Artificial Intelligence and Law **8:** 233–254, 2000.

A method for the computational modelling of dialectical argument with dialogue games

T. J. M. BENCH-CAPON, T. GELDARD and P.H. LENG
LIAL – Legal Informatics at Liverpool, Department of Computer Science, The University of Liverpool, Liverpool, England

Abstract. In this paper we describe a method for the specification of computational models of argument using dialogue games. The method, which consists of supplying a set of semantic definitions for the performatives making up the game, together with a state transition diagram, is described in full. Its use is illustrated by some examples of varying complexity, including two complete specifications of particular dialogue games, Mackenzie's DC, and the authors' own TDG. The latter is also illustrated by a fully worked example illustrating all the features of the game.

Key words: argument, dialogue games, legal information systems, persuasion, speech acts

1. Introduction

The purpose of this paper is to describe a method for the specification of computational models of dialectical argument. The basis of these models will be the notion of a formal dialogue as a context in which arguments can be evaluated, deriving from Hamblin (1970). Such formal systems have often been called *dialogue games*; *dialogue* because they take place between two people, and *games* to emphasise that within the dialogue there are a limited number of options available to the participants ("moves"), and that the use of these options are governed by a set of rules. Such dialogue games are normative, in that they prescribe what can be done in the dialogue, rather that attempting to be descriptive of actual practice. They are intended, however, to reflect practice by being an idealisation of it, and by allowing the modelling of actual dialogues within the rules of the game.

Models in the form of dialogue games have a number of attractions, chief of which are:

1. They provide a precise description of what can occur in the dialogue, and thus are suitable for implementation in a computer system;
2. They identify the component utterances in a dialogue, and supply semantics for these utterances;
3. They relate these semantics to a context.

The last is important: recall the remark of Wittgenstein:

> If you do not keep the multiplicity of language games in view you will perhaps be inclined to ask questions like: “What is a question?” – Is it the statement that I do not know such and such, or the statement that I wish the other person would tell me . . . ?” (Wittgenstein 1953, p. 12)

which stresses that we should not seek *the* meaning of the word, but rather investigate its use in the variety of contexts in which it is used.

The notion of dialogue game has been used in formal logic (e.g., Mackenzie 1979), informal logic (e.g., Walton 1989), and discourse analysis (e.g., Carlson 1983). More importantly for our current purpose, dialogue games have also been used widely in AI and Law: for example the influential paper of Gordon (1994), which used a game to model the process of legal pleading; the work of Prakken and Sartor (e.g., 1997), which has used games to analyse argumentation; and the recent thesis of Lodder (1998), which also explores the role of dialogue games and provides an excellent summary of their application to AI and Law.

Dialogue games also have relevance for Computer Science, addressing the need for description of the interaction between computers and other computers and between computers and humans. Such interactions are now often generalised as *agent-to-agent* communication, where an agent may be a computer system or a human, and communication is described in terms of *performatives*, communicative acts intended to produce some particular effect or response from the agent to whom it is directed. Leading work here is represented by KQML (Knowledge Query and Manipulation Language, Finin et al., 1994), which provides an extensible set of performatives for agent-to-agent communication. The performatives of KQML have been given context independent semantics in Labrou (1996), but elsewhere it has been argued (Barbuceanu and Fox 1995) that context is required to co-ordinate the performatives into meaningful conversations. It is just this coordinating context that modelling as dialogue games provides.

From this we conclude that the computational modelling of dialectical argument as dialogue games can give useful insights, applicable to AI and Law and more generally also, regarding:

1. The process of argument, as in Gordon (1994);
2. The logic of argument; as in Prakken and Sartor (1997);
3. The design of human-computer interaction with such systems; as in Bench-Capon et al. (1991).

Given the utility of such computational models, we believe it is desirable to have a sound method for constructing them. In this paper we will advance such a method. In Section 2 we describe the method, illustrating it with reference to the interaction with standard consultative expert systems. In Section 3 we use the method to describe the method of explanation presented in Bench-Capon et al. (1993). None of these dialogues, however, can be considered dialectical; rather they should be described as *information seeking* in the terminology of Walton and Krabbe (1995). In Section 4 we turn attention to what they call *persuasion dialogues* or *critical discussion*. First we define the well known game, DC, given in Mackenzie (1979), and then in Section 5, we apply the method to our own game, Toulmin Dialogue Game (TDG), based on the argument schema of Toulmin (1958). In section 6 we illustrate the use of TDG by presenting an example dialogue, and modelling it in terms of TDG. Section 7 makes some observations on TDG, and Section 8 contains some concluding remarks.

2. Specification Method

In this section we will describe our method for specifying interactions, illustrating it by reference to the standard consultative expert system model exemplified by MYCIN (Buchanan and Shortliffe 1984) and its descendents.

Our method involves the production of two documents:

1. A State Transition Diagram (STI), which has nodes corresponding to the participant in the dialogue whose turn it is to issue a performative, and arcs corresponding to the performatives that can be issued in that state;
2. For each performative represented in the STD, a specification of its semantics comprising:
   a) A description of the performative (including any side effects);
   b) Preconditions that must hold for the act to be performed. Such preconditions may apply to the speaker, the hearer or the context;
   c) Postconditions specifying the state of speaker, hearer and context immediately after the act has been performed;
   d) Completion conditions indicating the final state of the speaker, hearer and context after all the intentions have been fulfilled, possibly after further conversation has occurred.

The specification of the semantics follows the style of specification of KQML performatives in Labrou (1996). The use of the STD follows the method for the specification of *conversation classes* in Barbuceanu and Fox (1995). The STD can be said to define a class of conversations, since there will be many different legal traversals of the STD. A legal conversation will be a particular traversal, and so be an instance of the class defined by the STD.

In terms of dialogue games the STDs define the class of interactions which represent legal playings of the game, and the set of performatives the moves in the game.

We now show how the method is used by reference to the interactions with standard consultative expert systems.

2.1. INTERACTION WITH A SIMPLE CONSULTATIVE EXPERT SYSTEM

In this interaction the user of the system initiates the interaction by posing some question. The system will then solicit information from the user until there is sufficient information for it to supply an answer to the original question. The STD for this interaction is shown in Figure 1. As the diagram shows there are only two

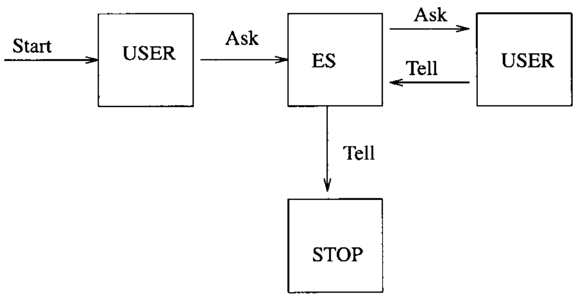

Figure 1. State transition diagram for simple expert system.

performatives involved here: the posing of some question, and the answering of the question. The user is represented by two states since there are distinct options available depending on whether it is the first or some subsequent move in the dialogue.

We can supply the semantics for these moves as follows:

Ask (Proposition)
Description: S wants to know whether Proposition is true
Preconditions: Proposition must be part of the vocabulary of H
S has control of the dialogue
Postconditions: H has control of the dialogue
Completion Conditions: H has issued a tell(Proposition)
Tell (Proposition)
Description: S states whether Proposition is true or false
Preconditions: H has issued an ask(Proposition)
S has a proof of Proposition, or its negation
S has control of the dialogue
Postconditions: H has control of the dialogue
Completion Conditions: None

Here *S* stands for the issuer of the performative, *H* for the recipient of the performative and *Proposition* is a parameter representing the propositional content of the question/answer.

One interesting point we need to consider – particularly if we wish to implement the system – is what choices need to be made at given points in the dialogue, since if there are choices, we need to have some principle or strategy which will enable us to decide which option to take. From this specification it is immediately clear that the user has no choice whatsoever; at any given state there is only one move available. The expert system does have a choice, but in practice this is resolved by issuing the tell as soon as the preconditions are satisfied. There are thus no strategic questions to resolve.

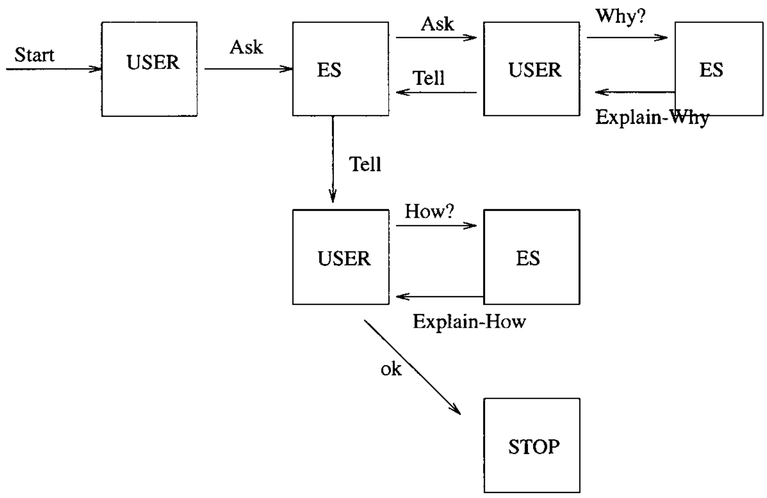

Figure 2. State transition diagram for expert system with explanation.

2.2. INTERACTION WITH AN EXPERT SYSTEM WITH EXPLANATION

For a more interesting example, consider the interaction with an expert system which includes the standard explanation facilities of *why* (i.e., why are you asking this question?) and *how* (i.e., how did you establish that answer?). The STD for such an interaction is shown in Figure 2.

This interaction has five new moves, two for each kind of explanation and one for the user to terminate the session. The semantics of these are as follows:

why (Proposition)?
Description: User wishes to know why a question is being asked
Preconditions: System has issued an ask (Proposition)
User has control of the dialogue
Postconditions: System has control of the dialogue
Completion Conditions: System has issued an explain-why.
Explain-why (Proposition)
Description: System prints rule currently under consideration
Preconditions: User has issued a why (Proposition)
System has control of the dialogue
Postconditions: User has control of the dialogue
Completion Conditions: None
How (Proposition)
Description: User wishes to know how Proposition was proven
Preconditions: System has issued a tell (Proposition)
User has control of the dialogue
Postconditions: System has control of the dialogue

Completion Conditions: System has issued an explain-how (Proposition)
Explain-how
Description: System prints a proof trace
Preconditions: User has issued a how (Proposition)
System has control of the dialogue
Postconditions: User has control of the dialogue
Completion Conditions: None
OK
Description: User quits the session
Preconditions: User has control of the dialogue
User has issued an ask which has been completed.
Postconditions: The session is finished
Completion Conditions: None

Note that in this interaction certain moves are only available to one or other of the parties to the dialogue. Only the user can request explanation, or terminate the session. We call such a game *asymmetric*, in that we can distinguish two *roles*, each with their own associated moves. In an information seeking dialogue such as this, one player will be the information seeker and the other the information provider. From this diagram we can see that the system (the information provider) has again only one choice which can be resolved as described for the system without explanation. The user (the information seeker) now, however, has far more choice, and needs to be able to resolve these choices. The user might choose a fixed strategy, such as minimal time through dialogue which would require that explanations be always avoided, or a strategy which always requested explanations when available (useful in training or debugging mode). Typically, however, the user will use an intermediate strategy, sometimes accepting the system output, and sometimes seeking explanation. The need to choose a strategy means that the user may require training or experience. In general user choice means that the user must learn how to get the most out of the system. where the system has choice we need to be able to supply rules – either deterministic or heuristic – which will give effect to some chosen strategy.

3. Explanation Based on Toulmin's Argument Scheme

The simple interactions we have discussed above have not been explicitly presented as dialogue games. In this section we will present the information seeking game of Bench-Capon et al. (1993), which was explicitly said to be a dialogue game. This is interesting both as an example of the use of the method to give a more precise reconstruction of a game initially presented in a less formal manner, and because the concepts used in the game prepare the ground for the later full specification of TDG.

The envisaged role of this game was to provide a more sophisticated explanation of the reasoning of a logic based expert system than is possible using the standard

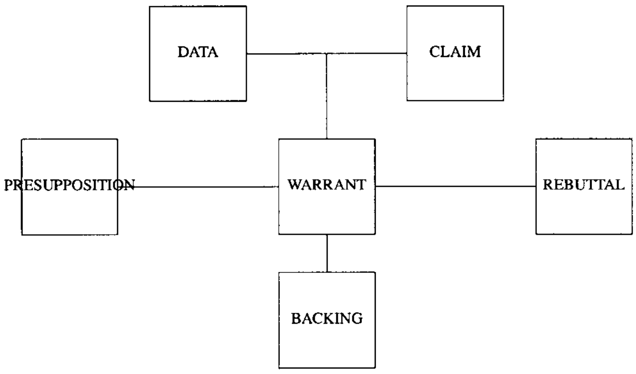

Figure 3. Toulmin's argument schema.

how query. It was intended to come into play after a consultation: in terms of Figure 2 the conversation can be seen as initiated by the *how* query in that interaction.

In this game explanations are based on the structure of argument proposed by Toulmin (1958). This structure, as modified in Bench-Capon et al. (1992), is represented by the schema show in Figure 3.

The key point about Toulmin's schema is that it is able to differentiate the parts played by different kinds of premise. The *claim* is the proposition that is under consideration. Thus *data* and *warrant* are conventional premises, the warrant being of the form *Antecedent* ⇒ *Claim* and the data being the antecedent of the warrant. The *presupposition* represents premises that are usually tacit, assumed background knowledge common to both parties unless explicitly challenged. The *rebuttal* is designed to accommodate defeasibility: it need not be shown false, but if true will defeat the argument. The *backing* is intended to justify the warrant by an appeal to an accepted authority. These distinctions enable the explanation to be presented in a more natural and informative manner that a straightfoward, unstructured, list of premises can allow.

The game elicits information supporting the claim which was the original answer from the expert system, structured according to this schema. For a full discussion of the schema, and its merits for explanation, see Bench-Capon et al. (1991). The STD for this game is given at Figure 4.

In this game the following moves are available to the user in respect of a claim under examination:

- Why? asking for supporting data
- So? Given a claim and supporting data, what is the warrant?
- Presupposing? Given a claim, are there any presuppositions that need to be satisfied?
- Unless? Given a claim is there any potential rebuttal?
- On account of? Given a warrant, what is the source of that warrant?
- OK – terminates the session

The system has its own set of moves:

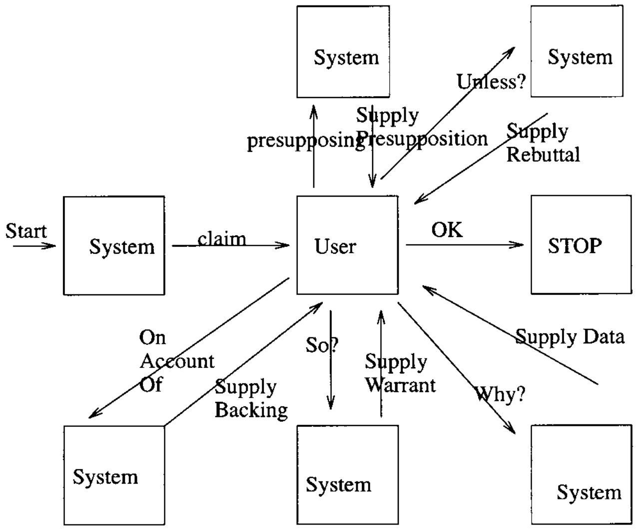

Figure 4. State transition diagram for Toulmin explanation game.

- Supply data – response to a Why?
- Supply warrant – response to a So?
- Supply presupposition – response to a Presupposing?
- Supply backing – response to an On Account Of?
- Supply rebuttal – response to an Unless?

We will not give the semantics for these moves here: instead we will wait until the full Toulmin game is described in the Section 6. However, we will remark at this point that it can be seen from the diagram that the system never has to make a choice between two possible moves. It was this forced character of the system play that made it possible to implement the explanation facility without any need to consider possible strategies and policies to be used by the system.

4. A Persuasion Dialogue – Mackenzie's DC

All the games so far have been information seeking games. Our aim, however, is to model dialectical interaction – namely persuasion dialogues and critical discussion (Walton and Krabbe 1995, p. 66). These games are symmetrical, in that both players can make exactly the same set of moves, should the situation demand. In particular players can switch roles so that at various times either may be the proponent or opponent of a claim under consideration. In this section we give the specification of a well known game which *is* an example of a persuasion dialogue, DC, first described in Mackenzie (1979).

MacKenzie's dialogue game DC is a symmetric two-person game the rules of which are intended, in particular, to prohibit circularity in the conduct of an argument. Central to DC is the concept of a *commitment*, expressing a proposition which one or both of the participants asserts or accepts to be true. As in the course of games players become committed to various propositions, and abandon these commitments, the game requires the maintenance of a *commitment store* for each participant, which is a record of the current state of their commitments. Changes to these commitment stores result from the application of postconditions of certain moves. Note that DC is a *non-cumulative* system, in that commitments may be withdrawn in the course of the argument, so propositions can be removed from, as well as added to, the commitment stores.

Figure 5 expresses the principal moves available in DC in the form of a state transition diagram. Essentially, there are six distinct states which can arise, each of which can symmetrically be attained by either participant; these are named by Mackenzie as Wilma and Bob, but here, more prosaically, we use A and B. In the initial state, the participant in control of the dialogue may advance a proposition (P), withdraw a previous commitment (W), ask a question (Q), challenge a proposition (C), or ask for an apparent contradiction to be resolved. Two different forms of the resolution demand (R1 and R2) are differentiated in the figure. The states defined in the figure express the possible developments which can ensue from each of these moves. Note that the game has no termination state; although DC prevents circularity in the argument, it does not prohibit new propositions being advanced at any stage so an argument can be continued indefinitely.

The specification of the moves describing DC incorporate both rules of the dialogue and the rules which determine the state of the commitment stores of each participant. The game requires a strict alternation of moves, so each move must have a precondition requiring S to have control of the dialogue, and a postcondition giving control to H. To avoid repetition, these are not included explicitly in the semantics below.

P: Proposition (P)
Description: S asserts the truth of some statement (which may include truth-functional compositions of statements, such as ‘not P’, ‘if P then Q’, etc.)
Preconditions: P must not be a statement to which both S and H are currently committed
Postconditions: Both S and H become committed to P
Completion Conditions: None

Q: Question (P)
Description: S enquires as to the truth of a statement P
Preconditions: None
Postconditions: None
Completion Conditions: The reply by H either asserts P, asserts ‘not P’, or withdraws commitment to P

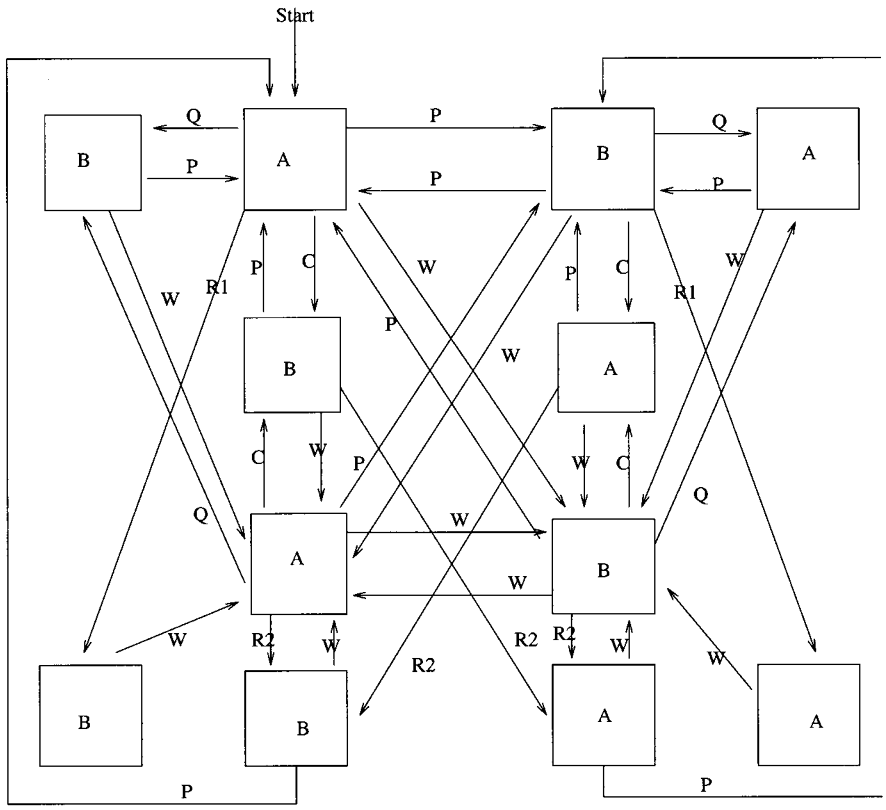

Figure 5. State transition diagram for MacKenzie's DC.

C: Challenge (P)
Description: S asks for a justification of the truth of a statement P
Preconditions: P cannot be a conditional whose consequent is an immediate consequence of its antecedent
Postconditions: H is committed to P; S is not committed to P; S is committed to 'Why P?'
Completion Conditions: H asserts Q, where Q is a statement not under challenge by S, giving rise to the commitment 'if Q then P'; or H withdraws P; or H invites S to Resolve a conditional whose consequent is P

W: Withdraw (P)
Description: S withdraws a commitment to the truth of a statement P
Preconditions: P cannot be a conditional whose consequent is an immediate consequence of its antecedent
Postconditions: S is no longer committed to P
Completion Conditions: H may invite S to Resolve a conditional whose consequent is P

R1: Resolve (P)

Description: S invites H to resolve an inconsistent conjunction of statements P

Preconditions: P is a conjunction of statements which are immediately inconsistent, and to which H is committed

Postconditions: None

Completion Conditions: H withdraws one of the conjuncts of P

R2: Resolve (P)

Description: S invites H to resolve a conditional P

Preconditions: P is of the form 'if Q then R' where Q is a conjunction of statements to which H is committed, R is an immediate consequence of Q, and H has challenged or withdrawn commitment to R

Postconditions: None

Completion Conditions: H withdraws one of the conjuncts of Q, or H confirms commitment to R

In general, in the above, postconditions of the moves define the state of the commitment stores arising, while those rules of the game that are not implicit in the state transition diagram are explicated in the preconditions and completion conditions. Note in particular the postconditions of a challenge, which include the storage of a commitment to the challenge. The completion conditions of the same move, which prohibit a responder from asserting a statement which is under challenge, provide the essential device by which circularity is prevented in the argument.

We have given a specification of DC using our method here because it is a well known game which has been used for modelling dialectical interaction on computers: see, for example Pilkington et al. (1993). However, we have stated elsewhere (Bench-Capon et al. 1992) that we find the resulting style of dialogues, which is very tied to that of a proof in formal logic, too stilted for modelling realistic dialogues. In the next section we will describe our game – TDG – which is an extension of the information seeking game presented in Section 3, which we believe to be very much more suited to modelling the sorts of argument found in practice.

5. Toulmin Dialogue Game (TDG)

In this section we will describe TDG, a persuasion dialogue game based on Toulmin's argument schema, presented in Figure 3. The point of using this schema is the structure it provides for an argument, in that instead of having only premises, and so using the flat, undifferentiated structure of formal logic, it allows us to ascribe different roles to premises, by dividing them into data, warrants, presuppositions and rebuttals, and allows for an appeal to authority through the backing. The purpose of this game is to extend the information seeking game described in Section 3 into a persuasion game, which can realise the full possibilities of dialectical interaction.

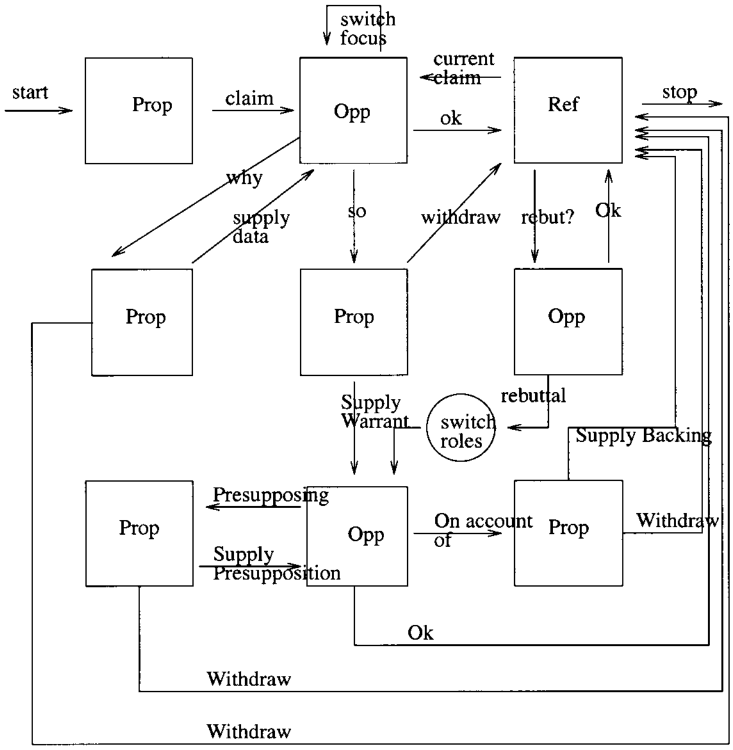

Figure 6. State transition diagram for TDG.

The game is symmetric, in that all the moves can be made by either of the participants, but at any given time the players will have different roles: one will be the proponent of the current claim and one will be the opponent of the current claim. At particular times the current claim will change, and the players will switch roles. We also have the notion of a *referee* conceived of as a third agent in the game, the role of which is to enforce implicit commitments. The STD for the game is shown in Figure 6.

In specifying the semantics of the moves of TDG we make use of – in addition to the commitment stores familiar from DC – the notion of a *claim stack.* "Stack" here is used in the standard sense of a "last in, first out" data structure, and is used to record the claims made during the course of the game. When a claim is made, implicitly or explicitly, it is pushed onto the stack; when it is resolved it is popped from the stack. We now specify the semantics for the moves: here *O* stands for the player opposing the current claim, *P* for the player proposing the current claim, and *Ref* for the referee, *C*, *D* and *S* are propositional variables.

claim (C)
Description: P asserts that C
Preconditions: P has control of the dialogue

Postconditions: O has control of the dialogue
C is pushed onto the claim stack
P is committed to C
Completion Conditions: C is popped from the claim stack
why (C)
Description: O seeks data supporting C
Preconditions: O has control of the dialogue
C is top of claim stack
Postconditions: P has control of the dialogue
Completion Conditions: C is not top of claim stack
OK (C)
Description: O accepts C
Preconditions: O has control of the dialogue
C is top of claim stack
Postconditions: C is popped from the claim stack
O is committed to C
O is not committed to not C
If not C is on claim stack, it is removed
Referee has control of the dialogue
Completion Conditions: None.
So (C)
Description: O requests the warrant for C
Preconditions: O has control of the dialogue
O is not committed to if D then C, for any D for which he is not committed to −D.
C is top of claim stack
Postconditions: P has control of the dialogue
Completion Conditions: C is not top of the claim stack
Presupposing (C)
Description: O requests the presupposition of C
Preconditions: O has control of the dialogue
If D then C is top of claim stack
Postconditions. P has control of the dialogue
Completion Conditions: If D then C is popped from the claim stack
On Account of (C)
Description: O requests the backing for the warrant of C
Presupposition: O has control of the dialogue
If D then C is top of claim stack
P has issued a supply warrant (C)
Postconditions: P has control of the dialogue
Completion Conditions: If D then C is popped from the claim stack.
Supply Data (C)
Description: P asserts that D and that D supports C

Preconditions: P has control of the dialogue
O has issued a Why (C)
C is top of the claim stack
Postcondition: P is committed to D
D is pushed on the claim stack
O has control of the dialogue
Completion Conditions: D is popped from the claim stack

Supply Warrant (C)
Description P asserts that If D then C
Preconditions: P has control of the dialogue
O has issued a So (C)
C is top of the claim stack
Postconditions: P is committed to If D then C
If D then C is pushed on to the claim stack
O has control of the dialogue
Completion Conditions: If D then C is popped from the claim stack

Supply Presupposition (C)
Description: P asserts that S
Preconditions: P has control of the dialogue
O has issued a presupposing (C)
If D then C is top of the claim stack
Postconditions: P is committed to S
P is committed to If not S then not C
S is pushed on to the claim stack
O has control of the dialogue
Completion Conditions: S is popped from the claim stack

Supply backing (C)
Description: P says that B is the authority for his argument for C
Preconditions: P has control of the dialogue
O has issued an on account of (C)
P has issued a supply warrant (C)
If D then C is top of the claim stack
Postconditions: R has control of the dialogue
O is committed to If D then C
If D then C is popped from the claim stack
Completion Conditions: None

Withdraw (C)
Description: P withdraws his commitment to C
Preconditions: P has control of the dialogue
C is top of the claim stack
Postconditions: C is popped from the claim stack
P is not committed to C
R has control of the dialogue

Completion Conditions: None

Switch Focus (C)

Description O wishes to consider a claim not currently top of the claim stack

Preconditions: C is not top of claim stack
C is on the claim stack
O has control of the dialogue
O is not committed to C

Postconditions: C is moved to top of claim stack
O has control of the dialogue

Completion Conditions: None

Current Claim (C)

Description: The referee passes control to the opponent of the current claim.

Preconditions: C is top of the claim stack
Ref has control of the dialogue
Player A is committed to C

Postconditions: Player B has control of the dialogue

Completion Conditions: None

End

Description: The referee terminates the dialogue

Preconditions: Ref has control of the dialogue
The claim stack is empty

Postconditions: The dialogue terminates

Completion Conditions: None

Rebut? (C)

Description: Player is invited to rebut an implicit commitment

Preconditions: Ref has control of the dialogue
C is top of claim stack
Player is not committed to C
Player is committed to if D then C for some D
Player is committed to D

Postconditions: Player has control of the dialogue
Player is opponent, other player is proponent

Completion Conditions: C is not top of claim stack

Rebuttal (C)

Rebuttal is the most complicated of the moves, and is perhaps best seen as a sequence of other moves:

a. Issuer becomes proponent, other players become opponent
b. Issuer claims −C
c. Other Player issues why(−C)
d. Issuer supplies data, D
e. Other player issues a switch focus (−C)
f. Other player issues so (C)
g. Issuer supplies warrant if D then −C

All the commitment stores are up-dated to reflect these moves, and the claim stack is also modified as if these moves were made explicitly.
This gives the semantics of Rebuttal as:
Rebuttal (C)
Description: Player provides a rebuttal, D, of C
Preconditions: Player has control of the dialogue
Player is not committed to C
Other player is committed to C
Postconditions: D is pushed on to the claim stack
Player is committed to $-$C
Player is committed to D
C is pushed on to claim stack
Player is committed to if D then $-$C
If D then $-$C is pushed onto claim stack
Player is proponent, other player is opponent
Opponent has control of the dialogue
Completion Conditions: $-$C is no longer on the claim stack

6. Example Dialogue Modelled in TDG

To show how we can use the game specified above to model the kinds of dialogue in which we are interested, consider the following dialogue concerning a question of inheritance. In the example upper case letters in brackets indicate propositional variables representing the propositions advanced, which will be used to refer to the statements below. Moves are labelled by player and the number of the move. Some explanatory annotations are given in italics.

A1: Alice should inherit John's money (I)
This is the initial claim
B1: Why?
The first question simply seeks the grounds for the claim, the data in our schema
A2: It seems only fair (F)
Some grounds arc advanced
B2: OK, but so what?
Here two moves are made. The truth of the data is accepted, but its relevance to the argument is questioned
A3: If it is fair that someone should inherit, they should inherit (F $\rightarrow$ I)
A rule which would licence the conclusion is advanced
B3: What's your authority for that?
Here a justification for the rule is sought. Note that here we do not seek a justification in terms of a logical argument for the rule, but rather in terms the authority from which the rule derives – making use of the backing in our schema

A4: I haven't got one.

No authority is available, and so the rule is withdrawn

B4: So why do you think Alice should inherit? (I)

Now there is no reason to think that F is a reason for I, a different reason is sought.

A5: She is his next of kin (K)

Another reason is supplied

B5: Suppose she is. Why should she get the money?

Here player B does not accept the truth of the data, but wishes to defer consideration of it, while exploring the connection between K and I

A6: The next of kin should inherit (K $\rightarrow$ I)

The relevant rule is supplied

B6: Is that always so?

Player B suspects that the rule is not always applicable and seeks the elements of the context which make it applicable to this case

A7: John died intestate. (W)

The background assumption is given

B7: True, but what's your authority

Player B accepts that W is true, but now wishes to know the authority for the rule

A8: Intestate Person's Act 1765.

This time player A has an authority

B8: Why do you say Alice is his next of kin? (K)

Player B accepts the rule but questions the premise, asking for grounds that support it.

A9: They were married a month ago. (M)

A supplies the grounds

B9: So they were. I agree she is his next of kin, but she murdered him. (S, S $\rightarrow -$I)

B accepts the truth of the grounds, and that there is a rule entailing K. On the face of it this would establish I. But B is aware of an exceptional circumstance which can defeat the argument.

A10: What's that got to do with it?

Player A now wishes to know the authority for the rule implicit in B's rebuttal. Note that it is now B who is the proponent of the claim $-I$

B10: No one shall profit from their crime – Blimplock (LCJ) Carter vs Carter, 1856.

B supplies an authority from case law

A11: Why do you say she murdered him?

Player A is forced to accept the rule, but can question the premise

B11: I read about the case in the Times. (R)

B supplies his data for S

A12: OK, OK. Alice shouldn't get a penny. ($-$I)

Player accepts the truth of R, and that R implies S, and so is constrained to accept that −I.

The intention here is to present natural dialogue, but one which shows that there are different ways of probing for information and challenging statements which require the distinction amongst premises that we find in the argument schema.

We can model this dialogue using the moves in TDG as follows. Note that some utterances in the example dialogue need to be expanded into several moves in the dialogue game.

A1 claim(I)
B1 why(I)
A2 Supply-data(I)
B2 i) ok(F); ii) current-claim(I); iii) so(I)
A3 Supply warrant(I)
B3 On Account of(I)
A4 i) Withdraw (F $\rightarrow$ I); ii) current-claim(I)
B4 why(I)
A5 supply data(I)
B5 i) switch focus(T); ii) so(I)
A6 supply-warrant(I)
B6 presupposing(I)
A7 Supply presupposition(I)
B7 I) ok(W); ii) current-claim(I); iii) on-account-of(I)
A8 i) supply-backing(I); ii) current-claim(I)
B8 i) switch focus(K); ii) why(K)
A9 supply-data(K)
B9 i) ok(M); current claim (K); iii) ok(K); iv) rebut?(I); v) rebuttal(I)
A10 on-account of(−)
B10 i) supply-backing(−I); ii) current claim (S)
A11 why(S)
B11 supply data(S)
A12 i) ok(R); ii) current claim (S); iii) ok(S); iv) rebut? (−I); v) ok(−I)

To illustrate further the operation of the dialogue game, the table in Appendix 1 lists all the moves and the turns on which they are completed, together with their effects on the control of the dialogue, the roles of the participants, the participant's commitment stores and the claim stack.

7. Observations on TDG

We believe that TDG represents a game with a set of moves rich enough to model reasonably realistic dialogues. The richness is essential if we are to capture elements of organisation and context integral to real arguments; more parsimonious games such as DC may capture the logic, but lose the other elements. Moreover the game has been specified in a precise way which facilitates debate about particular elements, or extensions to the game to capture new features. This last is an important point; we are not arguing that TDG is in any way canonical, but rather that it provides an extensible core of performatives derived from Toulmin's popular analysis, which could be adapted in a number of ways.

Apart from facilitating discussion, the form of specification in which we have presented TDG here provides a firm basis for implementation. Indeed a prototype, discussed in Bench-Capon (1998), has been implemented. This prototype will be used to explore the strengths and limitations of models of dialogue using TDG.

Some of the limitations of TDG are apparent without investigation:

1. It does not easily allow for the discussion of priorities amongst rules, and as such relies on the goodwill of the participants to reach a consensus on such points;
2. It does not permit arguments about backings of warrants; backings offered are assumed to be trustworthy and to licence the warrants they are claimed to support.
3. It does not allow for ungrounded claims. An obvious extension would be to permit a move by which a claim is supported by authority rather than some datum.
4. It highlights the need for heuristics to control the dialogue; this is less of a problem if both participants are human, but if one of them is intended to be a computer system, work needs to be done to ensure that choices between moves are sensible and productive. The notion of heuristics is important: it suggests that an additional layer is required above the protocol layer in models such as that advanced in Prakken (1995), which will contain these heuristics and tell us what it is to argue *well*.

Despite these limitations, TDG we feel that holds out the prospect for representing a wide range of realistic arguments.

8. Conclusions

The main purpose of this paper has been to argue for a method for the specification of dialogue games. The method produces:

1. A State Transition Diagram showing how the moves are co-ordinated into a dialogue.
2. Semantic definitions for these moves giving preconditions, postconditions and completion conditions.

We believe the method to have the advantages of straightforward applicability, transparency and precision, which allows it to form the basis of modelling any form of dialectical interaction. Its usefulness is not intended to be confined to those who want to build computational models: we commend it equally to linguists and the like who wish to model interactions in a formal or semi-formal way. The closeness to implementation of the specification will then provide an added bonus.

We have illustrated our method with a number of examples, culminating a full account of a particular game, TDG. We see TDG not as *the* answer, but rather as a basis for comparison, discussion, extension and adaptation. We do, however, see the method of explicit representation as moves in dialogue games and their relationships as the way forward for any analysis of dialogue intended to be modelled computationally.

Appendix 1: Table of Speech Acts in TDG Example

Note that the table only records changes: a blank cell indicates that there have been no changes from the previous cell.

Move	Comp	Control	Prop	Opp	A	B	Stack
0	–	A	A	–	–	–	
A1 c(I)	A12v	B			I		I
B1 w(I)	A2	A					
A2 sd(I)	B2i	B			I, F		F, I
B2i ok(F)	B2i	R				F	I
B2ii cc(I)	B2ii	B					
B2iii so(I)	A7	A					
A3 sw(I)	A4i	B			I, F, F→I		F→I, I
B3 oa(I)	A4i	A					
A4i w(F→I)	A4i	R			I, F		I
A4ii cc(I)	A4ii	B					
B4 w(I)	A5	A					
A5 sd(I)	B9iii	B			I, F, K		K, I
B5i sf(I)	B5i	B					I, K
B5ii so(I)	A6	A					
A6 sw(I)	A8i	B			I, F, K, K→I		K→I, I, K
B6 p(I)	A8i	A					
A7 sp(I)	B7i	B			I, F, K, K→I, W	W, K→I, I, K	
B7i ok(W)	B7i	R				F, W	K→I, I, K
B7ii cc(I)	B7ii	B					
B7iii oa(I)	A8i	A					
A8i sb(I)	A8i	R				F ,W, K→I	I, K
A8ii cc(I)	A8ii		B				
B8i sf(K)	B8i	B					K, I
B8ii w(K)	A9	A					
A9 sd(K)	B9i	B			I, F, K K→I, W, M		M, K, I
B9i ok(M)	B9i	R				F, W, K→I, W	K, I
B9ii cc(K)	B9ii	B					
B9iii ok(K)	B9iii	R				F, W, K→I, M, K	I
B9iv r?(I)	B9v	B					
B9v rl(I)	A12v	B	B	A			
B9v c(−I)	A12v	A				F, W,K→I, M, K, −i	−I, I

Move	Comp	Control	Prop	Opp	A	B	Stack
B9v w(−I)	B9v	B					
B9v sd(−I)	A12iv	A				F, W, K→I, M, K, −I, S	S, −I, I
B9v sf(−I)	B9v	A					−I, S, I
B9v so(−I)	B9v	B					
B9v sw (−I)	B10i	A				F, W, K→I, M, K, −I, S, S→−I	S→−I, S, −I, I
A10 oa(−I)	B10i	B					
B 10i sb(−I)	B10i	R			I, F, K, K→I, W, M, S→−I		S, −i, I
B10ii cc(S)	B10ii	A	B	A			
A11 w(S)	B11	B					
B11 sd(S)	A12i	A				F, W, K→I, M, K, −I, S, S→−I, R	R, S, −I, I
AI2i ok(R)	A12i	R			I, F, K, K→I, W, M, S→−I, R		S, −I, I
A12ii cc(S)	A12ii	A					
A12iii ok(S)	A12iii	R			I, F, K, K→I, W, M, S→−I, R, S		−I, I
A12iv r?(−I)	A12v	A					
A12v accept(−I)	A12v	R			I, F, K, K→I, W, M, S→−I, R, S, −I		[]
STOP							

References

Barbuceanu, M. & Fox, M. S. 1995. COOL: A Language for Communication in Multi Agent Systems. In Lesser, V. (ed.), *Proceedings of the First International Conference on Multiagent Systems*, MIT Press, Cambridge, Mass.

Bench-Capon, T. J. M., Lowes, D., & McEnery, A. M. 1991. Using Toulmin's Argument Schema to Explain Logic Programs. *Knowledge Based Systems* **4**(3), 177–83.

Bench-Capon, T. J. M., Dunne, P. E. & Leng, P. H., 1992. A Dialogue Game for Dialectical Interaction with Expert Systems. *12th Annual Conference on Expert Systems and Their Applications*, Avignon, EC2.

Bench-Capon, T. J. M., Coenen, F. P., & Orton, P. 1993. Argument Based Explanation of the British Nationality Act as a Logic Program. *Computers, Law and AI* **2**(1), 53–66.

Bench-Capon, T. J. M. 1998. *Specification and Implementation of Toulmin Dialogue Game.* Submitted to JURIX 98.

Buchanan, B. G. and Shortliffe, E. H. 1984. *Rule-Based Expert Systems*. Addison Wesley.

Carlson, L. 1983. *Dialogue Games: An Approach to Discourse Analysis*, Reidel, Dordrecht.

Finin, T., Fritzon, R., McKay, D., & McEntire, R. 1994. KQML as an Agent Communication Language. In *Proceedings of CIKM94*.

Gordon, T. E. 1994. The Pleadings Game: An Exercise in Computational Dialectics, *Artificial Intelligence and Law* **2**(4.

Hamblin, C. L. 1970. *Fallacies.* Methuen, London.

Labrou, Y. 1996. *Semantics for an Agent Communication Language*. Ph.D. Thesis, University of Maryland, Baltimore.

Lodder, A. R. 1998. *DiaLaw: On Legal Justification and Argument Games.* Ph.D. Thesis, University of Maastricht.

MacKenzie, J. D. 1979. Question-Begging in Non-Cumulative Systems. *Journal of Philosophical Logic* **8**: 159–177.

Pilkington, R. M, Hartley, J. R, Hintze, D., & Moore, D. J. 1993. Learning to Argue and Arguing to Learn: An Interface for Computer Based Dialogue Games. *Journal of Artificial Intelligence in Education* **3**(3): 275–29.

Prakken, H. 1995. From Logic to Dialectics in Legal Argument. In *Proceedings of the Fifth International Conference on AI and Law*, ACM Press, New York, pp. 165–74.

Prakken, H. & Sartor, G. 1997. Reasoning with Precedents in a Dialogue Game. In *Proceedings of the Sixth International Conference on AI and Law,* Melbourne, ACM Press: New York, pp. 1–9.

Toulmin, S. 1958. *The Uses of Argument.* Cambridge University Press,

Walton, D. N. 1989. *Informal Logic: A Handbook for Critical Argumentation.* Cambridge University Press, Cambridge.

Walton, D. N. & Krabbe, E. C. W. 1995. *Commitment in Dialogue.* State University of New York Press.

Wittgenstein, L. 1953. *Philosophical Investigations.* Blackwell; Oxford.