Arguments, dialogue, and negotiation

Leila Amgoud[1] and **Simon Parsons**[2] and **Nicolas Maudet**[3]

Abstract. In the past few years there have been a number of proposals for mechanisms for negotiation between agents that make use of argumentation. These proposals have largely been vague on the subject of how the generation and interpretation of arguments fits into the process of negotiation. This paper addresses this gap, proposing a particular protocol which is suitable for negotiation, and illustrating its use on an example from the literature.

1 Introduction

Negotiation is widely regarded as a key issue in building multi-agent systems. In most agent applications, the autonomous components need to interact with one another because of the inherent interdependencies which exist between them, and negotiation is the predominant mechanism for achieving this. In recent years, there have been a number of suggestions for systems of negotiation based upon argumentation, including work by Parsons and Jennings [7, 10], Reed [9], Sycara [11] and Tohmé [12].

All mechanisms for negotiation have at their heart an exchange of offers. Agents make offers that they find acceptable and respond to offers made to them. What distinguishes argumentation-based negotiation from other approaches is the fact that offers can be supported by arguments, which, broadly speaking, equate to explanations for why the offer was made. This permits greater flexibility than in other negotiation schemes since, for instance, it makes it possible to persuade agents to change their view of an offer by introducing new factors in the middle of a negotiation (just as a car salesperson might throw in free insurance to clinch a deal).

While this use of argumentation is a common theme in all the work mentioned above, none of those proposals explain when arguments can be used within a negotiation and how they should be dealt with by the agent that receives them. Thus the protocol for handling arguments is missing. This paper fills the gap by proposing an argumentation protocol which permits the same kind of reasoning as the system proposed in [7], and which can be used to underpin the negotiation illocutions introduced in [10].

2 A system of argumentation

In this section we briefly introduce the system of argumentation which forms the backbone of our approach. This is inspired by the work of Dung [5] but goes further in dealing with preferences between arguments. Further details are available in [1]. We start with a possibly inconsistent knowledge base Σ with no deductive closure. We assume Σ contains formulas of a propositional language $\mathcal{L}$. $\vdash$ stands for classical inference and $\equiv$ for logical equivalence.

Definition 1 *An* argument *is a pair* (H, h) *where* h *is a formula of* $\mathcal{L}$ *and* H *a subset of* Σ *such that i)* H *is consistent, ii)* $H \vdash h$ *and iii)* H *is minimal, so no subset of* H *satisfying both i) and ii) exists.* H *is called the* support *of the argument and* h *is its* conclusion.

In general, since Σ is inconsistent, arguments in $\mathcal{A}(\Sigma)$, the set of all arguments which can be made from Σ, will conflict, and we make this idea precise with the notion of undercutting:

Definition 2 *Let* (H_1, h_1) *and* (H_2, h_2) *be two arguments of* $\mathcal{A}(\Sigma)$. (H_1, h_1) undercuts (H_2, h_2) *iff* $\exists h \in H_2$ *such that* $h \equiv \neg h_1$. *In other words, an argument is undercut iff there exists an argument for the negation of an element of its support.*

To capture the fact that some facts are more strongly believed (or desired, or intended, depending on the nature of the facts) we assume that any set of facts has a preference order over it which derives from the stratification of the knowledge base Σ into non-overlapping sets $\Sigma_1, \ldots, \Sigma_n$ such that facts in Σ_i are all equally preferred and are more preferred than those in Σ_j where $j > i$. The preference level of a nonempty subset H of Σ, $level(H)$, is the number of the highest numbered layer which has a member in H.

Definition 3 *Let* (H_1, h_1) *and* (H_2, h_2) *be two arguments in* $\mathcal{A}(\Sigma)$. (H_1, h_1) *is* preferred *to* (H_2, h_2) *according to Pref iff* $level(H_1) \leq level(H_2)$.

We can now define the argumentation system we will use:

Definition 4 *An* argumentation system *(AS) is a triple* $\langle \mathcal{A}(\Sigma), Undercut, Pref \rangle$ *such that* $\mathcal{A}(\Sigma)$ *is a set of the arguments built from* Σ, *Undercut is a binary relation representing defeat relationship between arguments,* $Undercut \subseteq \mathcal{A}(\Sigma) \times \mathcal{A}(\Sigma)$, *and Pref is a (partial or complete) preordering on* $\mathcal{A}(\Sigma) \times \mathcal{A}(\Sigma)$. $\gg^{Pref}$ *stands for the strict pre-order associated with Pref.*

Example 1 *Let* $\Sigma = \Sigma_1 \cup \Sigma_2 \cup \Sigma_3$ *with* $\Sigma_1 = \{\neg a\}, \Sigma_2 = \{a, a \to b\}$ *and* $\Sigma_3 = \{\neg b\}$. *Now,* $(\{\neg a\}, \neg a)$ *and* $(\{a, a \to b\}, b)$ *are two arguments of* $\mathcal{A}(\Sigma)$. *The argument* $(\{\neg a\}, \neg a)$ *undercuts* $(\{a, a \to b\}, b)$. *The preference level of* $\{a, a \to b\}$ *is 2 whereas the preference level of* $\{\neg a\}$ *is 1, and so* $(\{\neg a\}, \neg a\}) \gg^{Pref} (\{a, a \to b\}, b)$.

The preference order makes it possible to distinguish different types of relation between arguments:

Definition 5 *Let A, B be two arguments of* $\mathcal{A}(\Sigma)$.

B strongly undercuts *A iff B undercuts A and it is not the case that* $A \gg^{Pref} B$.

[1] Department of Electronic Engineering, Queen Mary and Westfi eld College, University of London, London E1 4NS, United Kingdom
[2] Department of Computer Science, University of Liverpool, Chadwick Building, Peach Street, Liverpool L69 7ZF, United Kingdom
[3] IRIT, ENSEEIHT, 2 rue C. Camichel, 31071 Toulouse Cedex, France

If B undercuts A then A defends itself *against B iff A* $\gg^{Pref}$ *B.*
A set of arguments $\mathcal{S}$ defends *A if there is some argument in* $\mathcal{S}$ *which strongly undercuts every argument B where B undercuts A and A cannot defend itself against B.*

Henceforth, $C_{Undercut,Pref}$ will gather all non-undercut arguments and arguments defending themselves against all their undercutting arguments. In [2], it was shown that the set $\underline{\mathcal{S}}$ of acceptable arguments of the argumentation system $\langle \mathcal{A}(\Sigma), Undercut, Pref\rangle$ is the least fixpoint of a function $\mathcal{F}$:

$$\begin{aligned} \mathcal{S} &\subseteq \mathcal{A}(\Sigma) \\ \mathcal{F}(\mathcal{S}) &= \{(H,h) \in \mathcal{A}(\Sigma) | (H,h) \text{ is defended by } \mathcal{S}\} \end{aligned}$$

Definition 6 *The set of* acceptable *arguments for an argumentation system* $\langle \mathcal{A}(\Sigma), Undercut, Pref\rangle$ *is:*

$$\begin{aligned} \underline{\mathcal{S}} &= \bigcup \mathcal{F}_{i\geq 0}(\emptyset) \\ &= C_{Undercut,Pref} \cup \left[\bigcup \mathcal{F}_{i\geq 1}(C_{Undercut,Pref})\right] \end{aligned}$$

An argument is acceptable if it is a member of the acceptable set.

Example 2 *(follows Example 1) The argument* $(\{\neg a\}, \neg a)$ *is in* $C_{Undercut,Pref}$ *because it is preferred (according to Pref) to the unique undercutting argument* $(\{a\}, a)$*. Consequently,* $(\{\neg a\}, \neg a)$ *is in* $\underline{\mathcal{S}}$*. The argument* $(\{\neg b\}, \neg b)$ *is undercut by* $(\{a, a \to b\}, b)$ *and does not defend itself. On the contrary,* $(\{\neg a\}, \neg a)$ *undercuts* $(\{a, a \to b\}, b)$ *and* $(\{\neg a\}, \neg a) \gg^{Pref} (\{a, a \to b\}, b)$*. Therefore,* $C_{Undercut,Pref}$ *defends* $(\{\neg b\}, \neg b)$ *and consequently* $(\{\neg b\}, \neg b) \in \underline{\mathcal{S}}$*.*

The set of acceptable arguments mutually defend one another:

Definition 7 *Let A, B be two arguments of* $\mathcal{A}(\Sigma)$ *and* $\mathcal{S} \subseteq \mathcal{A}(\Sigma)$*, then A* disqualifies *B iff A strongly undercuts B and B does not strongly undercut A.* $\mathcal{S}$ strictly defends *A iff for all B such that B strongly undercuts A, then there is a* $C \in \mathcal{S}$ *such that C disqualifies B.*

Theorem 1 $\forall (H,h) \in \underline{\mathcal{S}}, \underline{\mathcal{S}}$ *strictly defends* (H,h)*.*

The proof of this theorem can be found in [1].

3 Argument and Dialogue

In practice we don't need to enumerate all the set of acceptable arguments in order to know the status of a given argument and this can be exploited [1] to give a proof theory for the system. The basic idea is to traverse the sequence $\mathcal{F}_1, ..., \mathcal{F}_n$ in reverse. Consider that A occurs for the first time in $\mathcal{F}_n$. We start with A, and then for any argument B_i which strongly undercuts A, we find an argument A_i in $\mathcal{F}_{n-1}$ which defends A. Now, because of Theorem 1, we are only interested in the strict defenders of an argument, and the strict defenders of A will disqualify the B_i. The same process is repeated for each strict defender until there is no strict defender or defeater.

We can think of this process in terms of a dialogue game between two players P and C. P makes the argument we are interested in and its defenders and the player C makes the counter-arguments or defeaters.

Definition 8 *An* argument dialogue[4] *is a nonempty sequence of moves,* $move_i = (Player_i, Arg_i)(i \geq 0)$ *such that:*

1. $Player_i = P$ *iff i is even,* $Player_i = C$ *iff i is odd.*
2. $Player_0 = P$ *and* $Arg_0 = A$*.*
3. *If* $Player_i = Player_j = P$ *and* $i \neq j$ *then* $Arg_i \neq Arg_j$*.*
4. *If* $Player_i = P$*,* $i > 1$*, then* Arg_i *disqualifies* Arg_{i-1}*.*
5. *If* $Player_i = C$ *then* Arg_i *attacks* Arg_{i-1}*.*

An argument dialogue tree *is a finite tree where each branch is an argument dialogue.*

[4] In [1] this is called simply a "dialogue"; here we use the term "argument dialogue" to distinguish these dialogues from those discussed later. We omit the term "argument" when it is clear that we mean an argument dialogue.

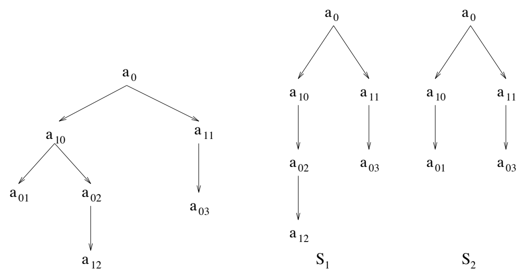

Figure 1. An argument dialogue tree and its candidate sub-trees

Example 3 *Let* $\langle \mathcal{A}, \text{Undercut}, Pref\rangle$ *be an AS such that* $\mathcal{A} = \{a_0, a_{01}, a_{02}, a_{10}, a_{11}, a_{12}\}$*, Undercut* $= \{(a_{10}, a_0), (a_{01}, a_{10}), (a_{12}, a_{02}), (a_{02}, a_{10}), (a_{03}, a_{11}), (a_{11}, a_0)\}$*. Let's suppose that* $a_{03} \gg^{Pref} a_{11} \gg^{Pref} a_0$*,* $a_{01} \gg^{Pref} a_{10} \gg^{Pref} a_0$ *and* $a_{12} \gg^{Pref} a_{02}$*,* $a_{02} \gg^{Pref} a_{10}$*. We are interested in the status of the argument* a_0*. The corresponding argument dialogue tree is presented in Figure 1 (left).*

The argument dialogue tree can be considered as an AND/OR tree. A node corresponding to the player P is an AND node, and a node corresponding to the player C is an OR node. This is because an argument is acceptable if it is defended against all its defeaters. The edges of a node containing an argument of P represent defeaters so they all must be defeated. In contrast, the edges of a node containing an argument of C represent defenders of P so it is sufficient that one of them defeats the argument of C.

Definition 9 *A player* wins *an argument dialogue iff he makes the last argument in the dialogue.*

A player who wins a dialogue does not necessarily win in all the subtrees of the dialogue tree. To formalize the winning of a dialogue tree, the concept of a solution sub-tree is defined.

Definition 10 *A* candidate sub-tree *is a sub-tree of an argument dialogue tree containing all the edges of each AND node and exactly one edge of each OR node. A* solution sub-tree *is a candidate sub-tree whose branches are all won by P.*

Thus the dialogue represented in example 3 has exactly two candidate sub-trees S_1 and S_2, Figure 1 (right).

Definition 11 *P* wins *an argument dialogue iff the corresponding dialogue tree has a solution sub-tree.*

Thus P wins the dialogue presented in Figure 1 because S_2 is a solution sub-tree.

Definition 12 *An argument A is* justified *iff there is an argument dialogue tree whose root is A, and which is won by the player P.*

Thus the argument a_0 is justified because the player P won the dialogue tree. The main result from the proof theory is:

Theorem 2 *Let $\langle \mathcal{A}, Undercut, Pref \rangle$ be an argumentation system. (i) $\forall x \in \mathcal{A}$, if x is justified then each argument of P belonging to the solution sub-tree is in $\underline{\mathcal{S}}$, in particular x. (ii) $\forall x \in \underline{\mathcal{S}}$, x is justified.*

In other words, the dialogue process constructs all acceptable arguments, and only constructs acceptable arguments and is thus sound and complete. The proof may be found in [1].

4 Towards multi-agent dialogues

The players P and C in the argument dialogue are not real individuals, they are merely a useful way of thinking about the construction of arguments. However, the idea of an argument dialogue can be extended to capture true dialogues by letting P and C be separate agents, as discussed in [3]. Here we extend that work by showing how specific support for realistic negotiation dialogues of the kind introduced in [7] and [10] can be added.

As in [7] we assume that each agent has a set of beliefs, B, a set of desires, D, and a set of intentions, I. Each of these sets is equipped with a (total or partial) preordering representing the preferences of the agent[5]. The basic knowledge base of an agent P is then $\Sigma_P^B = B_P \cup D_P \cup I_P$. Using Σ_P^B, P can build arguments concerning its own beliefs, desires and intentions as discussed above. However, we are more concerned with P's dialogical interactions with other agents.

To capture the dialogues between these agents we follow [3] in using a variant of the dialogue system DC introduced by MacKenzie [6]. In this scheme, agents make dialogical moves by asserting facts into and retracting facts from *commitment stores* (CSs) which are visible to other agents. Thus for a dialogue between P and another agent C, each agent "knows" everything in its own knowledge base, and everything in both commitment stores. Thus the overall knowledge available to P at any point in time is $\Sigma_P = \Sigma_P^B \cup \Sigma_C^{B'}$ where $\Sigma_C^{B'} \subseteq \Sigma_C^B$ and $\Sigma_C^B = B_C \cup D_C \cup I_C$. The contents of the two commitment stores $CS(P)$ and $CS(C)$ can be considered to be the state of the dialogue at any given point, and the CS of a single agent is the set of things it has agreed to.

For the moment we deal with propositional knowledge and assume we know that a particular proposition i is, for example, one of P's intentions because it resides in I_P, not because it is explicitly denoted as such. We do this for notational simplicity, bearing in mind that the problem of how to deal with first-order argumentation in which beliefs, desires and intentions are explicitly denoted is considered in depth in [8]. The latter also describes the logical machinery necessary to maintain consistency between these parts of the knowledge base—here we just assume the adoption of such techniques.

Despite these assumptions, the propositional language can usefully be extended to represent the type of information exchanged between agents in negotiation. As discussed in [10], negotiations often involve trade-offs with one agent accepting a request from another agent provided that this last accepts its request. For example: "If you let me use your laptop, I'll let use my printer", or "If you lend me a hammer, I'll give you a nail". To make it easier to represent this kind of information we introduce a new connective $\Rightarrow$. Thus we have a new language $\mathcal{L}'$ which contains propositional formulae and formulae $p \Rightarrow q$ such that p and q are propositional formulae. This connective allows us to capture "If you let me use your laptop, I'll let you use my printer" in the formula *Let_use_laptop* $\Rightarrow$ *Let_use_printer* and the latter is a formula of $\mathcal{L}'$.

[5] For now we assume that the agents concerned share the same preferences. We have considered how different sets of preferences can be combined elsewhere [4].

5 Dialogue moves

As mentioned above our work is inspired by MacKenzie's system DC. In that system, at each stage of the dialogue a participant has a set of legal moves it can make—asserting facts, challenging conclusions, asking for evidence, and so on. What we do here is to take a subset of the moves from DC which we have found useful in agent dialogues, and augment them with some new moves. We let $\mathcal{M}'$ denote the complete set of moves. For each move we describe how the move updates the CSs (the update rules), give the legal next steps possible by the other agent (the dialogue rules), and detail the way that the move integrates with the agent's use of argumentation (the rationality rules).

In the following descriptions, we suppose that agent P addresses the move to the agent C. The AS is therefore $\langle \mathcal{A}(\Sigma_P), Undercut, Pref \rangle$.

5.1 Basic dialogue moves

assert(p) where p is any formula in $\mathcal{L}$. This allows the exchange of information, such as "the weather is beautiful" or "It is my intention to hang a picture".

rationality the agent uses the AS to check if there is an acceptable argument for the fact p.

dialogue the other agent can respond with:

1. *accept*(p),
2. *assert*$(\neg p)$,
3. *challenge*(p).

update $CS_i(P) = CS_{i-1}(P) \cup \{p\}$ and $CS_i(C) = CS_{i-1}(C)$. This information is added to the CS of the agent making the assertion.

Note that C can only make a response if the rationality rule for that response is satisfied. Thus it can only respond to *assert*(p) with *assert*$(\neg p)$ if it has an acceptable argument for $\neg p$.

assert(S) where S is a set of formulae in $\mathcal{L}$ representing the support of an argument. Note that in DC players can only assert one propositional formula so are unable to produce a supporting argument in one step.

rationality the agent uses the AS to check if the related argument is acceptable.

dialogue the other agent can play:

1. *accept*(S),
2. *assert*$(\neg q)$, where $q \in S$,
3. *challenge*(q), where $q \in S$,
4. *promise*$(q \Rightarrow r)$, where $q \in S$.

update $CS_i(P) = CS_{i-1} \cup S$ and $CS_i(C) = CS_{i-1}(C)$.

Informally, this means that the responding agent can accept the whole support, challenge or deny an element of the support, or promise something in exchange for an element of the support. The next two moves allow an agent to elicit a response.

question(p) where p is a formula in $\mathcal{L}$.

rationality There is no rationality condition.

dialogue The other player can:

1. *assert*(p),
2. *assert*$(\neg p)$,
3. *question*(q),
4. *request*(q).

update $CS_i(P) = CS_{i-1}(P)$ and $CS_i(C) = CS_{i-1}(C)$.

question(p) denotes P asking if p is the case. C can answer either affirmatively (if it can show it to be the case) or negatively, by asking another question, or by making a request.

challenge(p) where p is a formula in $\mathcal{L}$.

rationality There is no rationality condition.

dialogue the other player can only *assert*(S) where S is the support of the argument (S, p), or S is the support of the argument (S, h) such that p belongs to S and h is one of P's intentions.

update $CS_i(P) = CS_{i-1}(P)$ and $CS_i(C) = CS_{i-1}(C)$.

5.2 Negotiation moves

The following moves are negotiation specific—while not strictly necessary for negotiation, they make it easier to capture some of the statements we wish our agents to make.

request(p) where p is any formula in $\mathcal{L}$.

rationality P uses the AS to identify a p in $\Sigma_C^{B'}$ such that $p \in H$ and (H, h) is an argument for one of P's intentions.

dialogue The other player can:

1. *accept*(p),
2. *refuse*(p),
3. *challenge*(p),
4. *promise*$(q \Rightarrow p)$.

update $CS_i(P) = CS_{i-1}(P)$ and $CS_i(C) = CS_{i-1}(C) \cup \{p\}$. A request is stored in the CS of the receiving agent because, if accepted, it becomes a commitment on that agent.

A *request* is invoked when an agent cannot, or prefers not to, achieve its intentions alone. The proposition requested differs from an asserted proposition in that it cannot be proved true or false—the decision on whether to accept it or not hinges upon the relation it has to C's intentions (see below).

promise$(p \Rightarrow q)$ where p and q are formulae in $\mathcal{L}$.

rationality P uses the AS to identify a p in $\Sigma_C^{B'}$ such that $p \in H$ and (H, h) is an acceptable argument for one of P's intentions, and to check that there is no acceptable argument (H', h') for one of its intentions h' such that $q \in H'$.

dialogue The other player can:

1. *accept*$(p \Rightarrow q)$,
2. *refuse*$(p \Rightarrow q)$,
3. *promise*$(s \Rightarrow p)$,
4. *challenge*(q).

update $CS_i(P) = CS_{i-1}(P) \cup \{q\}$
and $CS_i(C) = CS_{i-1}(C) \cup \{p\}$.

Broadly speaking, an agent will make a promise when it needs to request something from another, and has something it does not need (because the thing is not needed to achieve any intentions) which it can offer in return. In replying to a promise, an agent can accept, refuse, question why the requested thing is required, or suggest an alternative trade (C replying with $s \Rightarrow p$ is equivalent to P retracting its initial promise and replacing it with $p \Rightarrow s$).

5.3 Responding moves

The following are moves which are made in response to requests and assertions. The responses are context-specific, depending on the type of move made the turn before.

accept(p) where p is a formula in $\mathcal{L}$. After an assertion or request, an agent can respond with an explicit acceptance.

rationality In response to an assertion, P uses its AS to check if there is an acceptable argument for p. If so the move can be played. In response to a request, P has to check that there is no acceptable argument (H, h) for one of its intentions h, such that $p \in H$. In other words, it is only possible to accept a request if it doesn't invalidate the supporting argument for one of its intentions[6].

dialogue The other player can make any move except *refuse*.

update $CS_i(P) = CS_{i-1}(P) \cup \{p\}$ and $CS_i(C) = CS_{i-1}(C)$.

accept(S) S is a set of formulae in $\mathcal{L}$.

rationality P carries out the same rationality check for each $p \in S$ as it would if contemplating *accept*(p).

dialogue The other player can make any move except *refuse*.

update $CS_i(P) = CS_{i-1}(P) \cup S$ and $CS_i(C) = CS_{i-1}(C)$.

Accepting a set of formulae is just like accepting many individually.

accept$(p \Rightarrow q)$ where p and q are any formulae in $\mathcal{L}$.

rationality P carries out the same rationality check for p as it would if contemplating *accept*(p).

dialogue The other player can make any move except *refuse*.

update $CS_i(P) = CS_{i-1}(P) \cup \{p\}$
and $CS_i(C) = CS_{i-1}(C) \cup \{q\}$.

P checks that p does not scupper any of its plans, but is trusting enough to accept q. A less trusting agent might at least subject q to some check it would actually find q useful. Up to now we have not found this necessary.

refuse(p) where p is any formula in $\mathcal{L}$.

rationality P uses the AS to check if there is an acceptable argument (H, h) for one of its intentions h such that $p \in H$.

dialogue The other player can make any move except *refuse*.

update $CS_i(P) = CS_{i-1}(P) \setminus \{p\}$ and $CS_i(C) = CS_{i-1}(C)$.

Thus P will refuse requests which are necessary to achieve its intentions. There is also a *refuse* for promises:

refuse$(p \Rightarrow q)$ where p and q are any formulae in $\mathcal{L}$.

[6] As in [7] an argument for an intention is essentially a plan for achieving it, so allowing p would invalidate this plan.

rationality P uses the AS to check if there is an acceptable argument (H, h) for one of its intentions h such that $p \in H$.

dialogue The other player can make any move except *refuse*.

update $CS_i(P) = CS_{i-1}(P)\backslash\{q\}$
and $CS_i(C) = CS_{i-1}(C)\backslash\{p\}$.

As some proof of the utility of $\mathcal{M}'$ we can show:

Theorem 3 *The set of moves $\mathcal{M}'$ is sufficient to capture the communication language CL from [10].*

Proof (sketch) [10] proposed a communication language CL which contains the illocutions *request*, *accept*, *reject*, *offer*, *withdraw*, *appeal*, *threaten*, *reward*. The first three have obvious equivalents in $\mathcal{M}'$, *offer*, *withdraw* and *appeal* are syntactic sugar for *assert*s (both of single propositions and of support sets), while *reward* and *threaten* can be captured by *promise* since *promise*$(\neg p \Rightarrow q)$ threatens to do q unless the other does p. □

What this means is that $\mathcal{M}'$ can be used as a means of implementing the communication language CL in such a way that the protocol for building and interpreting arguments to support illocutions, lacking in [10], is provided.

6 An example negotiation dialogue

To give a flavour of the kind of negotiations which can be captured using $\mathcal{M}'$, we give an example based upon the home-improvement agents introduced in [7] demonstrating that our approach permits the same kind of reasoning. Agent P has the intention of hanging a picture, knows how to do this using a nail, but lacks a nail with which to do this. Agent C has the intention of hanging a mirror, knows how to do this, and has all the necessary resources to do so. One of these is a lone nail, which P has its eye on. The following dialogue ensues:

P: Please give me a nail.
C: No.
P: Why won't you give it to me?
C: Because I want to hang a mirror and for that I need a nail.
P: I understand.

In terms of our framework, the following takes place. First P tries to build an argument for its intention to hang the mirror and finds it needs a nail to do this. As a result it makes the move *request*$(nail)$, which inserts *nail* into CS(C). C finds that *nail* is part of its only argument to achieve its own intention, so replies with *refuse*$(nail)$, removing *nail* from its commitment store, and P responds with *challenge*$(nail)$. C answers the question with:

$$assert(mirror, nail, nail \wedge mirror \rightarrow hang\_mirror)$$

which has the effect of placing these formulae in CS(C). P has no response which defeats this, so *accept*s it, and the formulae are added to CS(P). The dialogue may be extended along the lines of that in [7] with P promising to exchange a screw for the nail, *promise*$(nail \Rightarrow screw)$, C *question*ing why this is useful, and P asserting that it can be used to hang the mirror. C cannot defeat this final assertion and so *accept*s it.

7 Conclusion

This paper has introduced a set of dialogue moves based upon MacKenzie's system DC [6], and shown how these can be operationalized in terms of a system of argumentation. Thus each move is both precisely defined in terms of the arguments an agent can build, and it is clear what moves an agent is allowed to make at a given point in time (this can be determined from the set of acceptable arguments). The moves are a superset of those in [3], including additional moves which simplify the handling of negotiation dialogues.

The resulting set of moves makes it possible to capture the kind of negotiation exchanges proposed in [10] as the minimum suitable set for argumentation-based negotiation, and to engage in the kind of negotiations discussed in [7]. Thus these moves seem adequate for supporting negotiations based on argumentation. Our approach is not only equal in scope to those in [7, 10] (and indeed other argument-based approaches) but goes somewhat beyond them in directly relating the arguments to the negotiation through the operationalisation of the dialogue moves. As a result the moves are intimately connected to the arguments that an agent makes and receives.

The flip-side of this close connection is that the moves "hard wire" the way the agent behaves. For instance, they would need to be revised to capture selfish agents who do not give up resources, or to capture agents which accept arguments based on their place in the social order as in [10]. Mechanisms for allowing more flexible social attitudes are the topic of our current work, along with the extension of the base language to something more expressive than propositional logic.

Acknowledgements: This work was partially supported by the Commission of the European Communities under contract IST-1999-10948, Sustainable Lifecycles in Information Ecosystems.

REFERENCES

[1] L. Amgoud, *Contribution a l'integration des préferences dans le raisonnement argumentatif*, Ph.D. dissertation, Thèse de doctorat de l'Université Paul Sabatier, Toulouse, Juillet 1999.

[2] L. Amgoud and C. Cayrol, 'On the acceptability of arguments in preference-based argumentation framework', in *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence*, pp. 1–7, (1998).

[3] L. Amgoud, N. Maudet, and S. Parsons, 'Modelling dialogues using argumentation', in *International Conference on Multi-Agent Systems*, Boston, MA, (2000).

[4] L. Amgoud and S. Parsons, 'An argumentation framework for merging conflicting knowledge bases', Technical report, Department of Electronic Engineering, Queen Mary and Westfield College, (2000).

[5] P. M. Dung, 'On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and *n*-person games', *Artificial Intelligence*, **77**, 321–357, (1995).

[6] J. MacKenzie, 'Question-begging in non-cumulative systems', *Journal of philosophical logic*, **8**, 117–133, (1979).

[7] S. Parsons and N. R. Jennings, 'Negotiation through argumentation—a preliminary report', in *Proceedings of the 2nd International Conference on Multi Agent Systems*, pp. 267–274, (1996).

[8] S. Parsons, C. Sierra, and N. R. Jennings, 'Agents that reason and negotiate by arguing', *Journal of Logic and Computation*, **8**(3), 261–292, (1998).

[9] C. Reed, 'Dialogue frames in agent communication', in *Proceedings of the 3rd International Conference on Multi Agent Systems*, pp. 246–253, (1998).

[10] C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons, 'A framework for argumentation-based negotiation', in *Proceedings of the 4th International Workshop on Agent Theories, Architectures and Languages*, pp. 167–182, (1997).

[11] K. Sycara, 'Argumentation: Planning other agents' plans', in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 517–523, (1989).

[12] F. Tohmé, 'Negotiation and defeasible reasons for choice', in *Proceedings of the Stanford Spring Symposium on Qualitative Preferences in Deliberation and Practical Reasoning*, pp. 95–102, (1997).