option(A,P)	P is something A is able to perform
SCCOE(A,B,K)	A is socially committed towards B that K is the case
Obliged(A,B,P)	A has an obligation to B to do P

The last two conditions arise as a result of linguistic or social convention.

Now update rules can be formulated for the different types of act they consider. The rules are intended to capture the effects of the acts in terms of conditions that the information states must meet if the act is to apply, along with the changes or updates to information states that will ensue when it does apply. For brevity, we show the various rules in the form:

Conditions
Updates

Utterances (locutionary acts) trivially update the participants' information states by adding the fact that the sentence has been uttered.

(1) A utters P
Add to CDU: Utter(A,P)

Poesio and Traum assume that it is a general property of core speech acts like 'Statement', 'Influencing-addressee-future-action' etc. that they impose an obligation on the addressee to perform an 'Understanding-act', and in some cases to 'address' the issue raised by the act:

(2) (for all core Acts) K contains Act(A,B,P)
Add to K: Obliged(B,A,Understanding-act(B,P))

(3) (for some core Acts) K contains Act(A,B,P)
Add to K: Obliged(B,A,Address(B,P))

(Note that K here is intended to include the whole current DRS: G, UDU, etc.) Most backward-looking acts will remove obligations added by the previous forward-looking act.

Some of the specific further update conditions associated with acts are as follows:

(4) G contains Statement(A,B,K)
Add to G: SCCOE(A,B,K)

If A states something, then he is committed to it (by linguistic and social conventions) being the case.

(5) G contains Assert(A,B,K)
Add to G: Try(A,Bel(B,K))
G contains Accept(B,K) ⇒ Add SCCOE(B,A,K) to G

If A asserts K, he is trying to get B to believe it, and if B accepts K, then he is also conversationally committed to it. Since Assert is a subtype of Statement the previous update rule will also apply.

Most backward-looking acts will remove obligations added by the previous forward-looking act. In addition, an Accept act following a Statement will trigger the antecedent of the conditional in rule (5)

In simplified form the sequence of updates for the section of dialogue earlier will be as follows. We begin with the simple fact of the utterance, via (1):

G,UDU,CDU,DU1

G=[]

DU1=
u1
Utter(A,"There's an engine at Avon")

CDU=DU1
UDU=⟨ DU1 ⟩

Recognising u1 as an act of assertion will update the CDU to include:

ce1, K1

ce1: Assert(A,B,K1)

K1=
x,w
engine(x)
Avon(w)
at(x,w)

generate(u1,ce1)

This will in turn trigger both the general and the specific updates associated with assertion, to add to CDU:

Obliged(B,Understanding-act(B,K1))
Obliged(B,Address(B,K1))

When the utterance of 'Okay' is processed as acknowledging ce1, ce1 is grounded (i.e. moved from CDU into G). This triggers the antecedent of rule (5) leading to the addition to G of:

SCCOE(A,B,K1)
Try(A,Bel(B,K1))

Poesio and Traum analyse acknowledgements and acceptances as one of a special set of Grounding acts. Grounding acts themselves do not get added to G: they simply cause an earlier discourse unit to be added to G:

(6) CDU contains Acknowledge(A,DU_n)
Add DU_n to G; remove DU_n from UDU
remove CDU from UDU

The effect of B's utterance of 'Okay' in the above dialogue is first to add the discourse units:

u2

u2 = Utter(B,"Okay")

and:

ce2

ce2 = Acknowledge(B,DU1)

The update rule (6) for acknowledgements is then triggered, causing the whole discourse unit DU1 derived from 'There is an engine at Avon' to become fully grounded.

Note that for Poesio and Traum, grounding and in fact all other aspects of conversational structure are treated uniformly as different types of act. Grounding has various linguistic consequences

apart from the obvious ones. For example, Poesio ([18]) points out that in general, cross-speaker anaphora involving assertions that are not grounded is rather unnatural:

```
A There is an engine at Avon
B (i) ?? It is red.
B (ii) Right. It is red.
```

A's use of an indefinite suggests that a new individual unknown to B is being introduced. It is therefore odd when B immediately uses a pronoun to refer to it, as in (i). However, if B acknowledges A's utterance first, as in (ii), the information becomes grounded and it is more natural to refer to a now mutually known entity with a pronoun.

3. GINZBURG

In [9, 10] Ginzburg presents a model of dialogue which aims to show how the notion of common ground fits into an update semantics for dialogue. He shares the assumption of the dialogue act tradition that conversational interaction can be characterised in terms of a small number of primitive move types which set up a restricted set of options or obligations at any given point in a dialogue. However, unlike Poesio and Traum the acts that Ginzburg postulates are not (obviously) derived from the speech act theory tradition. Furthermore, Ginzburg tries to relate the properties of dialogue much more to the semantic properties of individual utterances, particularly questions.

As for Poesio and Traum, the information states of participants in a discourse are structured: a participant has a set of private beliefs and goals ('unpublicised mental situation') as well as a 'gameboard', a semi-shared set of beliefs and propositions. A gameboard is structured into at least three components: a set of commonly agreed on FACTS; a (partially ordered) set of 'questions under discussion' (QUD) specifying the things currently open for discussion, which need not correspond to questions in the strict sense; and a LATEST-MOVE, specifying the content of the most recent conversational move.

The options available to a speaker at any point in a dialogue are conditioned by the content of his LATEST-MOVE, and his QUD. If some Q is topmost in his QUD, then a speaker can provide any information specific to Q. By information specific to Q Ginzburg means either information about Q (i.e. that might partly or completely resolve the question one way or another) or some further question on which Q might depend, as for example B's utterance in:

```
A: Are you going to the lecture today?
B: Who is speaking?
```

The notion of 'resolving' a question is discussed extensively by Ginsburg: essentially, a fact or piece of information in the 'unpublicised mental situation' resolves a question if it implies either a positive or a negative answer to the question which is complete enough for the current purposes. Being 'complete enough' will depend both on the local context of the utterance, and the interests and goals of the participants.

A speaker can also react to the content of LATEST-MOVE with whatever options are available. The options will vary according to the type of thing that is there, so that although Ginzburg does not discuss this in detail, this is one of the points at which the usual notion of dialogue games enters the picture. Ginzburg only discusses assertions and queries.

When a participant B's LATEST-MOVE contains something of the form 'A query Q', then he has two options: (a) accept Q for discussion, adding Q to the top of QUD, and producing a relevant utterance, or (b) reject Q for discussion, uttering a rejection phrase ('Never mind about Q', etc). In this latter case both Q and 'whether B discusses Q' get added to QUD, but the latter is the topmost item.

If B accepts Q for discussion then things will proceed so as to allow what Ginzburg calls QUD-DOWNDATING to apply. The QUD-DOWNDATING principle is stated: If Q is currently topmost in QUD, and P is a fact that either (a) resolves Q relative to the 'unpublicised mental situation', or (b) indicates that no information about Q can be provided then Q can be removed from QUD, P is added to FACTS and if (a) applies, also added to FACTS is the 'fact that P resolves Q'.

If B rejects A's query, and if A accepts B's rejection then Q will be removed from QUD via QUD-DOWNDATING. Alternatively A can discuss with B the rejection and if successful in overcoming it then 'whether B discusses Q' will be removed from A's and B's QUD and Q will finally be discussed as originally intended.

When B's LATEST-MOVE contains something of the form 'A assert P', then B again has two choices. He can *accept* P, in which case P will be added to B's FACTS. This acceptance counts as information that 'resolves' the top element in A's QUD, namely the question whether P. Now A can apply QUD-DOWNDATING, adding P to FACTS and removing 'whether P' from his QUD. The fact P is now grounded: it appears in both A's and B's FACTS.

The other option for B after A's assertion is to discuss 'whether P'. In this case B adds to the top of his QUD 'whether P' and produces a relevant utterance. This will usually lead to a 'side-sequence' or clarification subdialogue, perhaps something like:

```
A: There's an engine at Avon (= P)
B: Avon?
A: Yeah.
B: Okay.
```

Here A's response to B's query counts as resolving 'whether P' and QUD-DOWNDATING will apply. B's acceptance confirms that P is now in FACTS and is grounded.

Notice that in Ginzburg's framework, grounding is essentially regarded as involving the additional raising by any utterance of an implicit question about the interpretation of that utterance. This is why 'whether P' appears on top of B's QUD. If that implicit question can be resolved (in Ginzburg's sense) by information in the addressee's unpublicised mental situation, then the utterance can be accepted and thus become grounded for both participants. In the semantic framework that Ginzburg is assuming (a version of situation theory) this in fact falls out (almost) automatically. Propositions are modelled as relations between situations and 'states of affairs' which, to radically simplify, we can think of as represented by expressions like:

⟨at,engine,avon,true⟩.

Question-meanings are modelled as relations between situations and 'abstracts', which, again we can think of in a radically simplified way as expressions like:

λX.⟨at,engine,X,true⟩

('Where is the engine?')

or:

λX.⟨at,engine,Avon,X⟩

('Is the engine at Avon?')

i.e. what is the truth value of 'engine at Avon'?

Utterances are modelled as abstracts too, with various restrictions that must be met by the situation: e.g. 'there's an engine at Avon' might be something like:

λUSXY. ⟨assert(S,at(X,Y,true))⟩

where U must be a situation in which S is the speaker, X is an engine, and Y is named Avon. Thus any utterance can be correlated with a corresponding set of question-meanings, namely, what you get by abstracting over one or more of its components.

Ginzburg also postulates a move of implicit acceptance:

ACCOMMODATE-QUESTION: if Q is topmost in QUD, it is permissible to add the fact *that there exists a fact positively resolving Q* to FACTS. Optionally, remove Q from QUD.

This move corresponds to the silent move in our example dialogue earlier. It would apply in the current example if B's response had been something like:

```
A: There's an engine at Avon (= P)
B: Let's move it to Danville.
```

Of course, using this rule is not always safe. It may be the case that the other participant does not accept your utterance but for some reason has not made this explicit. In this case there will be a mismatch between what you regard as commonly accepted FACTS and what the other participant regards as commonly accepted FACTS. But in general if neither acceptance nor discussion of the currently salient QUD is offered, it is safe to assume it is accepted.

We can formulate rules to capture the updates and actions possible according to Ginzburg's theory, in the style of those of Poesio and Traum:

QUD-DOWNDATING:

Q topmost in QUD

Given info P resolving Q
Remove Q from QUD
Add P to facts

Accept-query:

LATEST-MOVE = A query Q

Produce relevant utterance
Add Q to QUD

Reject-query:

LATEST-MOVE = A query Q

Produce rejection phrase
Add Q to QUD
Add 'whether B discusses Q' to QUD

Accept-assertion:

LATEST-MOVE = A assert P

(Optionally) produce acceptance phrase
Add P to FACTS

Query-assertion:

LATEST-MOVE = A assert P

Produce query phrase
Add 'whether P' to QUD

ACCOMMODATE-QUESTION:

Q topmost in QUD

Add '∃P.P resolves Q' to FACTS
(Optionally) remove Q from QUD

We can illustrate Ginzburg's approach informally using the following example (which will assume is complete):

```
1. w: Where would you like to go?
2. c: Edwinstowe
3. w: Edwinstowe?
4. c: Yes
```

After utterance 1, Q= which(X,go(c,X)):

```
W: [LATEST-MOVE = W query Q]
   [QUD         = Q         ]

C: [LATEST-MOVE = A query Q]
```

C applies Accept-query, producing utterance 2, P = go(c,edwinstowe), the resolved form of the ellipsis:

```
W: [LATEST-MOVE = C assert P]
   [QUD         = Q         ]

C: [LATEST-MOVE = C assert P]
   [QUD         = Q         ]
```

W applies Query-assertion, with the query phrase being 'Edwinstowe?', resolved to whether(P) = go(c,edwinstowe)?

```
W: [LATEST-MOVE = W query whether(P)]
   [QUD         = whether(P), Q     ]

C: [LATEST-MOVE = W query whether(P)]
   [QUD         = Q         ]
```

C applies Accept-query, and produces a relevant utterance, 'yes', which we will resolve as true(P):

```
W: [LATEST-MOVE = C assert true(P)]
   [QUD         = whether(P), Q   ]

C: [LATEST-MOVE = C assert true(P)]
   [QUD         = whether(P),Q    ]
```

At this point, both W and C can apply QUD-DOWNDATING twice. 'True(P)' resolves 'whether(P)', and 'P' resolves 'Q'. Thus the final gameboards are:

```
W: [LATEST-MOVE = C assert true(P)]
   [QUD         =                 ]
   [FACTS       = Q, true(P)      ]
C: [LATEST-MOVE = C assert true(P)]
   [QUD         =                 ]
   [FACTS       = Q, true(P)      ]
```

Ginzburg is able to show how this mechanism can explain some otherwise puzzling observations. Take the following examples:

```
(1) A: Who left the Institute before 5. Why?
(2) A: Who left the Institute before 5.
    B: Why?
```

In the first, the elliptical why question can be interpreted to mean 'Why did they leave?', whereas in the second the only natural interpretation is more like 'Why are you asking that question?' His explanation is that in the first example, the speaker is using ACCOMMODATE-QUESTION to 'help himself' to the fact that there is some fact that positively resolves the question (i.e. that the presupposition associated with the question is satisfied). This grounds the question and the speaker is then able to continue to raise a second question that asks partly about the implicit information resolving the first question: namely, why did the people whose identification constitutes an answer to the first question leave the Institute at that time.

In the second case speaker B has not grounded the utterance. Therefore the top element of QUD is the question itself. B's utterance is therefore taken to be an instance where he is applying 'Accept-query' but producing an utterance on which the query depends rather than something which resolves it immediately.

4. COMPARISON

It is clear that there are many points of similarity between these two approaches. Both maintain a distinction between facts that are merely in the air, and those that are in some sense accepted by both participants. Poesio and Traum's 'G' and Ginzburg's 'FACTS' are clearly doing the same job. Both regard dialogue as imposing various conventions on the participants, by virtue of the utterance of some linguistic form in a particular context. In the case of Poesio and Traum this is modelled explicitly in terms of update rules belonging to particular dialogue acts which create obligations of particular kinds on the part of the addressee of the act. In Ginzburg's case it is less directly encoded but enforced by virtue of the different options that are available to a discourse participant when, say, a query is the LATEST-MOVE, or a question form is the top-most element in QUD.

In [7] an extensive empirical comparison is made of versions of both the theories discussed here, transcribing several Autoroute dialogues in full. One of the questions raised by this attempt is the extent to which dialogue moves and acts are anything other than a convenient label for a package of information state updates. In other words, if we have a fully worked out theory of the update of information states (although neither of the preceding claim to be this) could we dispense with explicit dialogue acts altogether?

Recall that in the speech act tradition, at least some proponents argued that illocutionary acts could be replaced or defined away by an explicit account of reasoning about belief and intention ([5]). Poesio and Traum, although to some extent within this tradition, do not claim that their treatment makes explicit dialogue acts redundant (see [20], p213), although to some extent their later paper is trying to take some steps along this route. However, it is actually rather difficult to see how they could dispense with explicit acts, since the update rules themselves are stated in terms of records of particular dialogue acts having occurred or even being grounded. In other words, information states themselves contain information about dialogue acts: it would be circular to attempt to define dialogue acts away in terms of information states.

In Ginzburg's approach, there is less overt reliance on dialogue acts and thus more apparent prospect of doing away with them. We are in fact not very far from being able to describe most of what was happening in the Edwinstowe dialogue using just a simple distinction between grounded and ungrounded material, and some purely semantic manipulations, although there is not space or time to demonstrate this here. We do however have to assume some rather powerful inferential machinery capable of giving content to notions like 'resolves', 'depends on', or 'relevant utterance'. Also, the rules we stated for updates as they stand still required as their triggering condition that some previous utterance had been classified as an assertion or a query. In order to completely define away the notion of dialogue act we would have to reduce these triggering conditions to predicates on information states.

This is quite a plausible aim to go for, in fact. Recall that many of the diagnostics that would distinguish between annotating a syntactic yes-no interrogative form as expressing the dialogue act 'yes-no question' (as opposed to 'check') rely on asking whether the information expressed is inferrable from anything mentioned earlier in the dialogue. If it is not, then the utterance is very likely to be a genuine yes-no question. If it is, the utterance is much more likely to be a check - part of a grounding game or side-sequence. This diagnostic can clearly be converted to a (rather complex) predicate on information states which define the different types of acts: similarly, other acts could be identified with tests on pairs or possibly sequences of information states representing the information available before and after the relevant utterance.

There is obviously a lot of work to do to spell out the details of this particular reductionist programme, and so dialogue games will be with us for some time yet. My own view, however, is that this route looks like the best way to go in order to be able to develop more flexible and responsive dialogue systems in the future.

5. REFERENCES

1. Allen, J. and Core, M. *DAMSL: Dialogue Markup in Several Layers*, draft contribution for the Discourse Resource Initiative, 1997.
2. H. Bunt. Dynamic interpretation and dialogue theory. In M.M. Taylor, D.G. Bouwhuis, and F. Neel, editors, *The structure of multi-modal dialogue*, volume 2. Amsterdam John Benjamins, 1997.
3. L. Carlson. *Dialogue Games*. Dordrecht: D. Reidel Publishing Co., 1983.
4. Clark, H. and Schafer, E. F. *Contributing to Discourse*, Cognitive Science, 13, 259-294.
5. P.R. Cohen and H.J. Levesque. Rational interaction as the basis for communication. In P.R. Cohen, J. Morgan, and M.E. Pollack, editors, *Intentions in Communication*. MIT Press, 1990.
6. P Cohen and C.R. Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3(3):177–212, 1979.

7. Cooper, R. and Larsson, S. and Matheson, C. and Poesio, M. and Traum, D. *Coding Instructional Dialogue for Information States*, Trindi project deliverable D1.1. February 1999.

8. Cooper, R. and Larsson, S. *Dialogue Moves and Information States*, Technical Report, University of Gothenberg, 1998.

9. Ginzburg, J. *An update semantics for dialogue* in H. Bunt, R. Muyskens, G. Rentier (eds). Proceedings of the First International Workshop on Computational Semantics, Tilburg, 1994.

10. Ginzburg, J. *Interrogatives: Questions, Facts, and Dialogue*, in Handbook of Contemporary Semantic Theory, ed. S. Lappin, Oxfor: Blackwell Publishers Ltd. 385-422.

11. Grosz, B. J., and C. L. Sidner. *Attention, Intentions, and the Structure of Discourse* Computational Linguistics" 1986, 12(3):175–204.

12. C.L. Hamblin. Mathematical models of discourse. *Theoria*, 37:130–155, 1971.

13. S. Isard *Changing the context* in E. Keenan (ed) Formal Semantics of Natural Language, Cambridge University Press, p 287–296.

14. Kamp, H. and Reyle, U. *From Discourse to Logic* Kluwer Academic Publishers, 1993.

15. Kowtko, J. and Isard, S. and Doherty, G. M. *Conversational Games within Dialogue*, Research Paper HCRC/RP-31, Human Communication Research Centre, Edinburgh University.

16. I. Lewin, M. Russell, D. Carter, S. Browning, K. Ponting, and S.G. Pulman. A speech-based route enquiry system built from general-purpose components. In *Eurospeech '93: Proceedings of the 3rd European Conference on speech communication and technology*, volume 3, pages 2047–2050, 1993.

17. R.K. Moore and S.R. Browning. Results of an exercise to collect 'genuine' spoken enquiries using woz techniques. In *Proceedings of the Institute of Acoustics 14 6*, pages 613–620, 1992.

18. Poesio, M. *Cross-speaker anaphora and dialogue acts*, in Proc of the ESSLLI workshop on Mutual Knowledge, Common Ground, and Public Information, H. Rieser and W. Heydrich (eds), Saarbrücken.

19. Poesio, M. and Traum, D. *Conversational Actions and Discourse Situations*, Computational Intelligence 13(3), 309-347.

20. Poesio, M. and Traum, D. *Towards an Axiomatization of Dialogue Acts*. in TWLT 13, Formal Semantics and Pragmatics of Dialogue, ed. J. Hulstijn and A. Nijholt, Proceedings of the 13th Twente Workshop on Language Technology, Twente, Holland, 1998.

21. S. G. Pulman, 1997, *Conversational Games, Belief Revision and Bayesian Networks*, CLIN VII: Proceedings of 7th Computational Linguistics in the Netherlands meeting, Nov 1996, ed. J. Landsbergen et al., 1-25.

22. N. Reithinger and E. Maier. Utilizing statistical dialogue act processing in verbmobil. In *Proceedings of 33rd ACL, Cambridge Mass.*, pages 116–121, 1996.

23. R. Stalnaker. *Assertion* in P. Cole (ed) Syntax and Semantics vol 9: Pragmatics, 315-332, NY: Academic Press.