Available online at www.sciencedirect.com
SCIENCE DIRECT®
Knowledge-Based Systems 19 (2006) 22–31
Knowledge-Based SYSTEMS
www.elsevier.com/locate/knosys

Representing dialogic argumentation

Chris Reed *

Division of Applied Computing, University of Dundee, Dundee DD1 4HN, Scotland, UK

Received 6 October 2003; accepted 18 August 2005
Available online 23 September 2005

Abstract

Dialogic argumentation is a crucial component in many computational domains, and forms a core component of argumentation theory. This paper compares two approaches to dialogue that have grown from two different disciplines; the descriptive–normative approach of applied philosophy, and the formal, implemented approach of computer science. The commonalities between the approaches are explored in developing a means for representing dialogic argumentation in a common format. This common format uses an XML-based language that views locutions as state-changing operations, drawing on an analogy with classical artificial intelligence planning. This representation is then shown to hold a number of important advantages in areas of artificial intelligence and philosophy.

Keywords: Argumentation; Dialogue; XML

1. Introduction

Argumentation has been used in knowledge representation for some time, as a means of handling the uncertain, incomplete and possibly inconsistent information that characterises many real-world domains. In domains as diverse as oncology and law, formal models of argumentation have been employed to represent information in a format that is both computationally flexible and tractable, and at the same time, intuitive and easy to understand (see, e.g. [1] for clinical examples [2] for legal examples, and [3] for a wide-ranging review). In many domains of application, it has been found that argumentation-based models can support a wide range of application types, including decision support, advice-giving, and knowledge elicitation. The reason for this is that there is a strong connection between the representation format, and the subsequent interaction with that format by a human. As this area has matured, however, a problem has begun to emerge. The various works in different domains have idiosyncratic ways of handling argument and argument-mediated human–computer interaction. In order to support expansion and growth, more generic models need to be developed.

In the theory of argument representation, this has begun to happen, with a fair degree of consensus and incremental development in areas such as logic programming (see, e.g. [4] for a particularly influential approach). Recent work has also started to take a more generic approach in modelling various types of real-world argument [5]. There remains a crucial piece of the puzzle to be fitted in: a generic means of representing dialogic argument.[1] A standard approach is needed for handling the ways in which knowledge is structured using argumentation and then exploited at the human–computer interface. It is this problem which is tackled here.

2. Background

There is an extensive literature on the specification of formal dialogue 'logics', which attempt to capture aspects of structured human–human communication, with the aim either of understanding, prescriptively improving, or artificially recreating such communication. These include work on specific types of discourse such as information seeking [6] critical discussions [7] and question–answer dialogues [8], and also across such types [9–11]. It also embraces work in applied domains such as legal theory [2,12] and areas of artificial intelligence [13].

Girle [14] has pointed out that despite the wide range of approaches, aims and domains, it is conspicuous that these various dialogue systems share two key common properties.

* Tel.: +44 1382 344145; fax: +44 1382 345509.
E-mail address: chris@computing.dundee.ac.uk

[1] The term *dialogic argument* is used here to refer to the sort of argument in which two or more parties engage, to be contrasted with monologic argument, which is prepared or presented by a single party in a 'one-shot deal'.

0950-7051/$ - see front matter
doi:10.1016/j.knosys.2005.08.002

Each system either explicitly represents, or can easily be viewed as (i) handling participant commitments; and (ii) having four types of state-changing rules. Let us take these two in turn.

Argumentation theory has, for the most part, revoked a mentalist approach to interlocutor states (that is, representing interlocutors' beliefs, goals, intentions, etc.), and in its place, adopted a more behaviourist or in some cases, functionalist view. In this approach, any objective notion of truth is eschewed, and instead subjective notions of acceptability and relevance form a foundation on which participants build commitments. In many dialogue systems, uttering a proposition commits the utterer to the propositional content ([7]: Chapter 2). There are, of course, many other ways of incurring propositional commitment, and in some dialogue systems, even this simple act may have rather more complicated consequences. But in essence, a commitment is something that a participant can be held to (and, often, takes on a burden of proof for). It is even clear from such a summary description that commitment does not entail belief (nor any other particular mental state)—a person may lie, for example. However, in contrast, belief (when used to characterise or motivate utterance) does entail commitment. Thus if, say, a speaker's belief in proposition P is used as a precondition or trigger for her uttering P, then this inevitably entails commitment to P. This is useful because it means that those (relatively few and typically computational) models that take a mentalistic approach can nevertheless be easily translated to a commitment model.

These various models also have four classes of state-changing rules: locution rules, commitment rules, structural rules and termination rules. Locution rules describe the locutions that are possible; structural rules describe how locutions may be combined into exchanges; commitment rules describe how locutions update the commitments of the participants; and termination rules describe the conditions under which the dialogue terminates. Locution rules are usually given at quite an abstract level in most of the examples cited above—underspecifying, for example, who it is that can utter each locution. Commitment rules are rather more specific, often detailing different types of commitment store, and exactly how locutions update them. Structural rules are interesting because they are typically quite brief, focusing on simple pairings (specifying that after an utterance of type X, should follow an utterance of type Y). This is in contrast to more elaborate approaches, common in AI, of specifying extensive protocols, often as finite state machines. Finally, termination rules are usually quite simple specifications (such as a proponent retracting their original thesis).

By way of brief example of the way in which formal systems are used to explain the dynamics of dialogic argumentation, consider the following:

(1) Bob: Let's buy the blue sofa.
(2) Wilma: Why the blue one?
(3) Bob: It'll go with the curtains.

An intuitive, simplistic formal system might be used to explain that at (1), Bob makes an assertion (a type of locution), that leads to the content of the utterance (*We should buy the blue sofa*) being added to his commitment store. At (2) Wilma employs a challenge locution to attack one of Bob's commitments (namely that *We should buy the blue sofa*). Assuming that a challenge requires a player either to concede or to defend the challenged statement, Bob here defends with another claim which is then also added to his commitment store. A formal system that accounts for this sort of behaviour would therefore specify rules expressing that assertions and challenges are permissible, that a challenge may follow an assertion, that assertions add their contents to the speaker's commitment store, and that a dialogue can terminate when challenges have been defended against.

The thesis of this paper is that the classes of rules common to all dialogue systems can be exploited in developing a common representation format for dialogic argumentation that is suitable for both the more and less formal applications of argumentation within artificial intelligence, knowledge representation and computer science, and for the more and less formal descriptive and normative work in argumentation theory, critical thinking and informal logic. The first step is to consider the theoretical and practical models which could form the basis for such representation.

3. Potential models

There are two models which are particularly promising candidates as starting points for representing dialogic argumentation: the [7] commitment based approach used in developing the formal characterisations PPD and RPD, and the predominant agent communication language specification approach typified by the FIPA ACL [15]. (In both cases, there is a range of similar, complementary approaches: the two selected are indicative of the approaches and offer particularly clear examples.) Both models take a sufficiently formal approach to be amenable to computational interpretation, whilst remaining sufficiently rich to handle a wide variety of naturalistic argumentation forms.

Walton and Krabbe [7] develop several systems of dialogues, working in the Hamblinian tradition [10]. They begin by sketching out a class of systems, called permissive persuasion dialogues (PPD) which have a number of common features, including (i) that there are exactly two players; (ii) that there are alternating moves; (iii) that there is a burden of proof such that challenged assertions must be defended; and many more ([10]: 133–140). They then move to laying out a specific system of this type, called PPD_0, which is explained in sufficient detail to form the basis of reasonably direct implementation [16]. The rules of PPD_0 build upon, specialise, and in some cases select from alternatives from within the specification of the PPD class. Thus, for example, PPD_0 specifies (i) that the move ΔsoP, which corresponds to offering premises, Δ, in support of a claim P, enters all the Δ into the speaker's commitment store; and (ii) that for every premise in the Δ of a previous move ΔsoP, a speaker must either concede or challenge that premise.

In contrast to the flexibility of PPD_0, Walton and Krabbe also introduce a more restrictive type of dialogue called Rigorous Persuasion Dialogue. Though it shares the same basic structure as PPD, RPD has some stark differences, including asymmetry between the interlocutors (one is a questioner, the other a respondent), and typically, that termination is reached quite quickly, after just a few exchanges. Again RPD represents a class, within which RPD_0 is detailed as an example. RPD_0 is characterised by a single initial thesis, which is put forward by a proponent and then questioned by an opponent. The rules of RPD_0 specify that the proponent must respond to each questioner put forward by the opponent.

Finally, Walton and Krabbe demonstrate how dialogues can be embedded within one another, taking as a canonical example the embedding of RPD_0 in PPD_0. The idea is that in a usually quite free discussion (characterised by a PPD), there are periods of 'tightening up', in which a more narrow dialogue game is pursued (characterised by an RPD). The rules of how RPD_0 can be embedded in a PPD are then detailed in another specific PPD, PPD_1.

The problem with Walton and Krabbe's work is that it is underspecified. Implementation (such as has been suggested by Dignum et al., and explored in unpublished pilot research at the University of Dundee) requires many refinements. Many of these refinements are of vital importance, and worthy of study in their own right, such as, for example, whether or not a given dialogue move should involve multiple locutions: [12] has investigated this problem as a separate issue divorced from Walton and Krabbe style dialogue specification). For direct implementation, exploitation in practical domains, and then comparison between types, it would be far more convenient if the dialogue were much more tightly specified, making implementation an engineering issue, rather than a set of open research questions.

There is a strong tradition of dialogue design in multi-agent systems, given that much of the power and flexibility of such systems lie in their distributed nature, and the robustness and malleability that arise from carefully specified inter-agent communication. Given its roots in theoretical computer science, much of this research has employed formal methods to fully specify agent dialogue structures, in contrast to the more abstract approach taken by Walton and Krabbe. From early work on the Contract Net Protocol [17], through more generic approaches using Finite State Machine specifications [15], to recent specification using UML extended for agent communication [18], the emphasis has been on a simple means of characterising extended exchanges between agents. This proliferation in languages has led to standardisation efforts such as FIPA [15], and though such efforts have been criticised for their inadequate coverage in specific domains [19], they have nevertheless been influential in the design and deployment of implemented systems.

FIPA characterises dialogue in two ways. First, it specifies locutions (inform, request, etc.) on the basis of the mental state of the interlocutors that must hold before the locution can be uttered, and on the update to that state after the locution is uttered. So for example, an inform can be uttered just in the case that the speaker believes the content, and as a result the hearer then believes (or at least, should believe) the content. The specifications in fact capture much more detail than this, and do so using a multi-modal logic, but in essence the characterisation is straightforward. To complement the specification of individual locutions, FIPA also constrains the ways in which extended exchanges may be composed by modelling dialogues as finite state machines. Thus, for example, a request from A to B is (ceteris paribus) to be followed by an inform from B to A.

For the task at hand, however, the multi-agent systems (MAS) approach falls short in several respects. First and foremost, the focus in MAS has, for obvious reasons, been placed squarely upon the design of a clear and unambiguous means of communication, rather than upon modelling real world communication. This means that MAS models are simply not suited to being extended to handle human–human or even human-computer interaction. Secondly, there are also computational limitations of implemented systems. One of the most crippling of these limitations is that the finite state machine approach which is pervasive throughout models of agent communication is inflexible and limited. Singh explains that "representations based on monolithic finite-state machines are suitable for only the most trivial scenarios. They cannot accommodate distributed execution, compliance testing or exceptions…" ([20]: 38).

The two approaches—the philosophical and the computational—are not entirely divorced, since some multi-agent systems research has attempted to exploit a commitment-based approach [20] building on Habermasian notions of social commitment. There is also preliminary work developing multi-agent models of Walton and Krabbe style dialogues, with an implementation of RPD offered in [16], and an analysis of the advantages conveyed by dialogue embedding in inter-agent communication in [21]. Some aspects of Walton and Krabbe style commitment have also been implemented in inter-agent communication [13]. These attempts at cross-over, however, have to date been rather limited, and tied to very specific results in the philosophical literature. There has been no attempt to bring the two approaches together in a more systematic way that allows both formal computational dialogue systems and more abstract philosophical dialogue systems to be described, specified and analysed, and then subsequently compared and further developed, in a similar manner.

4. Building a representation

How, then, can the computationally attractive approach proposed in multi-agent system languages be reconciled with the richer commitment based models of Hamblin, Walton and Krabbe, et al., in such a way that a wide range of both natural and artificial dialogues can be intuitively and straightforwardly represented?

The solution lies in developing the commonalities of dialogue systems described by Girle into a representation language. Here, that representation is achieved through XML because the Document Type Definition (DTD) specification

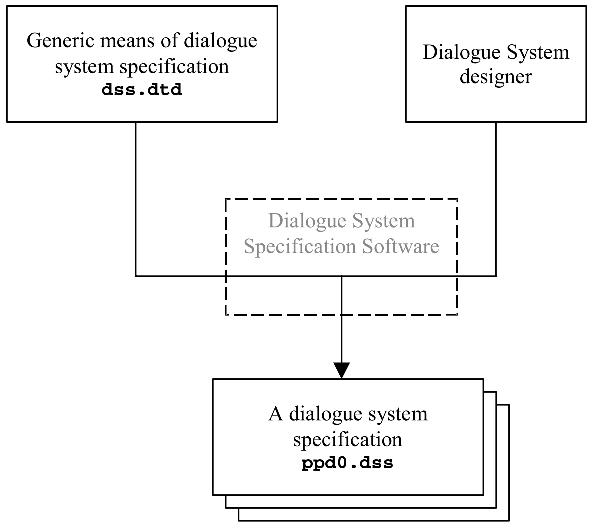

Fig. 1. Dialogue system specification.

for an XML language clearly reflects and embodies those commonalities between dialogue systems. Furthermore, a given dialogue can then be marked up according to the specification of a particular dialogue system. The picture is thus as in Fig. 1.

A DTD (called `dss.dtd`) lays out how a given dialogue system (PPD_0, DC, protocols of the FIPA ACL, or whatever) is to be specified. That DTD describes how the various components that make up all dialogue systems (locutions, commitments, rules, etc.) may be combined to characterise a particular dialogue system. A particular dialogue system (say PPD_0) can then be represented by constructing an XML file that conforms to `dss.dtd`. That representation is a dialogue system specification (or DSS, stored in a file that might be called `ppd0.dss`). The construction of such a specification can either be manual (as it is currently), by directly writing the DSS file, or, in the longer term, would be semi-automated through the use of software that supports an intuitive interface to help the user create a dialogue system quickly, either working from scratch, or from a more or less formal specification available in the literature.

Once a dialogue system specification is available, it then becomes possible (i) to mediate (that is, to provide support for) a human–human dialogue according to the rules of that dialogue system; (ii) to conduct a dialogue, with the machine playing the role of an interlocutor in that dialogue system; (iii) to mark up an existing dialogue (transcription) that was conducted according to the rules of that dialogue system; (iv) to explore and interact with a stored dialogue that was conducted according to the rules of that dialogue system. In every case, the dialogue conducted is governed by a dialogue system specification (such as `ppd0.dss`), and the dialogue itself is recorded as a separate file. Notice that a dialogue system specification characterises how a dialogue is to proceed—it does not govern how a dialogue should be represented. For this, a separate specification is required, one that is sufficiently generic to cover all the argument types. It turns out that this is relatively easy to specify, particularly as such a specification can build upon earlier work in specifying (monologic) arguments in XML, using the argument markup language AML [5]. Thus a dialectic markup language,[2] DML, defined in a DTD (`dialectic.dtd`) governs how any particular dialectical dialogue is to be represented. This governance is independent of the rules that are employed to control the unfolding of such a dialogue. So, with a dialogue system specification (such as `ppd0.dss` produced in Fig. 1), various types of software can build and interact with DML files. Furthermore, as DML builds on the monologic AML, existing software (viz. Araucaria [5]) can be employed to explore and manipulate the contents of each (monologic) turn in a dialogue.

Figs. 1 and 2 thus provide the overall picture of how the pieces of the solution fit together. Much remains to be done, including many of the ‘software’ boxes in those diagrams (some of these are described in [5]), but the focus of the remainder of this paper is upon how the dialogue system specification can be carried out.

Let us recap the two commonalities that Girle pointed out: (i) handling participant commitments; and (ii) having locution rules, structural rules, commitment rules, and termination rules. Our XML representation must have each of these as first class data objects that moreover are simple to specify and use. Our approach to developing such a representation exploits an analogy with another area of artificial intelligence.

The view of dialogue as a series of moves that change the state (of the dialogue and of the participants’ commitment stores) is reminiscent of the state-to-state changes captured in conventional AI planning of a STRIPS style [22]. That is, individual locutions are akin to operators that are applicable when the dialogue is in a particular state, and that make specific, localised changes to that state. The commitments, dialogic obligations,[3] argumentative structures, and recent utterances together define a state, upon which locutions work. Of course, the notion of locutions corresponding to planning operators is not a new one: in conjunction with Speech Act Theory [23], it has motivated a great deal of intention-based planning in natural language generation [24], and simple automated dialogue (e.g. in the TRINDI system [25]). Here, we can exploit the analogy not for planning purposes, however, but for representational consistency.

A locution is thus comprised of a set of preconditions and a set of postconditions, and both those sets are constructed as a partial state description composed of commitments, dialogic obligations, argumentative structures and recent utterances (that is to say, of information contained in the locution rules, structural rules and commitment rules), and then finally,

[2] Dialogue markup language might be a more intuitive name, but there is a strong thread of research examining the markup of real dialogues, in all their natural glory, encompassing intonation, interruptions, pauses, throat-clearings and so on. The current work does not include such features, as explained in Section 4, and to make this clear, the language defined is for dialectic markup. Of course, dialectic is not exactly coextensive with dialogic argumentation, but it is close enough to stand here.

[3] That is, the obligations for a participant as a result of particular locutions; so for example, in some systems, challenging an opponent’s point of view obliges that opponent to provide a support—this is burden of proof. (cf. [12,39]).

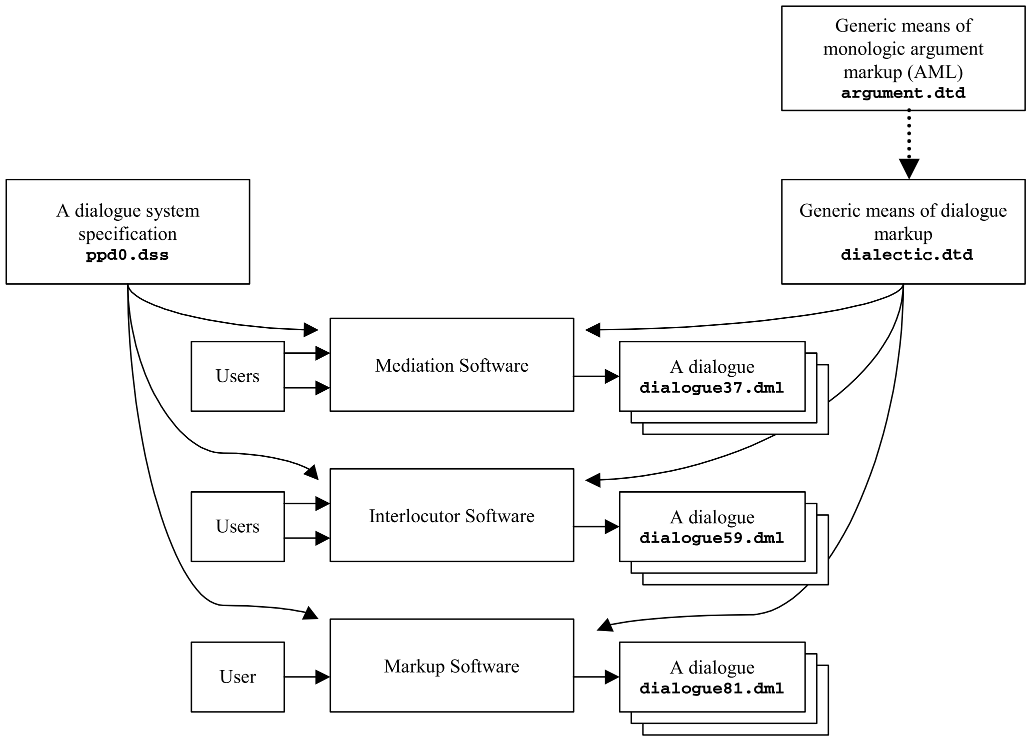

Fig. 2. Creation, manipulation and interaction with dialogue.

the termination rules are translated into a further set of ‘goal’ states.

A dialogue system can thus be captured completely by specifying the preconditions and postconditions of every possible locution, along with a characterisation of participants’ commitment stores and a list of the termination states. Pre- and postconditions can be completely specified by listing those dialogic obligations, commitment store entries and structural conditions that their locutions depend upon or establish. Dialogic obligations are simply future locutions; commitment store entries are references to commitments in particular stores; structural conditions are simply argumentation structures defined in AML. This (paragraph) is essentially the dialogue system specification DTD.

To demonstrate how the specification of an existing dialogue system is achieved, the ΔsoP (giving an argument for P) locution, used to respond to a challenge (marked by P??) in Walton and Krabbe’s [7] PPD_0 is rendered in this style, first informally, and then in XML. The relevant extracts from Walton and Krabbe’s description of PPD_0 (ibid: 150–151) are as follows:

“ΔsoP may be used only if P is not among the concessions of the listener and some earlier move contained P??” [Structural Rule #9]

“ΔsoP enters all the elements (explicit premises and warrant) of the argument *ΔsoP* into the set of assertions and the set of the concessions of the speaker” [Commitment Rule #6]

Informally, these could be represented as a locution called DsoP, with premise (structure) D uttered to support conclusion P by a particular participant, B:

Locution: DsoP (B, D, P)
Preconditions:
W has uttered P??
P is not a concession of W, i.e. $P \notin CW$
Postconditions
D is added to the assertions and concessions of B, i.e. $AB = AB \cup D$ and $CB = CB \cup D$

This can be translated into the XML offered by `dss.dtd` thus:

```
<LOCUTION locution_name=“DsoP” role=“B”>
 <PARAMETER id=“D”/>
 <PARAMETER id=“P”/>
<PRECONDITIONS>
 <COMMITMENT_STATE negated=“yes”>
  <COMMITMENT id=“P” role=“W” store=“A”/>
   </COMMITMENT_STATE>

   <LOCUTION_STATE>
    <U_LOCUTION  locution_name=“Challenge”
    role=“W”>
     <INSTANTIATE formal=“P” actual=“P”/>
    </U_LOCUTION>
   </LOCUTION_STATE>

  </PRECONDITIONS>


  <POSTCONDITIONS>

   <COMMITMENT_STATE>
```

```
24      <COMMITMENT id="D" role="B" store="A"/>
25     </COMMITMENT_STATE>
26
27    </POSTCONDITIONS>
28
29   </LOCUTION>
```

Line 1 defines the locution DsoP for a player B, lines 2 and 3 specify its parameters D and P. Lines 6–18 capture the preconditions of the locution: that W does not have P in his commitment store A (line 9) and that W has uttered a Challenge(P) locution (i.e. P??) (lines 13 and 14). Lines 21–27 capture the effects of the locution, namely, that D gets added to B's commitment store A (line 24).

With this clear specification, it is then easy to see how locutions could start to be chained together. Consider this extract (ibid.: 151)

"If the preceding move contained P??, the speaker must utter an argument ΔsoP, or utter nc(P) or na(P)." [Structural Rule #3(e)]

The formal specification of this includes, amongst other things, that dialogic obligation imposed by uttering P?? can be met by responding with a ΔsoP, as defined above. Thus where the locution ΔsoP has a U_LOCUTION precondition (referring to a locution that has been uttered), so the P?? locution has an O_LOCUTION postcondition (referring to a locution that is afterwards obligatory).

In this way, by linking turns in a dialogue with the structural, commitment and dialogic obligation aspects, the whole of PPD_0 can be captured using the syntax provided by `dss.dtd`. Of course, all the dialogue system can do is to prescribe what can happen: at any one stage in a real dialogue there are likely to be numerous possible moves available, and an agent—be it human or artificial—will have to decide between the legal moves.

Walton and Krabbe style dialogues are broad, normative systems that are designed to be applied to natural discourse. One of the advantages of the approach described here is that a single representation format can also handle applied, implemented, task-oriented, domain-specific dialogue specifications. A good example is the Contract Net (CNET) protocol [17], an early example of coordination for distributed reasoning. CNET offers a particularly applied example of dialogue design, with implementations in, inter alia [26] and [27]. Under the CNET protocol, a manager broadcasts a *task announcement* (referred to as a 'call for proposals' in later work), calling for bids for the execution of a particular task, with particular requirements and restrictions (including an expiration time for the bidding process). Individual task agents respond either by declining, or else by submitting a bid. The *bid* message may include a straightforward proposal for carrying out the task, or may request further information. The manager then selects one (or more) task agents to carry out the work, and responds with a corresponding *award* message to those agents. The act of bidding can be seen as a locution (and is often characterised that way in more recent re-engineered versions of the protocol such as that found in FIPA). An example of the bid locution is offered in [17]:

To: 25
From: 42
Type: BID
Contract: 22-3-1
Node Abstraction:
POSITION LAT 62N LONG 9W
SENSOR NAME S1 TYPE S

The general structure of the locution can be captured simply in a fragment of DSS thus:

```
1   <LOCUTION locution_name="bid" role="C">
2    <PARAMETER id="NA"/>
3
4    <PRECONDITIONS>
5
6     <LOCUTION_STATE>
7       <U_LOCUTION locution_name="TaskAnnouncement" role="M">
8        <INSTANTIATE formal="BS" actual="NA"/>
9       </U_LOCUTION>
10      </LOCUTION_STATE>
11
12    </PRECONDITIONS>
13
14
15    <POSTCONDITIONS>
16
17     <COMMITMENT_STATE>
18      <COMMITMENT id="NA" role="C" store="T"/>
19     </COMMITMENT_STATE>
20
21    </POSTCONDITIONS>
22
23   </LOCUTION>
```

Line 1 gives the name of the locution, and allows contractor nodes only to use it. Line 2 describes the parameter, the node abstraction. Lines 6–10 capture the previous task announcement utterance of a manager agent in which the bid specification matches the node abstraction of the current bid. Finally, lines 17–19 capture the commitment of the contractor node to perform the specified action in the event of the manager accepting the bid. (Bids in CNET are 'honest'.)

To capture the ability for a bid to include a request for information, an additional definition of the bid locution is required, just as it is for alternative pre- or postcondition states. (This second definition would not include commitment to carrying out the task as a postcondition, for example.) Some features of CNET dialogues do not seem to be features of dialogue in general, but occur instead as content material of a dialogue. A good example is the expiration time. In CNET, expiration time is a key component of the structure of a dialogue: a node simply will not submit a bid after the expiration time. This is partly due to differences between CNET and more recent work in multi-agent systems—the latter typically enforce agent autonomy as a cornerstone of system design [28]. But it is also partly a result of hard-wiring in CNET that is inappropriate for dialogue in general. For though the concept of an expiration time for replies is a common concept, it is expressed in the content language of the locutions, rather than in the structure of the locutions themselves.

To demonstrate the general applicability of the approach, it is also useful to focus on a system designed to explore certain philosophical principles—in this case, the principle of non-cumulativeness, in [8] system DC. The aim of Mackenzie (amongst many others subsequently) is to show how the fallacy of begging the question (the petitio principii, or circular reasoning) can be explained from a dialogical, rather than epistemic, point of view. The explanation builds on the concept of cumulativeness in dialogue, whereby once interlocutors take on commitment towards propositions, they cannot subsequently relinquish it. (Cumulativeness is thus analogous to monotonicity.) Mackenzie's DC characterises noncumulativeness for statements; the subsequent modification engineered in DD adds noncumulativeness for challenges to statements. One of the key foundations for noncumulativeness in both DC and DD is the commitment rule for statements, CR_S. For DC, this is described by Mackenzie thus:

"After a statement 'P', unless the preceding event was a challenge, 'P' is included in both participant's commitments." ([8]: 119)

In his appendix, Mackenzie captures this intuition more formally with:

"CR_S: After $\langle n, A, s\rangle$, where the event at $n-1$ is not $\langle n-1, B, \mathbf{Y}'t\rangle$, $C_{n+1}(A)=C_n(A)\cup\{s\}$; $C_{n+1}(B)=C_n(B)\cup\{s\}$"

where s is the utterance of a statement, $\mathbf{Y}'t$ indicates a challenge to statement t, A and B are interlocutors, and is $C_n(A)$ is the set of A's commitments at turn n.

Representing this in the DSS approach described above requires noticing the implicit ellipsis in Mackenzie's definition: a participant can either make an unfettered statement at some point in a dialogue, and have both commitment stores updated; alternatively a participant can make a more fettered statement, as a response to a challenge, and in this case, a different update to commitment stores takes place—this is captured in Mackenzie's commitment rule, CR_{Y_S}. This distinction can reasonably be interpreted in terms of two types of statement: fettered and unfettered, where CR_{Y_S} focuses on the former and CR_S on the latter. Both versions of the statement can be captured quite straightforwardly; the unfettered, for example, runs thus:

```
1   <LOCUTION locution_name="Unfetter-
    edStatement" role="B">
2   <PARAMETER id="S"/>
3
4
5   <PRECONDITIONS>
6
7   <COMMITMENT_STATE>
8   </COMMITMENT_STATE>
9
10   <LOCUTION_STATE negated="yes">
11     <U_LOCUTION locution_name="Why"
     role="W">
12     </U_LOCUTION>
13   </LOCUTION_STATE>
14
15  </PRECONDITIONS>
16
17
18  <POSTCONDITIONS>
19
20   <COMMITMENT_STATE>
21     <COMMITMENT id="S" role="B"/>
22     <COMMITMENT id="S" role="W"/>
23   </COMMITMENT_STATE>
24
25  </POSTCONDITIONS>
26
27 </LOCUTION>
```

Thus lines 10–13 capture the unfettered nature (i.e. that the previous turn was not a Why-challenge of any proposition at all—hence the absence of parameter instantiation between lines 11 and 12). Lines 20–23 then represent the update to both participants' commitment stores.

5. Discussion

With an ability to handle examples from philosophy at both the more theoretical and more applied ends of the scale, and then a further example from a particularly applied computational approach, the representational techniques proposed here have clear advantages in terms of representational adequacy and generality. This makes them a good candidate for facilitating interchange and re-use between these domains, and in particular, between human and computer dialogues.

The approach represents a starting point. Several important problems remain to be tackled, however. In the first place, the DTD provides an inelegant solution to handling disjunction amongst preconditions and postconditions. To define a locution which can be uttered in condition x or condition y, the entire locution must have two definitions differing only in x and y. Similarly, to represent the disjunction that either player a or player b may utter a given locution, again, the whole definition must be repeated. A more succinct means of characterising disjunction would reduce redundancy and increase clarity.

Secondly, the current DTD provides only a very crude means of coping with complex discourse obligations: it is not currently possible, for example to specify a locution that should happen some n moves after the current locution, or to specify future combinations of locutions that would be (im)possible as a result of the current locution.

Such specific technical problems will continue to present themselves as the approach is applied in new domains, and will lead to refinement and enriching of the theory, but the contention here is that the broad approach is sound, as demonstrated by the ease with which it can be applied to domains that have traditionally been distant and difficult to bring together.

Though the use of XML is often supported through a canonical list of advantages (including openness, ease of translation, existence of software and algorithms for processing), it is being used here not only for markup but also for straightforward knowledge representation. Though this is not uncommon (witness, for example, the DAML effort[4]), it demands further justification. In the first place, it forms a natural representation that is close enough to the idiosyncratic descriptions of the different systems to be human understandable even in its raw form. It is not completely alien to any of the communities that have originated dialogue system definitions. Secondly, it forms a good representation for automatically generating and implementing a dialogue system as an agent communication language. And finally, given that the approach as a whole integrates dialogue execution, dialogue markup and dialogue definition, having a common format for all three simplifies the process of design and implementation, and XML is a natural choice for the markup component in particular.

The use of XML for dialogue description is also instantly reminiscent of work on dialogue markup standards used in analysing real human-human dialogue fragments such as those in the HCRC MapTask corpus ([29,30]). The extensive corpus markup used in such work is based on standards like the Corpus Encoding Standard[5] which in turn is founded upon the widescale Text Encoding Initiative.[6] A striking feature of such research is that it typically focuses on lower level features of discourse. Unusually, the Maptask corpus does include reference to dialogue games played by participants, but [29] describe how these games are developed a priori, as a pretheoretical, and certainly preempirical, step. Though integrating the DSS/DML scheme described here with the Corpus Encoding Standard represents a task for future work, it is clear that at the very least, the two are complementary and do not currently cover the same ground.

Viewing moves in dialogue games as state transitions is also similar to the approach taken by the TRINDI and SIRIDUS projects, in which dialogue acts are taken to be updates to information states ([25,31]). TRINDI is very flexible with respect to its modelling of information states—in some applications for example, they are based on DRT [32]. The kind of state here encoded in DSS specifications would be broadly compatible with TRINDI-style modelling. In recent SIRIDUS work [31], negotiation in information states is addressed as a species of argumentation. Again, however, the design of the dialogue game with its rules and relations, is seen as the task of the researcher, and as something that is done offline, before system building and evaluation. So again, the flexibility offered by the commitment-based DSS/DML approach, whereby the focus is squarely upon the rapid design and deployment of systems of dialogue, would complement research such as that encompassed by SIRIDUS where the focus is instead upon the process of conducting a dialogue according to a given set of rules.

Finally, given that most philosophical theories of argument are more or less normative, there also remains the interesting challenge of being able to mark up deviations from the dialogic specification that occur in natural dialogues. Such analysis tools must be able to withstand the vagaries of real-world language use, and yet have at their core the dialogic norms captured by DSS/DML specifications.

6. Conclusions

The current (end 2003) status of this work is as follows. The (DTD for) monologic argument markup language, AML, has been stable for over a year, and is in use in several argumentation applications. These applications are available online[7] and are being used for research and teaching in a number of universities and schools [5]. This AML is also being used in the ongoing construction of a large corpus of argumentation, also available online to the academic community [33]. The (DTD for) dialogue markup language, which extends AML is at a first draft revision, which is being used to design and implement dialogue-generating software. The (DTD for) dialogue system specification is similarly at a first draft revision, and has been used to produce a description of a simplified version of PPD_0. The Araucaria software used for the AML-markup of monologic argument, its manipulation, diagramming, and corpus access, is complete, and has undergone several revisions as a result of user feedback. Online access to search tools for the corpus is also now available. Prototype implementations of PPD_0 and a simplified version of Mackenzie's BQD have been implemented manually to sketch an approach to the development of (a) computer mediated argument according to the rules of a dialogue system, and (b) computer controlled knowledge acquisition guided by the rules of a dialogue system. Full implementations of these software components, plus that of a tool for the markup of dialogue are all at the planning stage. Finally, a prototype implementation of a game-playing system that supports two human users playing any dialogue game specified using the DSS structure has been completed.

The ability to specify, explore and evaluate dialogue systems is of practical use in both theoretical and applied areas of philosophical research, as has been demonstrated by existing collaborations between those areas and Artificial Intelligence, such as ([34,35]). Similarly, prototyping and evaluating argumentation based agent communication

[4] See http://www.daml.org/

[5] See http://www.cs.vassar.edu/CES/

[6] See http://www.tei-c.org/

[7] At http://araucaria.computing.dundee.ac.uk

languages becomes a realistic target when design and implementation can be completed rapidly through a drag-and-drop interface. But in addition to the two areas upon which the work has drawn, there is also a range of other areas of potential application. The first is in conversation analysis, particularly with a view to identification of high-level structures. The work carried out as part of research into Dialogue Macrogame Theory, DMT [36] provides a good example. Mann's approach, like many in the area, is primarily empirical: collecting and analysing structures of conversations. Such corpus work can benefit from XML mark-up for analysis and reuse [30, 33] and the current work could provide a powerful representation language that would support new lines of inquiry. For example, it would be possible to match examples of actual text to proposed philosophical dialogue systems, and to allow experimental subjects to interact with one half of an actual dialogue to explore the directions that the dialogue could have taken—building, in effect, a natural profile of dialogue [37]. A second area of application is in discourse design for education, especially for online computer assisted learning systems. Jackson [38] identifies the important roles that dialectic can play in online learning. The dialogue system specification approach described here would allow rapid development of wide variety of dialogues through which students could engage with the online material. Empirical work could then assess the efficacy of different dialogue systems in different domains (Is Socratic dialogue good for mathematics teaching? Do critical discussions work well in teaching social psychology? etc.). Another example of application is in the public understanding of science, in which the presentation of complex issues in non-technical, easy-to-grasp ways is a real challenge—and one that presents a barrier to the fulfilment of governmental aims for public involvement in policy making. Presenting scientific information as arguments has been demonstrated to improve users' understanding (even in highly emotional situations such as genetic counselling [1]). By then providing access to those arguments through naturalistic dialogue offers a means to allow users to take an active part in exploration of an issue, and to interact at a pace and level of detail that suits the individual.

Though there are many further application areas, it is clear that the techniques offered by the approach are quite generic, and hold significant promise both as a technique for developing models of language use and reasoning in the humanities and social sciences, and also as a means of improving computer–computer and human–computer interaction.

References

[1] A. Coulson, D. Glasspool, J. Fox, J. Emery, RAGs: A novel approach to computerised genetic risk assessment and decision support from pedigrees, Methods of Information in Medicine 40 (2001) 315–322.
[2] T.F. Gordon, The Pleadings Game, Kluwer, Amsterdam, 1995.
[3] D.V. Carbogim, D.S. Robertson, J.R. Lee, Argument-based applications to knowledge engineering, Knowledge Engineering Review 15 (2) (2000) 119–149.
[4] P.M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and N-person games, Artificial Intelligence 77 (1995) 321–357.
[5] C. Reed, G.W.A. Rowe, Araucaria: software for solving puzzles in argument diagramming Applied Computing Technical Report, University of Dundee, Dundee, 2001.
[6] J. Hintikka, The logic of information-seeking dialogues: a model in: W. Essler, W. Becker (Eds.), Konzepte Der Dialektik, Vittorio Klostermann, Frankfurt, 1981 pp. 212–231.
[7] D.N. Walton, E.C.W. Krabbe, Commitment in Dialogue, SUNY Press, New York, 1995.
[8] J.D. Mackenzie, Question begging in non-cumulative systems, Journal of Philosophical Logic 8 (1979) 117–133.
[9] L. Carlson, Dialogue Games: An Approach to Discourse Analysis, Reidel, 1983.
[10] C.L. Hamblin, Mathematical models of dialogue, Theoria 37 (1971) 130–155.
[11] J.D. Mackenzie, Four dialogue systems, Studia Logica 49 (1990) 567–583.
[12] H. Prakken, Relating protocols for dynamic dispute with logics for defeasible argumentation, Synthese 127 (2001) 187–219.
[13] S. Parsons, P. McBurney, Argumentation-based dialogues for agent coordination, Group Decision and Negotiation 12 (4) (2003).
[14] R.A. Girle, Commands in dialogue logic in: D. Gabbay, H.-J. Ohlbach (Eds.), Formal and Applied Practical Reasoning, Springer, Heidelberg, 1996 pp. 246-260.
[15] FIPA, 'FIPA Communicative Act Library Specification' at www.fipa.org (2002).
[16] F. Dignum, B. Dunin-Keplicz, R. Verbrugge, Creating collective intention through dialogue, Logic Journal of the IGPL 9 (2) (2001) 289–303.
[17] R.G. Smith, The contract net protocol: high-level communication and control in a distributed problem solver, IEEE Transactions on Computers C29 (12) (1980) 1104–1113.
[18] B. Bauer, J. Muller, J. Odell, Agent UML: A formalism for specifying multiagent interaction in: P. Ciancarini, M. Wooldridge (Eds.), Agent Oriented Software Engineering, Springer, Heidelberg, 2000 pp. 91–103.
[19] C. Reed, T.J. Norman, N.R. Jennings, Negotiating the semantics of agent communication languages, Computational Intelligence 18 (2) (2002) 229–252.
[20] M. Singh, A social semantics for agent communication languages in: F. Dignum, M. Greaves (Eds.), Issues in Agent Communication, Springer, Heidelberg, 2000.
[21] C. Reed, Dialogue frames in agent communication Proceedings of ICMAS'98, IEEE Press, 1998. pp. 246–253..
[22] R.E. Fikes, N.J. Nilsson, STRIPS: a new approach to the application of theorem proving to problem solving, Artificial Intelligence 2 (3/4) (1971) 189–208.
[23] J. Searle, Speech Acts, Cambridge University Press, Cambridge, 1969.
[24] E.H. Hovy, Pragmatics and natural language generation, Artificial Intelligence 43 (2) (1990) 153–197.
[25] S. Larsson, D. Traum, Information state and dialogue management in the TRINDI dialogue move engine toolkit, Natural Language Engineering 6 (3/4) (2000) 323–340.
[26] T. Sandholm, An implementation of the contract net protocol based on marginal cost calculations Proceedings of the 11th National Conference on AI, AAAI'93 1993 pp. 256–262.
[27] C. Dellarocas, M. Klein, Designing robust, open electronic marketplaces of contract net agents Proceedings of the 20th International Conference on Information Systems 1999.
[28] M.J. Wooldridge, Introduction to Mutli-Agent Systems, Wiley, Chichester, 2002.
[29] J. Carletta, A. Isard, S. Isard, J.C. Kowtko, G. Doherty-Sneddon, A.H. Anderson, The reliability of a dialogue structure coding scheme, Computational Linguistics 23 (1) (1997) 13–32.
[30] A. Isard. An XML architecture for the HCRC maptask corpus, Proceedings of BI-DIALOG, (2001)
[31] R. Cooper, S. Ericsson, S. Larsson, I. Lewin, An information state approach to collaborative negotiation Proceedings of BI-DIALOG 2001 2001.

[32] H. Kamp, U. Reyle, From Discourse to Logic, Reidel, Dordrecht, Holland, 1993.

[33] Katzav J., Reed C.A., Rowe G.W.A., 'Argument Research Corpus, in: Proceedings of the Conference on Practical Applications of Linguistic Corpora (to appear) (2003).

[34] C. Reed, T.J. Norman, Argumentation Machines, Kluwer, Amsterdam, 2003.

[35] D.N. Walton, C. Reed, Diagramming, argumentation schemes and critical questions Proceedings of the 5th International Conference on Argumentation ISSA'2002 2002 SicSat, Ansterdan pp. 881–885.

[36] W.C. Mann, Dialogue macrogame theory Proceedings of the SIGdial Workshop 2002.

[37] E.C.W. Krabbe, Profiles of dialogue in: J. Gerbrandy, M. Marx, M. de Rijke, Y. Venema (Eds.), JFAK. Essays dedicated to Johan van Benthem on the Occasion of his 50th Birthday, Amsterdam University Press, Amsterdam, 1999.

[38] S. Jackson, Beyond web course automation: designing online dialogue Working Notes of NAU/Web'98 1998.

[39] D.R. Traum, J.F. Allen, Discourse obligations in dialogue processing Proceedings of ACL'94 1994.