Agents that Reason and Negotiate by Arguing

SIMON PARSONS, CARLES SIERRA and NICK JENNINGS,
Department of Electronic Engineering, Queen Mary and Westfield College, University of London, London E1 4NS, U.K.
E-mail: {S.D.Parsons,C.A.Sierra,N.R.Jennings}@qmw.ac.uk

Abstract

The need for negotiation in multi-agent systems stems from the requirement for agents to solve the problems posed by their interdependence upon one another. Negotiation provides a solution to these problems by giving the agents the means to resolve their conflicting objectives, correct inconsistencies in their knowledge of other agents' world views, and coordinate a joint approach to domain tasks which benefits all the agents concerned. We propose a framework, based upon a system of argumentation, which permits agents to negotiate in order to establish acceptable ways of solving problems. The framework provides a formal model of argumentation-based reasoning and negotiation, details a design philosophy which ensures a clear link between the formal model and its practical instantiation, and describes a case study of this relationship for a particular class of architectures (namely those for belief–desire–intention agents).

Keywords: Argumentation, negotiation, multi-context systems, multi-agent systems, BDI agents.

1 Introduction

An increasing number of software applications are being conceived, designed, and implemented using the notion of autonomous agents. These applications vary from email filtering [26], through electronic commerce [35, 47], to large industrial applications [20]. In all of these disparate cases, however,the notion of autonomy is used to denote the fact that the software has the ability to decide for itself which goals it should adopt and how these goals should be achieved [48]. In most agent applications, the autonomous components need to interact with one another because of the inherent interdependencies which exist between them. The predominant mechanism for managing these interdependencies at run-time is negotiation—the process by which a group of agents communicate with one another to try and come to a mutually acceptable agreement on some matter [3]. Negotiation is so central precisely because the agents are autonomous. For an agent to influence an acquaintance, the acquaintance has to be persuaded that it should act in a particular way. The means of achieving this state are to make proposals, trade options, offer concessions, and (hopefully) come to a mutually acceptable agreement—in other words to negotiate.

We are interested in building autonomous agents which negotiate. This paper makes four main contributions towards this goal. The first is to outline a generic model of negotiation for autonomous agents which need to persuade one another to act in a particular way. The second is to describe an approach to building agent architectures which have a clear link between their specification and their implementation. Our

approach is founded upon the natural correspondence between multi-context systems [13], which allow distinct theoretical components to be defined and interrelated, and the modularity present in agent architectures. To demonstrate the power and flexibility of this approach a number of variants of the widely used Belief–Desire–Intention (BDI) agent model [32] are specified with the same conceptual structures. The third contribution is to provide a general system of argumentation suitable for use by multi-context agents in a multi-agent environment, and to describe a specific version of this system which may be used by multi-context BDI agents. This is necessary because the move to both multi-context agents and then to BDI agents introduces additional issues over and above those which are handled by existing systems of argumentation. The fourth contribution is to present a well-grounded framework for describing the reasoning process of negotiating agents. This framework is based upon the use of argumentation both at the level of an agent's internal reasoning and at the level of negotiation between agents. Such an approach has been advocated (in a discursive rather than formal manner) by Hewitt [15] as the most natural means of viewing the reasoning and operation of truly autonomous agents in open systems.

This paper builds on our previous work in the fields of negotiation and argumentation. It is an extension of the work presented in [28, 38] in that, respectively, it provides a tighter integration of argumentation and the mental model of the negotiating agents, and it deals with arguments which justify positions (in addition to basic statements about positions). It also fixes some technical problems with the model of argumentation presented in [28]. The work described here is also complementary to the work described in [39] in that it concentrates on the way in which arguments are built and analysed rather than on the communication language and the negotiation protocol.

The remainder of the paper is structured so that each major contribution is presented in a separate section. Section 2 introduces a general framework for describing negotiation. Section 3 shows how multi-context systems can be used to specify agent architectures in general and BDI architectures in particular. Section 4 presents a system of argumentation suitable for use by multi-context agents in multi-agent systems. Section 5 illustrates how this framework can be used for argumentation-based negotiation. Section 6 then places our work in the context of previous work in the fields of multi-agent systems, negotiation and multi-context systems. Section 7 concludes and outlines a number of issues which require further investigation.

2 A framework for negotiation

Examination of the literature on negotiation from the fields of social psychology [30], game theory [36], and distributed AI [3, 23], reveals a significant level of agreement on the main stages involved in the process. We use this commonality to underpin our generic negotiation model which is outlined below.

2.1 A generic model of negotiation

Negotiation is a process that takes place between two or more agents who are attempting to achieve goals which they cannot, or prefer not to, achieve on their own. These goals may conflict, in which case the agents have to bargain about which agent

achieves which goal, or the agents may depend upon one another to achieve the goals, in which case they only have to discuss how to go about achieving the goals. In either case, the process of negotiation proceeds by the exchange of proposals, critiques, explanations and meta-information [28].

A *proposal*, broadly speaking, is some kind of solution to the problem that the agents face. It may be a single complete solution, single partial solution, or a group of complete or partial solutions. A proposal may be made either independently of other agents' proposals, or based on previous comments made by other agents. The following is a typical proposal:

A: I propose that you provide me with service X.

Proposals can be more complex than just suggestions for joint action—they may include suggested trade-offs or suggest conditions under which the proposal holds. Thus the following are also proposals:

A: I propose that I will provide you with service Y if you provide me with service X.

A: I propose that I provide you with service Y if you agree to provide me with service X at a later date.

Proposals are thus the basic mechanism of any negotiation, and the way in which negotiations begin is by one agent making a proposal to another.

An agent that has received a proposal can respond in two possible ways. The first of these is by making a *critique*. A critique may just be a remark as to whether or not the proposal is accepted, or a comment on which parts of the proposal the agent likes and which parts it dislikes. The following short dialogues are examples of proposals followed by critiques:

A: I propose that you provide me with service X.

B: I accept.

where the critique is an immediate acceptance, and:

A: I propose that I will provide you with service Y if you provide me with service X.

B: I don't want service Y.

where the critique is intended to provoke an alternative proposal.

The process of generating the critique is the method by which the agent evaluates the proposal, and by returning some or all of the critique to the originating agent the responding agent aims to provoke alternative proposals that are more acceptable. Generally speaking, the more information placed in the critique, the easier it is for the original agent to respond in a manner which is likely to lead to agreement.

As an alternative to offering a critique of a proposal, an agent can respond with a *counter-proposal*. A counter-proposal is just a proposal which is made in response to a previous proposal.[1] The following are two examples of proposals followed by counter-proposals:

A: I propose that you provide me with service X.

[1]Thus every counter-proposal is a proposal, but not every proposal is a counter-proposal—the opening proposal in a negotiation is never a counter-proposal.

B: I propose that I provide you with service X if you provide me with service Z.

where the counter-proposal extends the initial proposal, and:

A: I propose that I provide you with service Y if you provide me with service X.
B: I propose that I provide you with service X if you provide me with service Z.

where the counter-proposal amends part of the initial proposal. Providing a counter-proposal thus involves generating and sending an alternative proposal (which should be more favourable to the responding agent than the original).

On their own, proposals, counter-proposals and critiques are bald statements of what agents want. We suggest that if agents provide an explanation along with their statements, agreement is more likely to be reached more quickly. An *explanation* is additional information explaining why a proposal, counter-proposal or critique was made that an agent can supply in support of that proposal, counter-proposal or critique. We see an explanation as being a form of justification that the agent supplies for its position. This may take the form of an argument with which the agent seeks to persuade whoever it is negotiating with that its suggestion is valid. However, it might also be a simple statement of why it reached that conclusion. The following are examples of parts of a negotiation in which agents supply explanations:

A: I propose that you provide me with service X because I know that is one of the services you offer.
B: I propose that I provide you with service X if you provide me with service Z, because providing X for you will mean that I incur costs to the value of Z.

where the dialogue takes the form of a proposal with explanation followed by a counter-proposal with explanation, and both explanations are statements of why the proposals are made, and:

A: I propose that I provide you with service Y if you provide me with service X. I think this is good for both of us because I need X and I believe that you need Y.
B: I don't need Y but I do need Z.
A: Okay, I propose that I provide you with service Z if you provide me with service X.

where a proposal with explanation is followed by a critique with explanation, and this is followed by a proposal. The explanation which accompanies the initial proposal is an attempt to persuade B that the proposal is a good idea.

The role of *meta-information* is to focus the local search by agents for solutions. Thus, by supplying information about why it had a particular objection to a proposal, one agent might help another to focus its search for another, more acceptable, suggestion. Clearly explanations are one form of meta-information, but we also allow for meta-information to be supplied in the absence of any proposal or critique:

A: I propose that I provide you with service Y if you provide me with service X. I think this is good for both of us because I need X and I believe that you need Y.
B: I'm not interested in Y.
A: Okay, I'll provide you with W if you provide me with X.

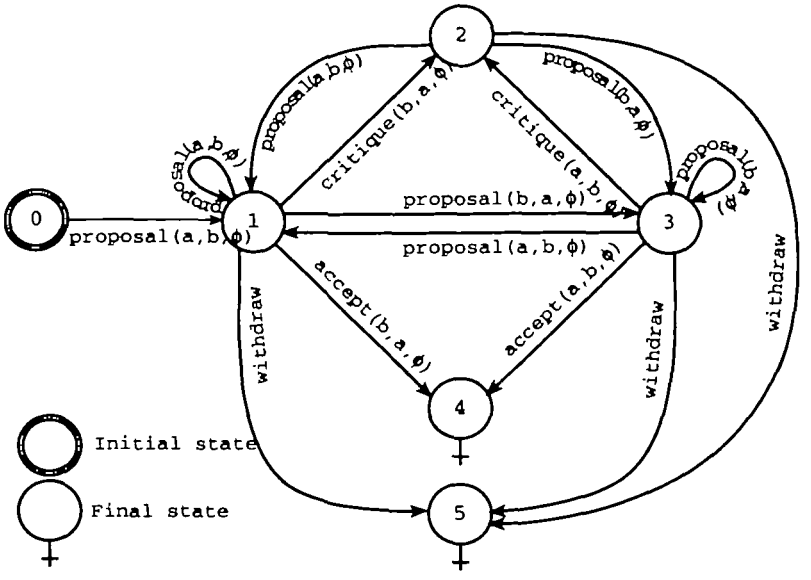

FIG. 1. The negotiation protocol for two agents

B: I'd rather have Z.

A: Right. I'll provide you with Z if you provide me with X

Here B's second utterance is a piece of meta-information which expresses a preference between services.

Given these concepts, the process of negotiation can be considered to be the following. The process starts when an agent generates a proposal. Other agents then either accept it, critique it, make counter-proposals, or provide meta-information. The first agent can then either make a fresh proposal, send clarifying meta-information, or respond to one of the counter-proposals (if any) with a counter-proposal to that proposal. This process continues until all the agents involved are, in some sense, happy with a proposal or it is felt that no agreement can be reached (in which case a participating agent withdraws). By 'happy' it is not meant that this is the optimum proposal from the point of view of the agent, but that it represents an acceptable compromise.

2.2 *A negotiation protocol*

Considering negotiation between just two agents, we can consider the process introduced above as defining the 'rules of encounter' which the two agents follow: In other words, it defines the basis of a protocol. Taking this a little further, we can specify the protocol as a form of state transition diagram which gives the various legal states that an agent may be in during a negotiation and thus the legal transitions between states which an agent is allowed to take. Such an analysis leads to Figure 1 which is broadly similar to that proposed in [39]. The process begins (State 0) when one agent makes a proposal to another, denoted by $\texttt{proposal}(\texttt{a},\texttt{b},\phi)$. Here ϕ denotes

both the proposal being made and any explanation the agent cares to give (it may, of course, choose to give none). As we shall see later, the idea of a pair of proposal and explanation maps onto the idea of an argument in the narrow technical sense in which we use the term 'argument' in this paper. Once the proposal has been sent (State 1) the first agent may make a second proposal without waiting for a reply, or the second agent can act, either by accepting the proposal, making a critique, making a counter-proposal or withdrawing from the process. When accepting, the agent may again give an explanation if desired, and a critique can also be supported by a reason. If a critique (State 2) or a counter-proposal (State 3) is made, either agent can keep the process moving by making another proposal which can then be responded to in the same way as the initial proposal. This process iterates until one of the agents '`accept`'s or '`withdraw`'s.

There are a few things that should be noted about this protocol as drawn in Figure 1. First, this is not an alternating offers model (unlike the model in [38]) in that agents may make counter-proposals without waiting for a response to a previous proposal. Second, the protocol makes no distinction between proposals and counter-proposals. Any of the proposals, except that which starts the negotiation, may be a counter-proposal, but equally, these other proposals need not be counter-proposals (since they may not be directly related to the original proposal). Third, the protocol includes two illocutions, `accept` and `withdraw` which are particular types of critique which bring the negotiation to a close. The string '`withdraw`' stands for both `withdraw(a, b)` and `withdraw(b, a)`. Thus the transition labelled by '`withdraw`' can be initiated by either agent. Fourth, as it stands the protocol only explicitly includes meta-information as explanations accompanying other utterances (in the ϕ in proposals and critiques). Since meta-information on its own can be supplied by any agent at any point, the reader is invited to imagine the diagram of the full protocol which includes a set of pairs of arcs from each state other than the first, back to that state. One arc of each pair would be labelled `meta-information(a, b,` ϕ`)`, and the other would be labelled `meta-information(b, a,` ϕ`)`.

2.3 Our proposal

This section has introduced a general way of describing negotiations between agents, where the term 'negotiation' is given the broad interpretation usual in the agent literature. In this sense, a negotiation is any dialogue between two or more agents which leads them to agree on some subject (for example a course of joint action, or the price for some service) about which they initially had different opinions.[2] What we are suggesting here is that if we consider agents to reason by using a particular formal system of argumentation then we get, almost for free, the basic support necessary to build agents which negotiate in the general sense described above. How this is done is the subject of Section 5. First we need to talk about how we propose to build agents which use this system of argumentation, and that in turn means that we need to describe how to build the kind of multi-context agents for which our system of argumentation is appropriate.

[2] Of course there are other ways of describing such dialogues, for instance that proposed by Walton and Krabbe [46], and in Section 5 we consider how this particular description relates to our proposal.

3 Specifying architectures for negotiating agents

There are many ways of designing and building agent systems. However, the most common means is probably through an agent architecture. The role of such architectures is to define a separation of concerns—they identify the main functions which ultimately give rise to the agent's behaviour and they define the interdependencies between them. This approach to system design affords all the traditional advantages of modularization in software engineering [40] and enables complex artifacts to be designed out of simpler components. However, one problem with much of the work on agent architectures is that it is somewhat ad hoc in nature. There is often little connection between the specification of the architecture and its implementation. This situation is clearly undesirable.

For this reason, we are looking to provide a means of developing agent architectures which have a clear link between their specification and their implementation. To do this, we make use of *multi-context systems* [13], a framework which allows distinct theoretical components to be defined and interrelated. We use different contexts to represent different components of an agent architecture, and specify the interactions between the components by means of the bridge rules between contexts. This makes it possible to move directly from the specification of the architecture to a formal description in terms of multi-context systems. Then, since each context contains a set of statements in a logic along with the axioms of that logic, it is possible to move directly to an implementation in which the various contexts are concurrent theorem provers which exchange information. In such an implementation, each theorem prover component corresponds directly to one of the components of the original architecture. This approach enforces a modular structure with well-defined interfaces, and thus accords well with good software engineering practice. To this end, Section 3.1 indicates the general method of using multi-context systems to specify agent architectures. Then Section 3.2 makes the discussion more concrete by indicating how a particular class of agent architecture—namely BDI agents—can be modelled with this approach. Finally, Section 3.3 provides an example of the specification of a pair of BDI agents in a particular domain.

3.1 Generic multi-context agents

Using the multi-context approach, an agent architecture consists of the following four components (see [27] for a formal definition):

- *Units*: Structural entities representing the main components of the architecture.
- *Logics*: Declarative languages, each with a set of axioms and a number of rules of inference. Each unit has a single logic associated with it.
- *Theories*: Sets of formulae written in the logic associated with a unit.
- *Bridge rules*: Rules of inference which relate formulae in different units.

Figure 2 shows an example architecture in which the units are u_1, u_2, u_3 and c. u_1 contains a propositional logic, u_2, u_3 and c contain a first order logic. No specific theories are given. The bridge rules are shown as arcs connecting the units.

Using the notation of [14], an agent is defined as a group of interconnected units represented by a pair $\langle \{u_i\}_{i \in I}, \Delta \rangle$ where I is the set of unit indices, u_i is the unit name

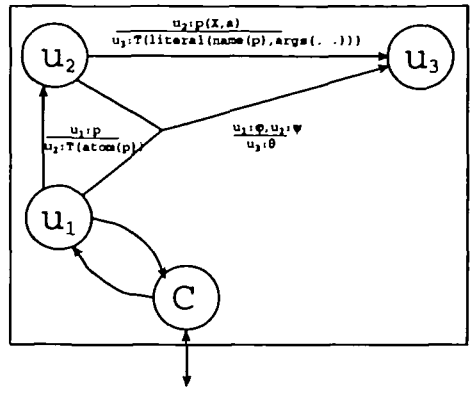

FIG. 2. An example multi-context agent

given to the tuple $\langle L_i, A_i, \Delta_i, T_i \rangle$, where L_i, A_i and Δ_i respectively are the language, axioms and rules of inference defining the logic, and T_i is the theory associated with the unit. Δ is the set of all bridge rules between the units.

Bridge rules can be understood as rules of inference with premisses and conclusions in different units. For instance:

$$\frac{u_1 : \varphi, u_2 : \psi}{u_3 : \theta}$$

means that formula θ may be deduced in unit u_3 if formulae φ and ψ are deduced in units u_1 and u_2 respectively (see Figure 2). We will also write such rules as

$$u_1 : \varphi, u_2 : \psi \Rightarrow u_3 : \theta,$$

where more convenient.

In our approach, bridge rules are used to enforce relations between the various components of the agent architecture. For example, in a BDI agent, a bridge rule between the intention unit and the belief unit might be

$$I : I(\alpha) \Rightarrow B : B(\lceil I(\alpha) \rceil),$$

meaning that if the agent has an intention α then it is aware of this fact. In this example, as in the specification of BDI agents developed later in the paper, the B and I are taken to be predicates in first-order logic. As a result, when a formula from the intention unit is embedded in the belief unit by means of a bridge rule, it is quoted using $\lceil \cdot \rceil$. This is the method for modelling modal logics as first-order theories proposed by Giunchiglia and Serafini [13].

In general, the nature of the units will vary between architectures. For example, a BDI agent may have units which represent theories of belief, desire and intention, whereas an architecture based on a functional separation of concerns may have units for cooperation, situation assessment and plan execution [19, 20]. However, for the purposes of this work, we assume that all agents have a dedicated *communication unit* (C in Figure 2) which is responsible for enacting the agent's communication needs. We assume the existence of this unit because: (i) we want to encapsulate

the agent's internal structure by having a unique and well-defined interface with the environment; and (ii) we wish to have a cognitive interpretation of the architecture—the communication unit acts metaphorically as the agent's sensors and actuators (eyes, mouth and ears) by means of which the agent's 'mind' is effectively situated in the environment.

Since the communication unit deals with both incoming and outgoing messages, we could split it into two units; one for incoming messages, and one for outgoing messages. However, we do not feel that this is necessary at the moment (though we do not rule out the possibility in the future). The reason for this is that we would like to keep the model relatively simple and so only introduce new units when either (i) they are necessary to capture different cognitive components (which is why we have different units for desires and intentions) or (ii) they are necessary to capture different logics (which is one reason why we have different units for beliefs and desires). At the moment we don't feel that either of these conditions apply to the different parts of the communication unit.

The formulae the agent can utter are thus determined by the language L_C used by the communication unit. In turn, L_C is the result of the nested embeddings that the different bridge rules make between the languages of the various units. In this sense, the bridge rules play a key role in the design of an architecture. As we will show in Section 3.2, important differences in behaviour can be attained simply by changing the pattern of 'combination' of the units. Moreover, interaction between agents is carried out exclusively by the interchange of illocutions. Listening to an illocution is a form of sensing and speaking is a form of action. Hence the communication unit is responsible for making effective the actions—illocutions—selected in the negotiation with the other agents.

The set of formulae that a given unit may contain depends on the unit's initial theory, axioms, inference rules and the incoming bridge rules. The formulae introduced by a bridge rule depend on the formulae present in the unit in the premiss of the bridge rule. These may, in turn, depend on the bridge rules leading to that unit, and so on. The communication unit will receive formulae from other agents that will contain new symbols, and so extend its alphabet [39]. In order to accommodate this dynamic expansion (which we believe to be the most natural way to enable flexible multi-agent communication) the language L_C must must be defined only partially. In addition, since formulae propagate from the communication unit to other units by means of the bridge rules, the languages of all other units must also be partial. The evolution of the reasoning process by the application of bridge rules and the communication between agents, extends these languages incrementally. For example, we can fix the set of predicates to be used in a certain language L_{FOL} but leave the definition of L_{FOL} parametric with respect to the terms the predicates may be applied over. By doing this, we under-specify the signature of L_{FOL}. For instance, we can declare a metapredicate (T) and then by means of bridge rules define which terms the predicate will apply over. The following:

$$\frac{u_1 : p}{u_2 : T(atom(p))}$$

is a bridge rule which embeds atoms of the theory of unit u_1 into the propositional metatheory of unit u_2, and

$$\frac{u_2 : p(X, a)}{u_3 : T(literal(name(p), args(variable(X), constant(a))))}$$

does a similar job in the case of a first order language defined as a metalanguage for u_2 in u_3 (in a similar way to that in which it is done in OMEGA [1]). The partial nature of the language is essential if the agents are to negotiate and argue, for these processes often involve the introduction of new concepts [39]. By definition, therefore, the agent's languages must be extensible.

An agent's deductive mechanism, $\vdash_i$, can be thought of as the relation between the utterances heard by the agent, the current theories of the agent's units and the utterances generated by the agent. This mechanism is realized by the use of an execution model based on the following assumptions:

1. *Concurrency.* The execution of each unit is a non-terminating deductive process (which may be formulated using dynamic logic [27]). All units execute concurrently. Moreover, the bridge rules are also concurrent processes. They examine the theories of the units in their premisses for sets of formulae that match them, whenever a new match is found the concluding formula is asynchronously added to the theory of its associated unit.[3]
2. *Reactivity.* The communication unit immediately processes (and thus adds to its theory) all messages it receives from other agents. This enables the agent to respond in an appropriate manner to important events which occur in the environment in which it is situated [2, 12].

3.2 Multi-context BDI agents

To provide a specific example of the method of approach advocated in the previous sub-section, we examine how a particular class of agent architecture—BDI agents—can be modelled. The particular theory on which the architecture is based is that of Rao and Georgeff. This model has evolved over time (as can be seen by comparing [33] and [34]) and in this section we account for the most recent approach [34] where three modalities are distinguished: B for beliefs—used to represent the state of the environment, D for desires—used to represent the motivations of the agent, and I for intentions—used to represent the ends (or goals) of the agent. In this work, we associate a separate unit for each of the modalities[4] (see Figure 3).

We could then give each of these units exactly the same interpretation as they are given in the Rao and Georgeff model—what we will refer to as the *direct interpretation*. This involves giving each modality a semantics in terms of possible worlds and the relation between modalities as relations between the associated possible worlds. This relation is often semantically modelled as inclusions between accessible worlds and syntactically modelled as axioms in the form of implications between the modalities.

[3] This asynchronous mechanism can be used to simulate synchronicity between the deduction of two units whenever necessary.

[4] In fact the general approach allows more than one unit for beliefs (as in [5]), desires or intentions if deemed appropriate. In the examples presented, however, this is not necessary.

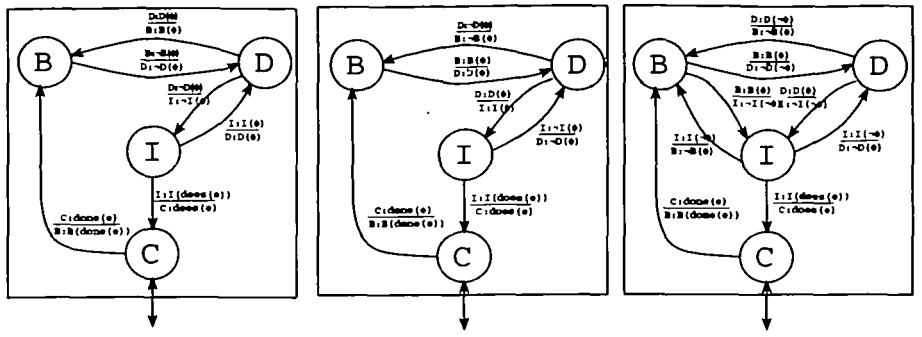

FIG. 3. Different types of BDI agent. From left to right, the relations between modalities correspond to strong realism, realism and weak realism

For instance, the fact that any intention-accessible world is also a belief-accessible world—the agent believes what it intends—is syntactically represented as $I(\alpha) \rightarrow B(\alpha)$. These implications have different deductive readings from each side of the connective (modus ponens or modus tollens) which is why some of the architectures we propose associate two bridge rules (in opposite directions) with each implication (see for instance Figure 3). In the direct interpretation, the logics in the B, D and I units embody the temporal logic CTL [10] (exactly as they do in Rao and Georgeff's model). In addition to the axioms of CTL which are common to all the units, each unit has its own axioms encoding the behaviour of the modality. In the examples of Figure 3, for instance, the axioms are the set K, D, 4 and 5 for B, and K and D for D and I.

This completes the discussion of the logics within each unit, and so we turn to considering the bridge rules. As stated above, the set of bridge rules determine the relationship between the modalities and hence the behaviour of the agent. Three well-established sets of relationships for BDI agents have been identified [34] (Figure 3):

- *Strong realism.* The set of intentions is a subset of the set of desires which in turn is a subset of the beliefs. That is, if an agent does not believe something, it will neither desire nor intend it [33].
- *Realism.* The set of beliefs is a subset of the set of desires which in turn is a subset of the set of intentions. That is, if an agent believes something, it both desires and intends it [7].
- *Weak realism.* A case in between strong realism and realism. Agents do not desire properties if the negation of those properties are believed, do not intend properties if the negation of those properties are desired, and do not intend properties if the negation of those properties are believed [31].

In addition to the bridge rules which relate beliefs, desires and intentions, we have rules relating intentions to formulae in the communication unit and formulae in the communication unit to beliefs. Now, the communication unit is responsible for communication between an agent and its peers, and is thus responsible for performing illocutionary acts, the contents of which are determined by the rest of the units in the architecture. In these BDI agents the communication unit will be responsible for

asking other agents to act in a particular way. This is expressed by the bridge rule that makes any intentions for somebody else to act, denoted by '$I(does(e))$' become a formula '$does(e)$' inside the communication unit. This unit, will then determine when and how to make this effective. Moreover, when an agent perceives (because it is told) that an action has been accomplished (denoted by the presence of '$done(e)$' in the communication unit), it makes the belief unit aware of this fact by means of another bridge rule acting upon the predicate '*done*'. Thus we consider that the argument of '*does*' and '*done*' contains information about the actions.

This completes the direct interpretation of the BDI model in terms of multi-context systems. However, this is not the interpretation we use in our work. Instead we prefer to build multi-context systems using an *indirect interpretation* in which the B, D and I are taken as predicates, as hinted at in Section 3.1. Such systems again have separate B, D and I units (along with a communication unit), and use the same sets of bridge rules as discussed above (exactly which set depends upon the kind of realism we want for our agents). To show exactly what we mean and to illustrate the process of defining a multi-context agent, we provide the specification of a strong realist BDI agent. We start by recalling that there are four components we need to specify:

1. the units present in the agent;
2. the logics in each unit;
3. the theories written in each logic in each unit; and
4. the bridge rules connecting the units.

For a strong realist BDI agent these four components are as follows.

Units: As discussed above, there are four units within a multi-context BDI agent, the communication unit, and units for each of the beliefs, desires and intentions.

Logics: For each of these four units we need to say what the logic used by each unit is. The communication unit uses classical first-order logic with the usual axioms and rules of inference. The belief unit also uses first-order logic, but with a special predicate B which is used to denote the beliefs of the agent. As mentioned above, under the modal logic interpretation of belief, the belief modality is taken to satisfy the axioms K, D, 4 and 5 [34]. Therefore, to make the belief predicate capture the behaviour of this modality, we need to add the following axioms to the belief unit (adapted from [4]):

$$\begin{array}{ll} \mathbf{K} & B : B(\varphi \rightarrow \psi) \rightarrow (B(\varphi) \rightarrow B(\psi)) \\ \mathbf{D} & B : B(\varphi) \rightarrow \neg B(\neg\varphi) \\ \mathbf{4} & B : B(\varphi) \rightarrow B(B(\varphi)) \\ \mathbf{5} & B : \neg B(\varphi) \rightarrow B(\neg B(\varphi)). \end{array}$$

The desire and intention units are also based on first-order logic, but have the special predicates D and I respectively. The usual treatment of desire and intention modalities is to make these satisfy the K and D axioms [34], and we capture this by adding the relevant axioms. For the desire unit:

$$\begin{array}{ll} \mathbf{K} & D : D(\varphi \rightarrow \psi) \rightarrow (D(\varphi) \rightarrow D(\psi)) \\ \mathbf{D} & D : D(\varphi) \rightarrow \neg D(\neg\varphi) \end{array}$$

and for the intention unit:

$$\mathbf{K} \quad I : I(\varphi \rightarrow \psi) \rightarrow (I(\varphi) \rightarrow I(\psi))$$
$$\mathbf{D} \quad I : I(\varphi) \rightarrow \neg I(\neg\varphi).$$

Each unit also contains the usual rules of inference including *generalization*, *particularization*, *modus tollens* and *modus ponens*. This completes the specification of the logics used by each unit.

Theories: For each of the four units we need to specify what logical expressions, written in the language of each unit, are present in each unit. This information can be seen as the domain information possessed by each unit, and since here we are making a general statement about what goes into every strong realist BDI agent built using our framework, it is no surprise to find that there are no specific theories that we include.

Bridge rules: The bridge rules are exactly those given in Figure 3 for strong realist BDI agents.

$$\begin{aligned} I : I(\alpha) &\Rightarrow D : D(\lceil\alpha\rceil) \\ D : \neg D(\alpha) &\Rightarrow I : \neg I(\lceil\alpha\rceil) \\ D : D(\alpha) &\Rightarrow B : B(\lceil\alpha\rceil) \\ B : \neg B(\alpha) &\Rightarrow D : \neg D(\lceil\alpha\rceil) \\ C : done(e) &\Rightarrow B : B(\lceil done(e)\rceil) \\ I : I(\lceil does(e)\rceil) &\Rightarrow C : does(e). \end{aligned}$$

The first four of these are derived directly from the model proposed by Rao and Georgeff and ensure consistency between what is believed, desired and intended. Thus the first ensures that anything the agent intends to do is also something it desires to take place, and the second ensures that anything it does not desire is never adopted as an intention. The last two bridge rules specify the interactions between the communication unit and other units as discussed above. Note that, as discussed previously, all the bridge rules result in quoted formulae. Because in this interpretation we always quote formulae in the conclusion of bridge rules, from this point on we simplify our notation by leaving the quotation implicit.

3.3 Home improvement BDI agents

Having described how to capture strong realist BDI agents in our multi-context framework, we continue by introducing an example which involves two such agents. This example will then be used later in the paper in order to illustrate our scheme for negotiation. It should be noted that the example is intended to be illustrative rather than persuasive in that it shows how our multi-context approach can be used to specify the agents, rather than showing that the agents can only be set up using our approach. Clearly it is possible to set up the agents without making them BDI agents. Furthermore, we acknowledge that the knowledge the agents use to reason about their world is rather simplistic. We could, of course, use more realistic theories of actions and planning but we feel that this would rather cloud the issue since:

(i) such theories would make the example more complicated; and

(ii) the usefullness of both BDI models in general and our approach in particular hinges more on the fact that they make it possible to clearly specify agents in general than on the specifics of those agents' approaches to representing actions.

Throughout the example there is some scope for confusion. The reason is that there are two languages in operation here. The first is the language of the B, D and I predicates in which the connectives are those of first-order logic. The second is the language quoted within the scope of the B, D and I predicates in which all the connectives are just terms of the relevant predicates. In this second language, conjunction is as usual, but $\rightarrow$ does not represent material implication. Instead it represents the relationship, admittedly a rather naive one, between the goals of the agent and the means the agent has to achieve them.

The example concerns two home improvement agents which are strong realists in the sense introduced above. Both have some information about what they seek to achieve. Agent a has the intention of hanging a picture, and it has various beliefs about resources and how they can be used to hang mirrors and pictures:

$$I : I_a(Can(a, hang\_picture)) \tag{3.1}$$

$$B : B_a(Have(a, picture)) \tag{3.2}$$

$$B : B_a(Have(a, screw)) \tag{3.3}$$

$$B : B_a(Have(a, hammer)) \tag{3.4}$$

$$B : B_a(Have(a, screwdriver)) \tag{3.5}$$

$$B : B_a(Have(b, nail)) \tag{3.6}$$

$$B : B_a(Have(X, hammer) \wedge Have(X, nail) \wedge Have(X, picture) \rightarrow Can(X, hang\_picture)) \tag{3.7}$$

$$B : B_a(Have(X, screw) \wedge Have(X, screwdriver) \wedge Have(X, mirror) \rightarrow Can(X, hang\_mirror)). \tag{3.8}$$

Note that we write $B(\varphi_1 \wedge \ldots \wedge \varphi_n)$ for $B(\varphi_1) \wedge \ldots \wedge B(\varphi_n)$ (similarly for other connectives) and that we subscript the belief, desire and intention predicates with the agent name.

This information comprises part of the theories of Agent a's intention and belief units. In particular (3.1) is the initial theory in the intention unit and (3.2–3.8) is part of the initial theory in the belief unit (we say 'part' because other bits of the theory will be introduced later). Similarly, agent b intends to hang a mirror and has various beliefs about its resources and the action of hanging mirrors:

$$I : I_b(Can(b, hang\_mirror)) \tag{3.9}$$

$$B : B_b(Have(b, mirror)) \tag{3.10}$$

$$B : B_b(Have(b, nail)) \tag{3.11}$$

$$B : B_b(Have(X, hammer) \wedge Have(X, nail) \wedge Have(X, mirror) \rightarrow Can(X, hang\_mirror)). \tag{3.12}$$

Both agents also have a simple theory of action that integrates a model of the available resources with their planning mechanism and forms another part of the theory

contained in their belief units. This theory needs to model the following ideas (with $i \in \{a, b\}$):

Ownership. When an agent (X) is the owner of an artifact (Z) and it gives Z to another agent (Y), Y becomes its new owner:

$$B : B_i(Have(X, Z) \wedge Give(X, Y, Z) \rightarrow Have(Y, Z)). \tag{3.13}$$

Unicity. When an agent (X) gives an artifact (Z) away, it no longer owns it:[5]

$$B : B_i(Have(X, Z) \wedge Give(X, Y, Z) \rightarrow \neg Have(X, Z)). \tag{3.14}$$

Benevolence. When an agent i has something (Z) that it does not intend to use and is asked to give it to another agent (X), i adopts the intention of giving Z to X. Naturally more complex cooperative strategies can be defined if desired:

$$B : B_i(Have(i, Z) \wedge \neg I_i(Have(i, Z)) \wedge Ask(X, i, Give(i, X, Z)) \rightarrow I_i(Give(i, X, Z))). \tag{3.15}$$

Both agents also have a similarly simplistic theory of planning (but again one which suffices for our example), again forming part of the theory of their belief units. In crude terms, when an agent believes that it has the intention of doing something and has a rule for achieving that intention, then the pre-conditions of the rule become new intentions. Recall that the $\rightarrow$ between the P_i and Q is not material implication.

Parsimony. If an agent believes that it does not intend something, it does not believe that it will intend the means to achieve it.

$$B : B_i(\neg I_i(Q)) \wedge B_i(P_1 \wedge \ldots \wedge P_j \wedge \ldots \wedge P_n \rightarrow Q) \rightarrow \neg B_i(I_i(P_j)). \tag{3.16}$$

Reduction. If there is only one way of achieving an intention, an agent adopts the intention of achieving its preconditions.

$$B : B_i(I_i(Q)) \wedge B_i(P_1 \wedge \ldots \wedge P_j \wedge \ldots \wedge P_n \rightarrow Q) \wedge \neg B_i(R_1 \wedge \ldots \wedge R_m \rightarrow Q) \rightarrow B_i(I_i(P_j)), \tag{3.17}$$

where $R_1 \wedge \ldots \wedge R_m$ is not a permutation of $P_1 \wedge \ldots \wedge P_n$.

Unique choice. If there are two or more ways of achieving an intention, only one is intended. Note that we use $\triangledown$ to denote exclusive or.

$$B : B_i(I_i(Q)) \wedge B_i(P_1 \wedge \ldots \wedge P_j \wedge \ldots \wedge P_n \rightarrow Q) \wedge B_i(R_1 \wedge \ldots \wedge R_m \rightarrow Q) \rightarrow B_i(I_i(P_1 \wedge \ldots \wedge P_n)) \triangledown B_i(I_i(R_1 \wedge \ldots \wedge R_m)) \tag{3.18}$$

where $R_1 \wedge \ldots \wedge R_m$ is not a permutation of $P_1 \wedge \ldots \wedge P_n$.

[5] As it stands this formula appears contradictory. This is because we have, for simplicity, ignored the treatment of time. Of course, the complete specification of this example (which is not our main focus) would need time to be handled. We could do this by including time as an additional argument to each predicate, in which case the unicity formula would read $B : B_i(Have(X, Z, t) \wedge Give(X, Y, Z, t) \rightarrow \neg Have(X, Z, t+1))$. Doing this would involve making the base logic for each unit 'time capable', for instance by using the system introduced by Vila [44].

As mentioned above, we acknowledge that both the theory of action and the theory of planning are rather naive. The interested reader is encouraged to substitute their own such theories if desired.

Thus we have specified the initial information the agents possess, (3.1–3.12), and provided limited theories of action, (3.13–3.15), and planning, (3.16–3.18), to enable the agent to operate. This completes the theories in the units of the two agents. Finally, we need to give both agents some domain dependent bridge rules to link inter-agent communication and the agent's internal states:

Request. When an agent (i) needs something (Z) from another agent (X), it asks for it:

$$I : I_i(Give(X, i, Z)) \Rightarrow C : Ask(i, X, Give(X, i, Z)). \tag{3.19}$$

Offer. When an agent (i) has the intention of offering something (Z) to another agent (X), it informs the recipient of this fact:

$$I : I_i(Give(i, X, Z)) \Rightarrow C : Tell(i, X, Give(i, X, Z)). \tag{3.20}$$

Trust. When an agent (i) is told of a belief of another agent (X), it accepts that belief:

$$C : Tell(X, i, B_X(\varphi)) \Rightarrow B : B_i(\varphi). \tag{3.21}$$

Awareness of intentions. Agents are aware of their intentions.

$$I : I_i(\alpha) \Rightarrow B : B_i(I_i(\alpha)). \tag{3.22}$$

$$I : \neg I_i(\alpha) \Rightarrow B : B_i(\neg I_i(\alpha)). \tag{3.23}$$

Impulsiveness. When an agent believes it has an intention, it adopts that intention.

$$B : B_i(I_i(\alpha)) \Rightarrow I : I_i(\alpha). \tag{3.24}$$

With these bridge rules, the specification of the two agents is complete. We have thus demonstrated how the multi-context approach can be used to specify BDI agents. In particular, we have defined two home improvement agents which we will return to in subsequent sections after we have discussed argumentation and its use in negotiation.

4 Agents and argumentation

The system of argumentation which we use as the basis for negotiation is based upon that proposed by Fox and colleagues [11, 22]. As with many systems of argumentation, it works by constructing series of logical steps (arguments) for and against propositions of interest and as such may be seen as an extension of classical logic. In classical logic, an argument is a sequence of inferences leading to a true conclusion. In the system of argumentation adopted here arguments not only prove whether propositions *are* true or false, but also suggest that propositions *might be* true or false. The strength of such a suggestion is ascertained by examining the propositions used in the relevant arguments. This form of argumentation may be seen as a formalization of work on informal logic and argumentation in philosophy [43], though it should be stressed that it was developed independently. It is summarized by the following schema:

$$\Gamma \vdash (\varphi, G),$$

where Γ is the set of formulae available for building arguments, $\vdash$ is a suitable consequence relation, φ is the proposition for which the argument is made, and G indicates the set of formulae used to infer φ, with $G \subseteq \Gamma$. The pair (φ, G) may also be extended to the triple (φ, G, α) to take account of the fact that φ may not be known to be true by giving it a degree of belief α [22].

The remainder of this section extends this system of argumentation to the multi-agent case and demonstrates how it can be used within the agent architecture introduced in Section 3. Again this is described first in a general setting in Section 4.1 and then, after a discussion of complexity issues (Section 4.2), in the setting of BDI agents (Section 4.3).

4.1 Multi-context multi-agent argumentation

We fit argumentation into our multi-context agents by building arguments using the rules of inference of the various units and the bridge rules between units. However, there is an important difference between the system of argumentation we employ and that used by other authors [8, 9, 24, 29]. This is as follows. Often the grounds of an argument are just the formulae from which the argument is built; it is taken for granted that the agent in question can build the necessary proof from the grounds when desired. However, this assumption does not necessarily hold in multi-agent systems. In particular, different agents may have different rules of inference within their units and different bridge rules between them. This means that there is no guarantee that other agents are able to reconstruct the proof for a formula from the formulae on which it is based. Hence, the grounds must contain complete proofs, including the rules of inference and the bridge rules employed, and we need to augment the notation for arguments to identify which rules of inference and which bridge rules are employed. We do this by exploiting the fact that rules of inference and bridge rules have a similar deductive behaviour and can be denoted in an identical way. We also need to identify the agent making the argument. We use

$$\Gamma \vdash_d \varphi$$

with $d = a_{\{r_1, \ldots, r_n\}}$, to mean that the formula φ is deduced by agent a from the set of formulae Γ by using the set of axioms, inference rules or bridge rules $\{r_1, \ldots, r_n\}$.[6] When there is no ambiguity the name of the agent will be omitted. The following are examples of the use of the notation to define deductive steps in agent a. In the first, the agent uses the 'request' bridge rule (3.19) to create a request from an intention, and in the second it applies an inference rule (mp stands for modus ponens) to two formulae in unit I:

$$\begin{aligned} \{I : I_a(Give(b, a, nail))\} &\vdash_{a_{\{3.19\}}} C : Ask(a, b, Give(b, a, nail)) \\ \{I : p, I : p \rightarrow q\} &\vdash_{a_{\{mp\}}} I : q. \end{aligned}$$

Making the rules of inference and bridge rules explicit means that they become part of the argument. This then makes it possible to build arguments about the applicability of such rules. As a result, agents which use different logics, and which therefore

[6] Here we give just the name of the axioms and rules. Strictly, however, we should give the axioms and rules themselves since agents will not necessarily use the same naming conventions.

use different rules of inference and bridge rules, are in principle able to engage in argumentation about which rules are valid. However, to do this in practice is complex since we need to find ways of representing the reasoning mechanism of other agents within individual agents so that each agent has a model of the ways in which its acquaintances reason. While it is one of our main lines of continuing research, we will say little more about it in this paper.

At this point we should also say a few words about the relationship between our description of arguments and the meta-theory of our agents. When we describe an argument we are making a statement in the meta-theory of the agent concerned since we are talking about what the agent may prove. Thus we could talk about arguments in general purely in terms of statements in the meta-theories of agents. However, we choose not to since we don't think that it adds anything to the explanation, and possibly even makes things less clear.

In the remainder of the paper we drop the 'B :', 'D :' and 'I :', once again to simplify the notation. With this in mind, we can define an argument in our framework:

DEFINITION 4.1
Given an agent a, an *argument* for a formula φ in the language of a is a pair (φ, P) where P is a set of grounds for φ.

It is the grounds of the argument which relate the formulae being deduced to the set of formulae it is deduced from.

DEFINITION 4.2
A set of *grounds* for φ in an agent a is an ordered set $\langle s_1, \ldots, s_n \rangle$ such that:

1. $s_n = \Gamma_n \vdash_{d_n} \varphi$;
2. every s_i, $i < n$, is either a formula in the theories of a, or $s_i = \Gamma_i \vdash_{d_i} \psi_i$; and
3. every p_j in every Γ_i is either a formula in the theories of agent a or ψ_k, $k < i$.

We call every s_i a *step* in the argument.

For the sake of readability, we will often refer to the conclusion of a deductive step with the identifier given to the step. Thus if we have an agent k which is equipped with propositional logic and the theory $\{a \wedge b\}$ then it would have an argument $(a, \langle \{a \wedge b\} \vdash_{k_{\{\wedge\text{-elimination}\}}} a \rangle)$. If, instead, k had the theory $\{a, a \rightarrow b, b \rightarrow c\}$, then it would have an argument $(c, \langle s_1, s_2 \rangle)$ where $s_1 = \{a, a \rightarrow b\} \vdash_{k_{\{mp\}}} b$, and $s_2 = \{b, b \rightarrow c\} \vdash_{k_{\{mp\}}} c$.

We distinguish tautological arguments, those arguments which do not rely on formulae from the agent's theories.

DEFINITION 4.3
An argument (φ, P) is *tautological* if all deductive steps in P are built using only rules of inference, bridge rules and axioms of the logics of the agent's units.

So, considering agent k again, the agent can build a tautological argument for any of the axioms and theorems of propositional logic. Thus the agent can build the argument $(a \rightarrow (a \vee b), \langle s_1 \rangle)$ where $s_1 = \{\} \vdash_{k_{\{A2\}}} a \rightarrow (a \vee b)$.[7] Clearly the notion of

[7] Where '*A2*' stands for the second axiom of the formulation of propositional logic given by Whitehead and Russell in their *Principia Mathematica* and restated by Hughes and Cresswell [17], which is $a \rightarrow (a \vee b)$.

a tautological argument will vary between agents when agents use different rules of inference and different bridge rules. Thus agents which use such different rules will differ in the way in which they classify arguments. The effects of this are, once again, out of the scope of this paper.

It is also helpful to distinguish consistent arguments (since we allow inconsistent ones even though we don't make use of them).

DEFINITION 4.4
We say that an argument (φ, P) is *consistent* if there are no $s_i, s_j \in P$ such that $s_i = \Gamma_i \vdash_{d_i} \psi$ and $s_j = \Gamma_j \vdash_{d_j} \neg\psi$. We also call such an argument *non-trivial*.

Now, because in argumentation a proof for a formula only suggests that the formula *may* be true (rather than indicating that it *is* true), we can have arguments for and against the same formula. In particular, given an argument for a formula, there are two interesting types of argument against it; arguments which rebut it and arguments which undercut it.

DEFINITION 4.5
An argument (φ_i, P_i) *rebuts* an argument (φ_j, P_j) if φ_i attacks φ_j.

Note that the notion of 'attack' is defined in Section 4.3; for the moment it is considered primitive, but can be thought of as meaning that the arguments disagree over the truth of φ_i and φ_j.

DEFINITION 4.6
An argument (φ_i, P_i) *undercuts* an argument (φ_j, P_j) if there exists $s_k \in P_j$ such that (1) s_k is a formula and φ_i attacks s_k, or (2) $s_k = \Gamma_k \vdash_{d_k} \psi$ and φ_i attacks ψ.

The reason that we don't define what we mean by 'attack' here is that it depends upon the logic in which arguments are built. Thus, in propositional logic, it makes sense for any formula φ to attack its negation $\neg\varphi$ and vice versa. However, in a modal logic of intention this no longer makes sense since such a notion of attack will not capture the clash between $I(p)$ and $I(\neg p)$. However, the ideas of undercutting and rebutting hold whatever the kind of attack. To illustrate them, let us revisit our friend k assuming it now has an expanded theory $\{a, a \rightarrow b, b \rightarrow c, a \rightarrow d, d \rightarrow \neg c, a \rightarrow e, e \rightarrow \neg b\}$. Now k can build three arguments:

$$(c, \langle \{a, a \rightarrow b\} \vdash_{k_{\{mp\}}} b, \{b, b \rightarrow c\} \vdash_{k_{\{mp\}}} c \rangle),$$
$$(\neg c, \langle \{a, a \rightarrow d\} \vdash_{k_{\{mp\}}} d, \{d, d \rightarrow \neg c\} \vdash_{k_{\{mp\}}} \neg c \rangle),$$
$$(\neg b, \langle \{a, a \rightarrow e\} \vdash_{k_{\{mp\}}} e, \{e, e \rightarrow \neg b\} \vdash_{k_{\{mp\}}} \neg b \rangle).$$

Since in propositional logic c and $\neg c$ attack one another and b and $\neg b$ attack one another, the second of these rebuts the first, while the third undercuts the first.

Relationships between arguments such as rebutting and undercutting have been widely studied, for instance by [8, 24, 29, 45]. The notions that we use here are broadly in line with the consensus on the issue. However, there is another form of conflict between arguments which stems from the inclusion of rules of inference and bridge rules in the argument. This is, as hinted at above, that one argument might attack the use of a rule used to build another argument. This form of attack is beyond the scope of this paper, so we will discuss it no further. It should be noted that, unlike some other authors, we do not present a universal definition of what it

means for one argument to attack another. We firmly believe that the form of attack depends upon the underlying language, and so, in our terms, will depend upon which units arguments are built in and what the units represent. We discuss notions of attack relevant to BDI agents in Section 4.3.

Our motivation for classifying arguments in terms of rebutting and undercutting is that it allows us to split arguments into classes of acceptability, again following [9] and our previous work on argumentation in multi-agent systems [28]. We have, in order of increasing acceptability:

A1 The class of all arguments that may be made from Γ.

A2 The class of all consistent arguments that may be made from Γ.

A3 The class of all arguments that may be made from Γ for propositions for which there are no rebutting arguments that may be made from Γ.

A4 The class of all arguments that may be made from Γ for propositions for which there are no undercutting arguments that may be made from Γ.

A5 The class of all tautological arguments that may be made from Γ.

Informally, the idea is that arguments in higher numbered classes are more acceptable because they are less questionable. Thus, if we have an argument for a proposition φ which is in class A4, and an argument for ψ which is in A2, then the better argument is that for φ. Since any argument from any class is included in all classes of lower acceptability, there is an order over the acceptability classes defined by set inclusion:

$$A_5(\Gamma) \subseteq A_4(\Gamma) \subseteq A_3(\Gamma) \subseteq A_2(\Gamma) \subseteq A_1(\Gamma).$$

Thus arguments in smaller classes are more acceptable than arguments in larger classes. Acceptability is important because it gives agents a way of deciding how to revise what they know (see Section 5). Clearly the acceptability class of an argument is local to an agent since it is depends upon the database from which the argument is built.

We should also point out that, even when handling contradictory arguments, the process of building arguments is monotonic. If we can build an argument for φ in standard propositional logic, then we can always build an argument for it, even if we are able to build an argument for $\neg\varphi$ later. However, the process of coming to conclusions using arguments is non-monotonic. If we have an argument for φ and no argument for $\neg\varphi$, then we conclude φ. If later we can build an argument for $\neg\varphi$ which is more acceptable than the argument for φ, then we change our conclusion to $\neg\varphi$.

4.2 Complexity analysis

The computational complexity of the argumentation process is clearly dependent upon the language in which the arguments are built. Furthermore, it is possible to state the construction of an argument for a formula φ from a set of formulae Γ as a satisfiability problem—is $\Gamma \cup \{\neg\varphi\}$ satisfiable? Thus the complexity of building arguments depends upon the complexity of satisfiability in the language in question. If the language is full first-order logic, then the problem of building an argument for φ is semi-decidable since satisfiability in first-order logic is semi-decidable. Similarly, if the language is

full propositional logic, the problem is decidable but NP-complete. However, if we restrict the language to propositional Horn clauses (which, with a finite language, can be a fully instantiated set of first-order Horn clauses) things are rather better. Indeed, the problem of building an argument is not only decidable but also may be achieved in time proportional to the number of propositions in the language [16]. The problem of building a rebutting argument is equivalent to building an argument for a proposition, so this is also decidable and takes time proportional to the size of the language. In the worst case, undercutting an argument involves attempting to rebut every step in the argument, and so is also decidable in time which is proportional to the product of the number of propositions in the language and the length of the argument in question.

4.3 Argumentation in BDI agents

To instantiate our argumentation model within the context of a particular agent architecture, like the one proposed in Section 3.2, we need to say exactly when two formulae attack one another. This is a rather more complex issue than is the case in single agent argumentation when two formulae attack one another if one is the negation of the other. In our BDI agents, the complication comes largely from the 'modalities' since there is no conflict between an agent which believes φ, that is $B_i(\varphi)$, and one which believes $\neg\varphi$, that is $B_j(\neg\varphi)$. Conflicts only occur when:

1. agents have opposite intentions (since then they actively intend to bring about incompatible results);
2. one agent intends to change a particular mental state in another agent; in other words the agent intends to persuade another agent to believe (or desire or intend) the negation of one of its current beliefs (respectively desires or intentions).

That is:

1. $I_i(\varphi)$ attacks $I_j(\neg\varphi)$. For example, 'Carles intends to be Prime Minister', $I_{Carles}(Prime(Carles))$, attacks 'Simon intends that Carles is not Prime Minister', $I_{Simon}(\neg Prime(Carles))$.
2. $I_i(M_j(\varphi))$ attacks $M_j(\neg\varphi)$ where M stands for any one of B, D, or I. For example, 'Kate intends that Simon believes that God exists', $I_{Kate}(B_{Simon}(God))$, attacks 'Simon believes that God does not exist', $B_{Simon}(\neg God)$.

In the first case Simon and Carles are in conflict about who should be Prime Minister. In the second case there is a conflict because Kate wants to change Simon's beliefs to a view that is the opposite of what he already believes. The second case can be generalized so that $I_i(M_{j_1}(M_{j_2}(\varphi)))$ attacks $M_{j_1}(\neg M_{j_2}(\varphi))$ and also attacks $M_{j_1}(M_{j_2}(\neg\varphi))$ where j_1 and j_2 are agent identifiers and the M_j are placeholders for any of B, D and I. Thus we get the following definition:

DEFINITION 4.7
Given agents i and j, we say that a formula φ_i of the language of agent i *attacks* a formula φ_j of the language of agent j if one of following cases hold:

1. $\varphi_i = I_i(\varphi)$ and $\varphi_j = I_j(\neg\varphi)$

2. $\varphi_i = I_i(M_{j_1}(M_{j_2}(\ldots M_{j_k}(\ldots M_{j_n}(\varphi)\ldots)\ldots)))$ and either
   (a) $\varphi_j = M_{j_1}(M_{j_2}(\ldots \neg M_{j_k}(\ldots M_{j_n}(\varphi)\ldots)\ldots))$ with $1 \leq k \leq n$, or
   (b) $\varphi_j = M_{j_1}(M_{j_2}(\ldots M_{j_k}(\ldots M_{j_n}(\neg\varphi)\ldots)\ldots))$.

With this notion of attack, our use of rebut, undercut and the acceptability classes is a natural extension of the use proposed by Elvang-Gøranssen *et al.* [9] to the multi-agent case. The difference is as follows. The notion of attack proposed by Elvang-Gøranssen *et al.* would recognise the conflict between $I_a(\varphi)$ and $\neg I_a(\varphi)$ (which in our approach would be inconsistency), but would not identify the conflict between $I_a(\varphi)$ and $I_b(\neg\varphi)$. Our extension, by virtue of the fact that it looks inside the modalities, is able to detect this latter type of attack. This is important because it is the latter form of attack that figures most prominently in interactions between agents. Because it does not seem as important in the interaction between agents, at the moment we have nothing much to say about the handling of inconsistency within our multi-context agents. However, it might well be the case that an agent will have to deal with contradictory beliefs $B_a(\varphi)$ and $\neg B_a(\varphi)$, and if it becomes necessary to handle such situations, it seems likely that we can make use of the argument-based approaches to dealing with inconsistency which already exist.

5 Negotiation as argumentation

The next point to address is how negotiation by argumentation proceeds, considering, for simplicity, just the two-agent case.[8]

5.1 Argumentation and the negotiation process

The first step is the selection by agent a of an intention to be satisfied, $I_a(\varphi)$. Agent a may first try to find a proof for it based on its own resources. If this is not possible, then the use of an external resource is necessary and a negotiation with the owner of the resource is started. Let's assume the owner is called b. In this latter case, agent a builds an argument (ψ_a, P_a), where ψ_a is a proposal containing the requirement for the resource to be transferred, and then passes it to agent b.[9] Having received this argument, agent b then examines (ψ_a, P_a) to see if it agrees with the suggestion. The simplest case is when agent b can find no reason to disagree with the suggestion, and so simply responds with a message to indicate its agreement.[10] More interesting cases occur when agent b does not agree with the suggestion, and there are two types of situation in which this may happen.

The first situation is that in which the suggestion directly conflicts with b's objectives. This state of affairs is detected when b can build an argument (ψ_b, P_b) such

[8] It should be stressed that the limitation to the two-agent case is purely pragmatic in that it makes the description easier. There is no obvious reason why the procedure described here cannot be extended to an arbitrarily large number of agents.

[9] Of course, an agent need not pass the grounds for its requirement to other agents if this would not be in its interests, but if it does, negotiation is likely to be completed more quickly (as discussed in Section 2). For the purposes of this paper we assume that the '*Ask*' is always passed with the argument for the formula in question.

[10] Thus, for the purposes of this paper, we assume that an agent accepts a proposal unless it can build an argument against it.

that ψ_b attacks ψ_a. In other words, this kind of conflict occurs when b can build an argument that rebuts the initial suggestion (Definitions 4.5 and 4.7 case 1). The second kind of conflict occurs when agent b does not reject the suggestion made by a, but one of the steps by which the suggestion is reached. In other words, b can build an undercutting argument (ψ'_b, P'_b) for (ψ_a, P_a) (Definition 4.6). This may occur because ψ_a conflicts with one of agent b's intentions (Definition 4.7 case 1), or because in constructing the suggestion, agent a made an incorrect assumption about one of b's beliefs, desires or intentions (Definition 4.7 case 2). In either case b informs a of its objection by sending back its attacking argument.

Whatever the form of attack, the agents can reach agreement so long as a can either find an alternative way of achieving its original objective, or a way of persuading b to drop its objection. If either type of argument can be found, a will submit it to b. If agent b can find no reason to reject the new suggestion, it will be accepted and the negotiation will end. Otherwise the process may be iterated (see Figure 1).

Considering this kind of negotiation process, it is clear that it falls within the framework suggested in Section 2. Firstly, it provides a means of generating proposals by constructing arguments for an agent's intentions. This construction process also has the effect of generating explanations, in the form of the grounds of these arguments, which can be passed to other agents if desired. Once the proposal is made, it is evaluated by other agents which attempt to build arguments against it. Any such arguments are critiques. Attempting to build arguments against the proposal also gives a means of generating counter-proposals. Furthermore, when the grounds of arguments are passed between agents they serve as a guide to acceptable solutions and so act as meta-information, and one can imagine agents passing parts of the grounds for their critiques or counter-proposals in isolation as additional meta-information.

The acceptability classes are necessary for two reasons. Firstly, they are the means that an agent uses to determine how strongly it objects to proposals. If, when evaluating a proposal, the agent discovers the proposal falls into classes A4 or A5, then it is accepted. If the proposal falls into class A2 or A3, then it is a suggestion that might be accommodated by finding an alternative way of achieving the initial intention. If the proposal falls into class A1 then there is something seriously wrong with it, and a completely new proposal is indicated. The second use of acceptability classes is to evaluate proposals internally before sending them as suggestions. Clearly it is sensible for an agent to vet its proposals to ensure that they are not detrimental to it, and the acceptability class mechanism provides a way of rating possible suggestions to ensure that only the best is sent.

5.2 An example of two negotiating agents

Using the home improvement agents specified earlier, we illustrate the ideas introduced in Sections 4 and 5.1.

Step 1: Agent a tries to find a proof for $Can(a, hang(picture))$ because of its intention $I_a(Can(a, hang(picture)))$. The most useful[11] proof it can build is based

[11] Note that because of the fact that in a strong realist agent any intention is also a belief, a can build a proof based only on $I_a(Can(a, hang(picture)))$ by application of the strong realist bridge rules. However, this proof gives no indication of how the intention can be achieved and so is not as

on $B_a(Have(a, nail))$, which in turn, by the theory of planning, makes the belief $B_a(I_a(Give(b, a, nail)))$ true. This is transformed, by means of bridge rule 3.24, into $I_a(Give(b, a, nail))$. More formally, agent a builds an argument

$$(I_a(Give(b, a, nail)), P_a)$$

where P_a is[12]

$$\begin{aligned}
\{I_a(Can(a, hang(picture)))\} &\vdash_{3.22} B_a(I_a(Can(a, hang(picture)))) && (5.1)\\
\{(5.1), (3.17), (3.7)\} &\vdash_{mp} B_a(I_a(Have(a, nail))) && (5.2)\\
\{(3.6), (3.13)\} &\vdash_{mp} B_a(Give(b, Y, nail) \rightarrow Have(Y, nail)) && (5.3)\\
\{(5.3), (5.2), (3.17)\} &\vdash_{mp} B_a(I_a(Give(b, a, nail))) && (5.4)\\
\{(5.4)\} &\vdash_{3.24} I_a(Give(b, a, nail)). && (5.5)
\end{aligned}$$

This is then converted into an action using bridge rule 3.19:

$$\{(5.5)\} \vdash_{3.19} Ask(a, b, Give(b, a, nail)).$$

When agent a generates the argument $(I_a(Give(b, a, nail)), P_a)$ it is placed in acceptability class A_4 since a cannot build any undercutting arguments against it and so a deems it to be a suitable suggestion to be passed to b.

Step 2: Unit C of agent b receives the formula $Ask(a, b, Give(b, a, nail))$, which, as specified, brings with it the argument:

$$(I_a(Give(b, a, nail)), \langle (5.1), (5.2), (5.3), (5.4), (5.5) \rangle).$$

Now, agent b has its own goal, $I_b(Can(b, hang(mirror)))$, which as we shall see forms the basis of its argument:

$$(I_b(\neg Give(b, a, nail)), P_b)$$

where P_b:

$$\begin{aligned}
\{I_b(Can(b, hang(mirror)))\} &\vdash_{3.22} B_b(I_b(Can(b, hang(mirror)))) && (5.6)\\
\{(5.6), (3.12), (3.17)\} &\vdash_{mp} B_b(I_b(Have(b, nail))) && (5.7)\\
\{(5.7), (3.14)\} &\vdash_{mt} B_b(I_b(\neg Give(b, Y, nail))) && (5.8)\\
\{(5.8)\} &\vdash_{pt} B_b(I_b(\neg Give(b, a, nail))). && (5.9)
\end{aligned}$$

This argument rebuts the argument for $I_a(Give(b, a, nail))$. This means that for agent b both arguments are in class A_2 (since they mutually rebut one another but they are consistent). Assuming the agents are rational, and given that both arguments are in the same class, b will probably prefer (by some utility analysis) the second argument since this enables it to satisfy one of its intentions (adherence to the argument

useful as the proof we detail.

[12] In what follows, 'mp' stands for *modus ponens*, 'mt' stands for *modus tollens* and 'pt' stands for *particularization*. Because of space limitations, we omit the axioms of the unit in which the deduction is made. Recall that we use equation numbers to refer to the conclusion of a step rather than the step itself (thus (5.1) stands for $B_a(I_a(Can(a, hang(picture))))$).

proposed by a would clobber its intention of hanging the mirror). According to our negotiation model (Section 2), b will return the second argument to a as a critique.

Step 3: When agent a receives the argument from b it classifies both its original argument and the incoming argument as class A_2 since they are both rebutted (by each other). Thus its original argument moves from A_4 to A_2. In response, agent a generates a new argument which provides an alternative way of hanging the mirror that will satisfy b's goal without using the nail:

$$(B_a(\neg I_b(Have(b, nail))), P'_a)$$

where P'_a is[13]

$$\{\neg I_a(Can(a, hang(mirror)))\} \vdash_{3\,23} B_a(\neg I_a(Can(a, hang(mirror)))) \quad (5.10)$$

$$\{(5.10), (3.16), (3.8)\} \vdash_{mp} \neg B_a(I_a(Have(a, screw))) \wedge \neg B_a(I_a(Have(a, screwdriver))) \quad (5.11)$$

$$\{(5.11)\} \vdash_{sr} \neg I_a(I_a(Have(a, screwdriver))) \wedge \neg I_a(I_a(Have(a, screw))) \quad (5.12)$$

$$\{(5.11), (3.3), (3.5), (3.15)\} \vdash_{mp,pt} B_a(Ask(b, a, Give(a, b, screw)) \rightarrow I_a(Give(a, b, screw))) \wedge B_a(Ask(b, a, Give(a, b, screwdriver)) \rightarrow I_a(Give(a, b, screwdriver))) \quad (5.13)$$

$$\{(5.13), (3.8)\} \vdash_{mp} B_a(Ask(b, a, Give(a, b, screw)) \wedge Ask(b, a, Give(a, b, screwdriver)) \rightarrow Can(b, hang(mirror))) \quad (5.14)$$

$$\{(3.18), (5.14), (3.12)\} \vdash_{mp} B_a(\neg I_b(Have(b, nail))). \quad (5.15)$$

This argument is classified in A_4 since a can neither rebut nor undercut it. Agent a then sends this latest argument to b as a counter-proposal. Agent b cannot build any arguments which attack this new argument and so it is classified as being in A_4. Given the strength of the new argument, b accepts it. Here the crucial point is that b cannot construct a rebuttal for the new argument as a subargument of its previous argument because it can no longer use the reduction planning rule (3.17). This is because b has now acquired a new rule for hanging mirrors (as part of the new argument), and, because of the 'Trust' bridge rule (3.21), has added this rule to its set of beliefs. Moreover, the second argument can no longer be maintained for the same reason, so a's original argument is reclassified as being in A_4. Hence agent a will receive the nail, agent b will ask for the screw and the screwdriver and both will reach their goals.

Note that step 5.12 is crucial in the construction of the undercutting argument. This step depends upon the fact that agent a has the bridge rules associated with strong realism and so can go from $\neg B_a(I_a(Have(a, screw)))$ to $\neg D_a(I_a(Have(a, screw)))$ and hence to $\neg I_a(I_a(Have(a, screw)))$. If the agent did not have these bridge rules (e.g. it had those of realism or weak realism) a would not have been able to come up

[13] 'sr' stands for the set of bridge rules associated with *strong realism*.

TABLE 1. Walton and Krabbe's classification of dialogues

Type of dialogue	Initial situation	Participant's goal	Goal of dialogue
Persuasion	Conflict of opinions	Persuade other party	Resolve or clarify issue
Inquiry	Need to have proof	Find and verify evidence	Prove (disprove) hypothesis
Negotiation	Conflict of interests	Get what you most want	Reasonable settlement
Information seeking	One party lacks information	Acquire or give information	Exchange information
Deliberation	Dilemma or practical choice	Co-ordinate goals or actions	Decide best course of action

with its final suggestion. This gives some hint of the flexibility of our approach and shows that changing some basic assumptions about the relations between the units makes a substantial difference to the behaviour of the agents.

5.3 Other views of dialogue

This section has shown how agents built using our multi-context approach can use argumentation as a means of negotiating in the sense described in Section 2. As mentioned there, this is a particular view of what it means to negotiate—a view which is common within the field of multi-agent systems. However, this is a broad view, and there is merit in considering other ways of classifying the kind of dialogue which we have demonstrated using our approach. A suitable framework for performing this kind of classification is that provided by Walton and Krabbe [46]. Walton and Krabbe distinguish six basic types of dialogue based on the situation at the start of the dialogue, the goals of the dialogue itself, and the goals of the participants in the dialogue. The results of their deliberations are summarised in Table 1.

Using this classification, we can see that the example dialogue we have presented includes elements of persuasion, negotiation, information seeking, and deliberation. Taking a high level view of what is going on, namely that two agents are trying to decide on a common course of action, the dialogue is what Walton and Krabbe term a 'deliberation'. However, because the initial proposal identifies a conflict of interests, in particular about the use of the limited resources available to the agents, the agents also engage in a 'negotiation'. This negotiation proceeds by 'persuasion', in which the agents clarify the situation, and the persuasion is achieved by means of 'information

seeking', where the agents share information (in the grounds of the arguments they exchange). Thus one view of what we are proposing is that it is a general mechanism for inter-agent dialogue which allows the agents to shift seamlessly between the various types of dialogue identified by Walton and Krabbe. For completeness, we should point out that one can also cast the high level view of the agent interaction as being an 'inquiry' since the agents are attempting to assemble a proof that they are able to achieve their goals using the resources at their disposal.

The idea that argumentation provides an overarching framework for different types of dialogue also has echoes in the work of Loui and Moore [25]. They argue that the game theoretic account of negotiation fails to take account of a number of important aspects of the negotiation process, from our point of view most notably: arguing for a proposal, informing, reporting overlooked possibilities, and pursuing sub-dialogues. They then argue that all these aspects can be captured by a model of negotiation which draws upon ideas from artificial intelligence. Thus our work and that of Loui and Moore to some extent mutually support one another. Loui and Moore provide an eloquent justification for models of negotiation such as ours to be taken seriously, and we provide a concrete example of just the kind of model that they argue in favour of (though our model would need to be extended in order to capture all the kinds of dialogue that they deal with).

6 Related work

This paper has dealt with a number of topics from various research areas—including argumentation-based reasoning, formal models of agent architectures, multi-context systems, and multi-agent negotiation. Therefore a complete review of all related literature is not possible. Instead we make passing comments on the first three areas and concentrate more fully on the final one because it is closest to our focus in this paper. Traditionally work on argumentation-based reasoning has concentrated on the operation of a single agent which argues with itself in order to establish its beliefs [8, 9, 24, 29]. As indicated and discussed in Section 4, this basic approach and framework needed to be extended to account for the multi-agent case in which several traditional assumptions do not hold. Previous work which has produced formal models of agent architectures, for example dMARS [18], Agent0 [37] and GRATE* [19], has failed to carry forward the clarity of the specification into the implementation—there is a leap of faith required between the two. Our work, on the other hand, maintains a clear link between specification and implementation through the direct execution of the specification developed in our running example. There are also differences between our work and previous work on using multi-context systems to model agents' beliefs. In the latter [14], different units, all containing a belief predicate, are used to represent the beliefs of the agent and the beliefs of all the acquaintances of the agent. The nested beliefs of agents may lead to tree-like structures of such units (called *belief contexts*). Such structures have then been used to solve problems like the three wise men [5]. In our case, however, any nested beliefs are included in a single unit and we provide a more comprehensive formalization of an autonomous agent in that we additionally show how other attitudes can be incorporated into the architecture.

In terms of automated negotiation and argumentation there are a number of related items of research. Bussmann and Müller [3] draw upon social psychology to devise

a negotiation model and algorithm that can be employed by agents in a cooperative environment. Their model is much richer than those found in traditional multi-agent systems (see [23] for a review) and accords well with our generic model. However, it lacks a rigorous theoretical underpinning and it assumes that agents are inherently cooperative. Lâasri *et al.* [23] present a similarly rich negotiation model, although drawn from a predominantly multi-agent systems background, but again make the limiting assumption of cooperating agents.

Rosenschein and Zlotkin's research [36] is representative of a growing body of work on negotiation which is based on game theory. This work does not make the cooperating agent assumption; indeed agents are regarded as self-interested utility maximizers. Despite producing some important results, including some related to deceit and lying in negotiation, their work embodies a number of limiting assumptions. The main concerns are that the agents are assumed to have complete knowledge of the payoff matrix, and hence of the other agents' preferences, and also that precise utility values can be provided. Our approach inherently assumes a partial information perspective and is more qualitative in nature.

Sycara's work on the Persuader system [41] employs argumentation as part of a system that operates in the domain of labour negotiations. Although demonstrating the power and elegance of the approach, her system has a centralized arbitrator to handle the disagreements and is thus less general than ours. This work led to subsequent research by Kraus *et al.* [21] into providing a logical model of the process of argumentation. Their approach involves defining a new logic to define the agent's properties and then identifying five different types of argument that can be used in conjunction with their model (threats, rewards, appeals to precedent, appeals to prevailing practice, and appeals to self-interest). Our approach differs in that we adopt a system of argumentation as our start point and put in place the basic infrastructure for using argumentation as the negotiation metaphor. Their five types of argument, and many others besides, could be implemented in our system simply by instantiating different behavioural rules within the individual agents [39].

The final piece of related work we will discuss is perhaps the most closely related. Thomé [42] has proposed a model of negotiation in which (i) agents negotiate by exchanging arguments, (ii) the generation of arguments is guided by looking at the relationship between the arguments, and (iii) these relationships are defined in terms of which arguments attack and defeat which other arguments. Furthermore, the model is intended to allow agents to converge on a solution to resource allocation problems by giving them an ever more complete view of the real state of the world (which, as in our model, is assumed to be incomplete initially). Thus Thomé's model clearly has a lot in common with ours. However, there are significant differences, most of which result from the rather different perspectives we have on the problem. Our model starts from a clear picture of what an agent looks like, and grounds the system of argumentation we propose in that, tying the relationship between arguments to the mental states of the agents. Thomé on the other hand starts from outside the agents, giving a more abstract view of argumentation which allows him to define the relationship between arguments in terms of how likely they are to lead to agreement. This in turn means that he is able to ensure that the negotiation procedure he suggests will lead to agreement, since at each step agents make proposals which are more likely to be accepted than those made previously.

7 Conclusions and future work

This paper has presented a formal model of argumentation-based reasoning and negotiation for autonomous agents. The model indicates how agents capable of flexible and sophisticated argumentation can be specified both in general terms and in terms of a particular type of agent (namely a BDI agent). We have shown how agents can construct arguments to justify their proposals, how agents can critique proposals and how agents can exchange arguments to help guide their problem solving behaviour towards mutually acceptable solutions. There are three important benefits in terms of the practical implementation of our agents which follow from using the multi-context approach [6]. First, the modular organization of the architecture's components (in our case the BDI modalities) in different units reduces the complexity of the theorem proving mechanism. Second, it is easier to define proof strategies as combinations of the simple deductive elements in the system (local reasoning in the units and the application of bridge rules) than it is to have a monolithic, all encompassing approach. Third, we are able to show a clear link to potential implementations of agents which negotiate and reason in the manner we have advocated. This link can be achieved by implementing the various units as concurrent theorem provers with connections between them as specified by the bridge rules.

We see this work as being an important step in our overall aim of building agents which negotiate. In particular, we see it as a necessary extension of work detailed in [39]. In that paper we described a negotiation protocol which allows for the exchange of complex proposals and a language for expressing such proposals, and suggested that agents would build proposals that included compelling arguments for why the proposal should be adopted. This paper backs up the suggestion by indicating how argumentation can be used to construct proposals, create critiques, provide explanations and meta-information, and how an exchange of arguments may be used to guide two agents to agreement on some topic.

A number of issues raised in this paper require further investigation. Most prominent amongst these is the need to produce an implementation which supports both the generic definition of agent architectures and the specific instantiations for particular types of agent. Secondly, the notion of attacking inference steps, as discussed in Section 4.1, needs to be more fully elaborated to both ascertain whether it is useful for negotiating agents and whether it can be achieved in a tractable manner. Thirdly, the means by which agents generate and rate arguments needs to be expanded. Acceptability classes provide a means of ordering arguments, but it is likely that we will require the ability to provide a more fine-grained ranking (see step 2 of the example in Section 5). Thus agents need detailed strategies and tactics, based on models of their acquaintances and records of past encounters, to make more refined choices about the quality of the arguments they are presented with. Finally, agents need effective internal mechanisms for tracking and maintaining their arguments and propagating changes in their preferences as their knowledge changes over time (as illustrated in step 3 of the example in Section 5).

Acknowledgements

The authors would like to thank the anonymous referees for their perceptive comments on earlier versions of this paper. The second author was partly supported by Spanish MEC grant PR95-313 and the Spanish CICYT project SMASH, TIC96-1038-C04001. Carles Sierra is on sabbatical leave from Artificial Intelligence Research Institute—IIIA, CSIC, Campus UAB, 08193 Bellaterra, Barcelona, Spain.

References

[1] G. Attardi and M. Simi. A formalisation of viewpoints. *Fundamenta Informaticae*, **23**, 149–174, 1995.

[2] R. A. Brooks. Intelligence without reason. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pp. 569–595, 1991.

[3] S. Bussmann and H. J. Müller. A negotiation framework for cooperating agents. In *Proceedings of the CKBS-SIG Conference*, pp. 1–17, 1992.

[4] B. F. Chellas. *Modal Logic: An Introduction*. Cambridge University Press, Cambridge, 1980.

[5] A. Cimatti and L. Serafini. Multi-agent reasoning with belief contexts: The approach and a case study. In *Proceedings of the 3rd International Workshop on Agent Theories, Architectures and Languages*, pp. 62–73, 1994.

[6] A. Cimatti and L. Serafini. Multi-agent reasoning with belief contexts III: Towards the mechanization. In *Proceedings of the IJCAI Workshop on Modelling Context in Knowledge Representation and Reasoning in Artificial Intelligence*, pp. 35–45, 1995.

[7] P. R. Cohen and H. J. Levesque. Intention is choice with commitment. *Artificial Intelligence*, **42**, 213–261, 1990.

[8] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, **7**, 321–357, 1995.

[9] M. Elvang-Gøransson, P. Krause and J. Fox. Dialectic reasoning with inconsistent information. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pp. 114–121, 1993.

[10] E. A. Emerson. Temporal and Modal Logic. In *Handbook of Theoretical Computer Science*, J van Leeuwen, ed. pp. 996–1071. Elsevier, 1990.

[11] J. Fox, P. Krause and S. Ambler. Arguments, contradictions and practical reasoning. In *Proceedings of the 10th European Conference on Artificial Intelligence*, pp. 623–627, 1992.

[12] M. P. Georgeff and A. L. Lansky. Reactive reasoning and planning. In *Proceedings of the 6th National Conference on Artificial Intelligence*, pp. 677–682, 1987.

[13] F. Giunchiglia and L. Serafini. Multilanguage hierarchical logics (or: How we can do without modal logics). *Artificial Intelligence*, **65**, 29–70, 1994.

[14] F. Guinchiglia. Contextual reasoning. In *Proceedings of the IJCAI Workshop on Using Knowledge in Context*, 1993.

[15] C. Hewitt. Open information systems semantics for distributed artificial intelligence. *Artificial Intelligence*, **47**, 79–106, 1991.

[16] W. Hodges. Logical features of Horn clauses. In *Logical Foundations, Handbook of Logic in Artificial Intelligence and Logic Programming*, Volume 1, J. van Leeuwen, ed. pp. 449–503. Oxford University Press, 1993.

[17] G. E. Hughes and M. J. Cresswell. *An Introduction to Modal Logic*. Methuen, London, 1968.

[18] F. F. Ingrand, M. P. Georgeff and A. S. Rao. An architecture for real-time reasoning and system control. *IEEE Expert*, **7**, 34–44, 1992.

[19] N. R. Jennings. Controlling cooperative problem solving in industrial multi-agent systems using joint intentions. *Artificial Intelligence*, **75**, 195–240, 1995.

[20] N. R. Jennings, E. H. Mamdani, J. Corera, I. Laresgoiti, F. Perriolat, P. Skarek and L. Z. Varga. Using ARCHON to develop real-word DAI applications Part 1. *IEEE Expert*, **11**, 64–70, 1996.

[21] S. Kraus, M. Nirkhe and K. Sycara. Reaching agreements through argumentation: a logical model (preliminary report). In *Proceedings of the Workshop on Distributed Artificial Intelligence*, 1993.

[22] P. Krause, S. Ambler, M. Elvang-Gøransson and J. Fox. A logic of argumentation for reasoning under uncertainty. *Computational Intelligence*, **11**, 113–131, 1995.

[23] B. Lâasri, H. Lâasri, S. Lander and V. Lesser. A generic model for intelligent negotiating agents. *International Journal of Intelligent and Cooperative Information Systems*, **1**, 291–317, 1992.

[24] R. Loui. Defeat among arguments: a system of defeasible inference. *Computational Intelligence*, **3**, 100–106, 1987.

[25] R. Loui and D. Moore. Dialogue and deliberation. *Negotiation Journal*, (submitted).

[26] P. Maes. Agents that reduce work and information overload. *Communications of the ACM*, **37**, 31–40, 1996.

[27] P. Noriega and C. Sierra. Towards layered dialogical agents. In *Proceedings of the 3rd International Workshop on Agents Theories, Architectures and Languages*, pp. 157–171, 1996.

[28] 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.

[29] J. L. Pollock. Justification and defeat. *Artificial Intelligence*, **67**, 377–407, 1994.

[30] D. G. Pruitt. *Negotiation Behaviour*. Academic Press, London, 1981.

[31] A. Rao and M. Georgeff. Asymmetry thesis and side-effect problems in linear time and branching time intention logics. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*, pp. 498–504, 1991.

[32] A. Rao and M. Georgeff. BDI agents: From theory to practice. In *Proceedings of the 1st International Conference on Multi-Agent Systems*, pp. 312–319, 1995.

[33] A. S. Rao and M. P. Georgeff. Modeling Rational Agents within a BDI-Architecture. In *Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning*, pp. 473–484, 1991.

[34] A. S. Rao and M. P. Georgeff. Formal Models and Decision Procedures for Multi-Agent Systems. Technical Note 61, Australian Artificial Intelligence Institute, 1995.

[35] J. A. Rodríguez, P. Noriega, C. Sierra and J. Padget. A Java-based electronic auction house. In *Proceedings of the 2nd International Conference on The Practical Application of Intelligent Agents and Multi-Agent Technology*, pp. 207–224, 1997.

[36] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter*. The MIT Press, Cambridge, MA, 1994.

[37] Y. Shoham. Agent-oriented programming. *Artificial Intelligence*, **60**, 51–92, 1993.

[38] C. Sierra, P. Faratin and N. R. Jennings. A service-oriented negotiation model between autonomous agents. In *Proceedings of the 8th European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, pp. 17–35, 1997.

[39] 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.

[40] I. Sommerville. *Software Engineering*. Addison Wesley, Wokingham, 1992.

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

[42] F. Thomé. 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.

[43] F. H. van Eemeren, R. Grootendorst, F. S. Henkemans, J. A. Blair, R. H. Johnson, E. C. W. Krabbe, C. Plantin, D. N. Walton, C. A. Willard, J. Woods and D. Zarefsky. *Fundamentals of Argumentation Theory: A Handbook of Historical Backgrounds and Contemporary Developments*. Lawrence Erlbaum Associates, Mahwah, NJ, 1996.

[44] L. Vila. On temporal representation and reasoning in knowledge-based systems. IIIA Monographies, Barcelona, Spain, 1994.

[45] G. Vreeswijk. The feasibility of defeat in defeasible reasoning. In *Proceedings of the 1st International Conference on Knowledge Representation and Reasoning*, pp. 526–534, 1989.

[46] D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of New York Press, Albany, NY, 1995.

[47] M. P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, **1**, 1–23, 1993.

[48] M. J. Wooldridge and N. R. Jennings. Intelligent agents: Theory and practice. *The Knowledge Engineering Review*, **10**, 115–152, 1995.

Received 31 July 1997