Reaching agreements through argumentation: a logical model and implementation ⋆

Sarit Kraus

Department of Mathematics and Computer Science
Bar-Ilan University 52-900 Ramat-Gan, Israel
e-mail: sarit@cs.biu.ac.il and
Institute for Advanced Computer Studies,
University of Maryland, College Park MD 20742

Katia Sycara

School of Computer Science
Carnegie Mellon University Pittsburgh, PA 15213
e-mail: katia@cs.cmu.edu

Amir Evenchik

Motorola Israel Ltd. 3 Kremenetski St.
Tel-Aviv 67899 Israel e-mail: evenchik@cig.mot.com

Abstract

In a multi-agent environment, where self-motivated agents try to pursue their own goals, cooperation cannot be taken for granted. Cooperation must be planned for and achieved through communication and *negotiation*. We present a logical model of the mental states of the agents based on a representation of their beliefs, desires, intentions, and goals. We present *argumentation* as an iterative process emerging from exchanges among agents to persuade each other and bring about a change in intentions. We look at argumentation as a mechanism for achieving cooperation and agreements. Using categories identified from human multi-agent negotiation, we demonstrate how the logic can be used to specify argument formulation and evaluation. We also illustrate how the developed logic can be used to describe different types of agents.

Furthermore, we present a general Automated Negotiation Agent which we implemented, based on the logical model. Using this system, a user can analyze and explore different methods to negotiate and argue in a non-cooperative environment where no centralized mechanism for coordination exists. The development of negotiating agents in the framework of the Automated Negotiation Agent is illustrated with an example where the agents plan, act, and resolve conflicts via negotiation in a Blocks World environment.

Preprint submitted to Elsevier Preprint *7 June 1998*

1 Introduction

In a multi-agent environment, where self-motivated agents try to pursue their own goals, cooperation cannot be taken for granted. Cooperation must be planned for and achieved through communication and *negotiation*. Negotiation often involves *argumentation* in the form of an exchange of messages or a dialogue. Arguments are utterances whose aim is to change the intentions (and consequently the actions) of the listener. Within the context of negotiating self-interested agents, this change of intentions could make agents more cooperative. There are different arguments that could be used by one agent to change the intentions of another. Irrespective of what argument is used, the recipient agent must evaluate the argument and decide whether or not to change its intentions and actions.

For example, imagine two mobile robots on Mars, each built to maximize its own utility. R_1 requests R_2 to dig for a certain mineral. R_2 refuses. R_1 responds with a threat: "If you do not dig for me, I will break your antenna". R_2 is faced with the task of evaluating this threat. Several considerations must be taken into account, such as whether or not the threat is bounded, what R_1's credibility is, how important it is for R_2 to have its antenna intact, so on and so forth. R_1 may take a different approach if R_2 refuses to dig, and respond with a promise for a reward: "If you dig for me today, I will help you move your equipment tomorrow." Here, R_2 needs to evaluate the promise of future reward.

Argumentation is essential to bringing about agreement in non-cooperative situations when agents have incomplete knowledge about each other or the environment. In such situations, agents impart information to each other via the exchanged messages. Argumentation may also be called for when agents either do not have the ability or the time to make inferences. This is the case when agents have bounded inference systems which either may not be complete or may not be closed under inferences [71,105].

In order to negotiate effectively, an agent needs the ability to (a) represent and

⋆ This material is based upon work supported in part by NSF Grant No. IRI-9423967, NSF grant No. IRI 9724937, NSF Grant No. IRI-9612131, NSF Grant No. IRI-9712607, ONR Grant No. N00014-96-1-1222 and Army Research Lab under contract number DAAL0197K0135. We would like to thank Madhura Nirkhe for her contribution to the development of the formal model and to Ariel Stollman for the help in the implementation.

maintain a model of its own beliefs, desires, goals, and intentions, (b) reason with other agents' beliefs, desires, goals, and intentions, and (c) influence other agents' beliefs, intentions, and behavior. When agents are non-collaborative, the process of argumentation is an iterative exchange of proposals towards reducing conflict and promoting the achievement of the individual goals of the agents.

Arguments are used by a persuader as a means to dynamically change the preferences, intentions, and actions of a persuadee, to increase the willingness of the persuadee to cooperate. Over repeated encounters, agents may analyze each other's patterns of behavior to establish an analogue to the human notions of credibility and reputation. This may influence the evaluation of arguments, as we will see in scenarios such as the "threats" described later. By observing the reactions to the arguments, the sending agent can update and correct its model of the recipient agent, thus refining its planning and argumentation knowledge.

In this paper we develop a formal logic that forms a basis for the development of a formal axiomatization system for argumentation. We offer a logical model of the mental states of the agents based on a representation of their beliefs, desires, intentions and goals. We present argumentation as an iterative process of exchanges among agents to persuade each other and bring about a change in intentions. Our work on the formal mental model overlaps with the work of others who have developed formal models for communicative agents (e.g., [16,164,138,69,129,115,64,19,86,133]) and for mental models of agents (e.g., [75,162,152]). We will discuss related work in Section 5 and point out the differences of our work with respect to that of others. The main difference from previous work is that we have developed our formalization from the argumentation point of view. We present a set of axioms that allows the agents to automatically generate and evaluate arguments in a multi-agent environment.

Based on our formalization, we have developed and implemented a general Automated Negotiation Agent (ANA) which acts and negotiates in a simulated multi-agent environment. In the simulation system, several ANA agents can be defined and created. Each of these agents is assigned an initial set of mental states and inference rules which guide it in every step and decision that it takes (goal seeking, argument generation and selection, request evaluation, and so on). Once created, the agent will try to accomplish its desires, using arguments, if needed.

Both the mental states and the different inference rules are based on our formal logic model. Each of the agents changes its mental states according to a rule which applies at the time of the change. The ability to define mental states for each of the agents, as well as to define different inference rules for argument generation, allows the user of the system to test different argument types and to assess their impact on the effectiveness of the agent's negotiation capability. This also allows the user to evaluate ways of selecting the most appropriate argument at any stage of the negotiation. These capabilities are illustrated through an extensive example, where agents negotiate in a Blocks World environment.

The paper is organized as follows. Section 2 presents the logical argumentation formalism and the various agent types which might be engaged in argumentation. Section 3 describes the various argument types we have identified and how an agent can evaluate argument appropriateness. Section 4 discusses the general Automated Negotiation Agent (ANA) and its capabilities for argument generation and evaluation, based on the logical argumentation axiomatization. Section 5 situates our work within the related literature. Section 6 presents concluding remarks.

2 The Mental Model

We have a set of agents, not necessarily cooperative, with the ability to exchange messages. Their mental states are characterized by using the notions of beliefs, goals, desires, intentions, and local preferences. Each agent has a set of desires. The agent's activities are motivated by the will to fulfill these desires. At any given time, an agent selects a consistent subset of its desires. This serves as its set of current goals. An agent ascribes different degrees of importance to different goals. It prefers to fulfill goals of higher importance. The set of goals motivate the agent's planning process.

The planning process may generate several intentions. Some of these are in what we would like to classify as the "intend-to-do" category and refer to actions that are within the direct control of the agent. Others are among the "intend-that" category [13,56,57,158]. These are propositions not directly within the agent's realm of control, that the it must rely on other agents for satisfying. [1] Often, there is room for argumentation when intend-that actions are part of a plan. Argumentation is the means by which an agent, the persuader, attempts to modify the intention structure of another agent, the persuadee, to include the actions the persuader wants it to do. While an agent tries to influence the intentions of other agents, other agents may try to convince it as well. The role of persuader and persuadee is not fixed, but dynamically assumed during the agent interactions. Thus, during a negotiation process,

[1] The proposition may include a negation. When fulfillment of the proposition is beyond the control of the agent, it can be achieved by convincing another agent to abandon a relevant intention, or by convincing it to take an action that will make the proposition true.

each agent may update its intentions and goals after receiving a message from another agent. If the argumentation happens to fail, the agent which sent it must revise its arguments, its plans, and/or seek other sources of satisfying the portion of its plan in question.

An agent's belief set includes beliefs concerning the world and beliefs concerning mental states of other agents. An agent may be mistaken in both kinds of beliefs. It may update its beliefs by observing the world and after receiving messages from other agents. Each agent's actions are based upon its mental model of other agents. The types of arguments (see Section 3) that a persuader generates depend on its knowledge of a persuadee's mental model. An important piece of knowledge for argument selection is a persuader's assessment of the relative importance of a persuadee's goals. For example, a threat is effective if it threatens an important persuadee goal. Incomplete information or information that is contrived by a deceitful agent may result in a discrepancy between the actual and portrayed mental models. Argumentation is especially crucial in these situations, since it can be used to establish a common platform of agreement despite these differences.

2.1 The Formal Model

We will use *minimal structures* [15] style semantics for each of the notions of beliefs, desires, goals, and intentions. The modal operators have certain desired properties from the point of view of our axiomatization. We assume that the agent may not be omniscient (may not have as beliefs all consequences of its "primitive" beliefs [157,105].) Its set of beliefs may not be consistent, and it may not be aware of the inconsistency. As we discuss later, omniscience (or the lack of it) is very important in the context of negotiation and argumentation where agents usually transfer facts and their conclusions.

The set of an agent's desires may not always be consistent either. For example, an agent may desire to earn money today, but also to go on a vacation, and the two desires may lead to a contradiction (see also [152]). Usually, an agent has preferences among its contradicting desires [2]. The set of goals is a consistent subset of the set of desires. Similarly, we have some implicit properties in mind for actions in the "intend-to-do" category. When an action serves to contribute to one or more of the agent's desires, the agent may have the intention to act. The intention may contribute directly to the fulfillment of a desire, or indirectly, through another intention. The action may have a side-effect [16,12] that does not contribute to any of the agent's desires. In such

[2] The issue of how an agent forms its original set of desires is not in the scope of this paper. However, this set may change over time, as the agent gathers more information and updates its knowledge.

a case, the agent does not intend the side-effect. Thus we require that the intentions be consistent but not confined to side-effects. [3]

Briefly, we have a set of time lines, each of which extends infinitely far from the past into the future (see [152]). We use time lines instead of more usual worlds because they provide a simple, useful way of incorporating time into our system. With each time point, time line and predicate, we associate a set of sequences of elements (intuitively, the sequence of elements that have the property of the predicate, at the time point of the time line).

A notion of *satisfaction* of a sentence ψ in a time line of a structure, given an interpretation, is defined (denoted by $M, l, \bar{v} \models \psi$, see Section 2.3). The *intension* of a sentence in the language is the set of time-lines in which the sentence is satisfied, i.e., $\|\psi\| = \{l | M, l, \bar{v} \models \psi\}$. [4] A sentence is a belief at a given time point at a given time line if its intension is belief-accessible. According to this definition, the agent's beliefs are not closed under inferences; the agent may even believe in contradictions. We will later define different types of agents, in accordance with different properties of their beliefs.

Similarly, we assume that accessibility relations associated with desires, intentions, and goals are between time lines and time points, and sets of time lines [157]. An agent *intends* (resp. *desires*, has *goal*) ψ at time t, if the intension of ψ ($\|\psi\|$) is a member of the set of sets of time lines that are intention-accessible (resp. desires-accessible, goals-accessible) at time t. We further impose restrictions on the set of sets of time-lines that are intention-accessible (resp. desires-accessible, goals-accessible) to an agent. An agent intends ψ in order to contribute to φ if it intends ψ, intends φ and intends that ψ implies φ. An agent prefers ψ over φ at a give time t, if the agent prefers $\|\psi\|$ at time t over $\|\varphi\|$ at time t.

A message may be one of the following types: a request, response, or a declaration. A response can be an acceptance or a rejection. A message may carry an argument as a justification. Arguments are produced using special argumentation axioms. An agent can send and receive messages. Unlike Werner's approach [164], we do not assume that receiving a message in itself changes the mental state of the agent. Even receiving an informative message does not change the agent's beliefs, unless the agent evaluates the message and decides

[3] While the issue of how to model the concept of intentions is a very involved topic removed from the main focus of our work, we devote some effort to tailoring our semantics of the intention and desire operators to reflect these desired properties. Our main concern remains identifying the process of change in these modalities during argumentation.

[4] Note that if two sentences have the same intensions $\|\psi\| = \|\varphi\|$, then they are *semantically equivalent.*

that it should add it to its beliefs [5]. Evaluating a received message is useful especially since we assume that agents are untrustworthy, and may even be untruthful. Only an evaluation process following an argument may change the internal state of the agent.

2.2 Syntax

We denote by $\mathcal{A}gents$ the set of agents. We assume that there are four modal operators for agent i: Bel_i for beliefs, $Desire_i$, for desires, $Goal_i$ for goals and Int_i for intentions. It may be the case that the agent is motivated by the need to satisfy its own goals or desires, or that it is convinced to perform an action following an argument. [6] In addition, we assume that there is another modal operator, $Pref_i$ which is associated with the agent's preferences among goals, desires, and intentions. As we mentioned above, following [152], the basis for our formalism is a simple temporal language. Informally, we have a set of time lines (which play the role of "worlds" in the modal logic). The set of time lines is infinite. We also have a set of time points. At every time in each time line, some propositions are true (and the rest are false). [7]

Our variables and constants are sorted. We have a set TC of time point constants, a set TV of time point variables $(t, \hat{t}, t_1, t_2, ...)$, a set AC of agent constants, a set AV of agent variables $(i, j, ...)$, a set AcC of action constants and a set AcV of action variables $(\alpha, \beta, ...)$, a set PC of preference values constants and a set PV of preference values variables $(p, p_1, p_2, ...)$, and a set $Pred$ of predicate symbols including two special 2-ary predicates Do and $Capable$. We denote by $Variables$ the set of all variables (including AV, AcV, TV and PV), by $Constants$ the set of all constants (including AC, AcC, TC and PC), and by $Terms$ the set of variables and constants. We also use the symbol nil. We first define the set of the well-formed formulas (wff) of our language.

(1) If $t_1, t_2 \in TC \cup TV$, then $t_1 < t_2$ is a wff.
(2) If $x_1, x_2 \in Terms$, then $x_1 = x_2$ is a wff.
(3) If $P \in Pred$ is a k-ary predicate, $x_1, ..., x_n$ are terms, and $t \in TC \cup TV$, then $[t, P(x_1, ..., x_n)]$ is a wff (read as: $P(x_1, ..., x_n)$ *is true at time* t).

[5] The new information may be inconsistent with the agent's current beliefs. We leave this for future discussion. See for example, [6,26,53,96,166,103,125,22,66,27].
[6] Our intention model is closer to Shoham [138]'s *Dec* and Thomas et al. [152]'s *Comit* than to Cohen's and Levesque [16]'s "Intend".
[7] We have extended [152] to deal with the FOL case. We prefer this approach, where time can be expressed explicitly, over others where time periods cannot be expressed in the language (for example [16]), since threats and arguments both evolve in time. We use an extension of first order logic, rather an extension of propositional logic since it is useful in the formalism of argumentation.

(4) If φ is a wff and ψ is a wff, then so are $\varphi \& \psi$ and $\neg\varphi$. If φ is a wff and $x \in Variables$, then $\forall x \varphi$ is a wff. $\exists, \vee, \rightarrow$ have their usual meanings.

(5) If φ and ψ are wffs, $t \in TC \cup TV$, $i, j \in AC \cup AV$ and $p \in PC \cup PV$, then the following expressions are wffs:
   (a) $[t, Bel_i \varphi]$ (*i believes* φ *at time* t),
   (b) $[t, Desire_i(\varphi, p)]$ (*i desires* φ *at time* t *with preference* p),
   (c) $[t, Goal_i \varphi]$ (*i has a goal* φ *at time* t),
   (d) $[t, Int_i \varphi)]$ (*i intends* φ), $[t, Int_i(\varphi, \psi)]$ (*i intends* φ *at time* t *to contribute to* ψ),
   (e) $[t, Pref_i(\varphi, \psi)]$ (*i prefers* φ *over* ψ *at time* t),
   (f) $[t, Agent(\psi, i)]$ (*i is the agent of* ψ).

(6) If φ and ψ are wffs, then the following expressions are messages:
   (a) $Request(\psi, \varphi)$ (ψ *is requested with the argument* φ),
   (b) $Reject(\psi, \varphi)$ (ψ *is rejected with the argument* φ),
   (c) $Accept(\psi, \varphi)$ (ψ *is accepted with the argument* φ),
   (d) $Decl(\psi)$ (ψ *is declared*),
   (e) $Accept(\psi)$, $Request(\psi)$, $Reject(\psi)$ (ψ *is accepted (resp. requested or rejected) with no argument*)).

(7) If m is a message, $t \in TC \cup TV$ and $i, j \in AC \cup AV$, then $[t, Receive_{ij} m]$ (*i receives m from j at time t*) and $[t, Send_{ij} m]$ (*i sends m to j at time t*) are wffs.

We assume that there are two, 2-ary predicates in $Pred$, Do and $Capable$, where $[t, Do(i, \alpha)]$ is read as α is done by i at time t and $[t, Capable(i, \alpha)]$ is read as at time t agent i is able to perform α. We will sometimes use the abbreviation $[t, \varphi \& \psi]$ for $[t, \varphi] \& [t, \psi]$ and will freely interchange $[t, \neg\varphi]$ and $\neg[t, \varphi]$. We will use similar abbreviations for $\vee$ and $\rightarrow$.

Requests and responses may include arguments. For example, an agent a may send its opponent, agent b, a message at time period t, requesting b to let him use its printer at time t_1; with the threat that otherwise a will break it at time t_2. Formally, it can be expressed as :
$[t, Request([t_1, Do(b, let.use.printer], \neg[t_1, Do(b, let.use.printer)] \rightarrow [t_2, Do(a, break.printer)])]$.
As can be seen from the example, the operator Do is useful in requests, responses, and arguments and was added to the language to be able to explicitly specify the agent who will carry out an action.

2.3 Semantics

We start with the semantics of the various sentences of our language. This will be followed by the semantics for our modal operators.

Time is a pair $\langle T, \prec \rangle$, where T is a set of time points and $\prec$ is a total order on T (unbounded in both directions).

A BDIG model M is a structure
$< \Xi, L, \mathcal{A}gents, A, B, G, D, It, P, RECEIVE, SEND, \Phi, v, \mathcal{M} >$, where

(1) Ξ is a set of elements in the agent's environment, and $\mathcal{M}$ is a set of messages.
(2) L is a set of time-lines.
(3) $\mathcal{A}gents$ is a set of agents.
(4) $B : L \times T \times \mathcal{A}gents \rightarrow 2^{2^L}$ is the belief-accessibility relation.
(5) $G : L \times T \times \mathcal{A}gents \rightarrow 2^{2^L}$ is the goals-accessibility relation.
(6) $It : L \times T \times \mathcal{A}gents \rightarrow 2^{2^L}$ is the intention-accessibility relation.
(7) $D : L \times T \times \mathcal{A}gents \rightarrow 2^{2^L}$ is the desire-accessibility relation.
(8) $P : L \times T \times \mathcal{A}gents \times 2^{2^L} \rightarrow I\!R$ indicates for each agent the value (preference) it associates with different propositions at a given time.
(9) Φ interprets predicates and v interprets constants.
(10) $RECEIVE : L \times T \times \mathcal{A}gents \times \mathcal{A}gents \rightarrow \mathcal{M}$ indicates the messages received by the agents; $SEND : L \times T \times \mathcal{A}gents \times \mathcal{A}gents \rightarrow \mathcal{M}$ indicates the messages sent by the agents. [8]
(11) $A : L \times T \times \rightarrow 2^{2^L} \rightarrow \mathcal{A}gents \cup \{nil\}$ allocates an agent (if any) that performs an action in a given time period.

The domain of quantification is $\Theta = \Xi \cup T \cup \mathcal{A}gents \cup \mathcal{M} \cup I\!R$. Given this, $\Phi : Pred^k \times L \times T \rightarrow \Theta^k$. $\bar{v}$ is the extension of v to all $Variables$. If for any extension v' of v $M, l, v' \models \psi$, we say that M, l satisfy ψ ($M, l \models \psi$). Given a structure M, and a wff ψ, we denote by $||\psi||$ the set $\{l \mid l \in L, M, l, \bar{v} \models \psi\}$. The definition of satisfaction is as follows:

(1) If $t_1, t_2 \in TC \cup TV$, then
$M, l, \bar{v} \models t_1 < t_2$ iff $\bar{v}(t_1) \prec \bar{v}(t_2)$.
(2) If $x_1, x_2 \in Terms$, then
$M, l, \bar{v} \models x_1 = x_2$ iff $\bar{v}(x_1) = \bar{v}(x_2)$.
(3) If $P \in Pred$ is a k-ary predicate, $x_1, ..., x_n$ are terms, and $t \in TC \cup TV$, then
$M, l, \bar{v} \models [t, P(x_1, ..., x_k)]$ iff $< \bar{v}(x_1), ..., \bar{v}(x_k) > \in \Phi[P, l, \bar{v}(t)]$.
(4) If φ is a wff, ψ is a wff and $x \in Variables$, then
  (a) $M, l, \bar{v} \models \neg\varphi$ iff $M, l, \bar{v} \not\models \varphi$;
  (b) $M, l, \bar{v} \models \varphi \& \psi$ iff $M, l, \bar{v} \models \varphi$ and $M, l, \bar{v} \models \psi$;
  (c) $M, l, \bar{v} \models \forall x\ \varphi$ iff for every $\bar{v}'$ which agrees with $\bar{v}$ everywhere, except possibly on x $M, l, \bar{v}' \models \varphi$.
(5) If φ and ψ are wffs, $t \in TC \cup TV$, $i, j \in AC \cup AV$ and $p \in PC \cup PV$, then
  (a) $M, l, \bar{v} \models [t, Bel_i\varphi]$ iff $\|\varphi\| \in B(l, \bar{v}(t), \bar{v}(i))$;

[8] We note that if for all $l \in L, t \in T$ and $i, j \in \mathcal{A}gents$, $RECEIVE(l, t, i, j) = SEND(l, t, j, i)$, then the communication is reliable. See also [40].

(b) $M, l, \bar{v} \models [t, Desire_i(\varphi, p)]$ iff $\|\varphi\| \in D(l, \bar{v}(t), \bar{v}(i))$ and $P(l, \bar{v}(t), \bar{v}(i), \|\varphi\|) = \bar{v}(p)$;
(c) $M, l, \bar{v} \models [t, Goal_i\varphi]$ iff $\|\varphi\| \in G(l, \bar{v}(t), \bar{v}(i))$;
(d) $M, l, \bar{v} \models [t, Int_i\varphi]$ iff $\|\varphi\| \in It(l, \bar{v}(t), \bar{v}(i))$;
(e) $M, l, \bar{v} \models [t, Pref_i(\varphi, \psi)]$ iff $P(l, \bar{v}(t), \bar{v}(i), \|\varphi\|) > P(l, \bar{v}(t), \bar{v}(i), \|\psi\|)$;
(f) $M, l, \bar{v} \models [t, Agent(\varphi, i)]$ iff $A(l, \bar{v}(t), \|\varphi\|) = \bar{v}(i)$.

(6) If m is a message, $t \in TC \cup TV$ and $i, j \in AC \cup AV$, then
(a) $M, l, \bar{v} \models [t, Receive_{ij}m]$ iff $\bar{v}(m) \in RECEIVE(l, \bar{v}(t), \bar{v}(i), \bar{v}(j))$
(b) $M, l, \bar{v} \models [t, Send_{ij}m]$ iff $\bar{v}(m) \in SEND(l, \bar{v}(t), \bar{v}(i), \bar{v}(j))$;

We extend the semantics to deal with intending ψ to contribute to φ as follows: $M, l, \bar{v} \models [t, Int_i(\psi, \varphi)]$ iff $M, l, \bar{v} \models [t, Int_i(\psi)]$, and $M, l, \bar{v} \models [t, Int_i(\varphi)]$ and $M, l, \bar{v} \models [t, Int_i(\psi \to \varphi)]$. Note that in our model the domain of quantification for all the time lines (that play the role of possible worlds) is the same [81]. We also assume that the constant symbols are rigid designators.

A BDIG model $M =< \Xi, L, \mathcal{A}gents, A, B, G, D, I, P, RECEIVE, SEND, \Phi, v, \mathcal{M} >$ is said to *validate* a formula φ if for every $l \in L$, $M, l \models \varphi$. A formula φ is valid if it is validated by any BDIG model.

The agents in our general framework have very little reasoning power, and their attitudes do not have any appropriate properties, as stated in the next proposition. First, all tautologies and inference rules of first order logic are valid in BDIG models. Second, by virtue of a model that is based on *intensions* of formulas, it is difficult to distinguish between semantically equivalent beliefs, intentions desires and goals. Therefore, the following inference rule (R1), is always sound in a minimal model structures. It indicates that if an agent believes φ, and φ is equivalent to ψ, then the agent believes ψ. [9]

Proposition 1 The following formal system is sound and complete for validity in BDIG models.

$$\textbf{(A0)} \quad \text{All tautologies of first order logic.} \tag{1}$$

$$\textbf{(MP)} \quad \text{From } \varphi \text{ and } \varphi \to \psi \text{ infer } \psi. \tag{2}$$

$$\textbf{(GR)} \quad \text{From } \varphi \text{ infer } \forall x\varphi. \tag{3}$$

(**R1**) From $\varphi \leftrightarrow \psi$ infer

$$[t, Bel_i\varphi] \leftrightarrow [t, Bel_i\psi], \quad [t, G_i\varphi] \leftrightarrow [t, G_i\psi],$$
$$[t, Int_i\varphi] \leftrightarrow [t, Int_i\psi], \quad [t, D_i(\varphi, x)] \leftrightarrow [t, D_i(\psi, x)].$$

[9] The introduction of the agent's language G to the system as was suggested in [106] will reduce the effect of this rule; the agent will believe, desire, intend, or have a goal ψ only if ψ is in its language.

□

The proof is done using the canonical model technique (e.g., [15]) and ideas from [47,157,106]. Note that since in our model the domain of quantification for all the time lines is the same, and, similarly, the constant and variable assignment is the same for all the time lines, we do not run into the classical problems of "quantifying in."

Even though the agents in our model are so simple, this model can be the basis for introducing different properties to the attitudes, given specific features that the designer of an agent would like to impart to the agent. Furthermore, different types of agents can satisfy different axioms, which can be characterized in a precise way by appropriate conditions with respect to the accessibility relations of the models. We will discuss this issue in the following sections.

2.3.1 Properties of the Modalities

In all the following axiom schemas, we will assume that the unbounded variables are universally quantified as follows: $\forall l \in L, a \in \mathcal{A}gents, \tau, \tau' \in T$. In addition, in all the axiom schemas, we assume that $i \in AC \cup AV$, $t \in TC \cup TV$ and that ψ and φ can be replaced by any wff in the language.

Let us start with the semantics for the intention operator (It). Following Bratman [12], we would like the intentions to be consistent. This can be achieved by introducing two constraints on the intention-accessibility relation It.

$$\textbf{(CINT1)} \qquad \emptyset \notin It(l, \tau, a). \tag{4}$$

$$\textbf{(CINT2)} \qquad \text{If } U \in It(l, \tau, a) \ \text{ and } \ V \in It(l, \tau, a) \text{ then } U \cap V \neq \emptyset. \tag{5}$$

The following axiom (schema) and inference rule are sound with respect to the above conditions. [10]

Proposition 2 A BDIG model that satisfies conditions (CINT1:4) and (CINT2:5) validates the axiom

$$\textbf{(INT1)} \qquad [t, \neg Int_i \text{false}] \tag{6}$$

and validates the inference rule

$$\textbf{(INT2)} \qquad \text{From } \varphi \to \neg\psi \text{ infer } [t, Int_i\varphi] \to \neg[t, Int_i\psi]. \tag{7}$$

[10] When referring to a condition or an axiom we use both its name and the equation serial number. For example, in (CINT1:4), 4 specifies the equation number of the condition labeled with (CINT1).

□

The consistency of the set of intentions at any given time is a basic premise for the argumentation system. For example, suppose an agent wants its opponent to intend-to do α which contributes to its intentions and goals. This intention (α) may contradict other intentions of its opponent. Due to the consistency requirement, the agent must convince its opponent (using argumentation) to give up its original contradictory intentions to make place for α. As we will see later, the consistency requirement can guide the persuader's argument generation process.

There are several other properties of intentions that are controversial, for example when an agent intends φ and intends ψ, whether it intends the conjunction, and vice versa, i.e., if the agent intends $\varphi \& \psi$, whether it intends each of them separately. The first property is more acceptable (see [73]). If, for example, an agent promised another agent to do α_1, and it intends to do it and also promised it to do α_2, and it intends to do it, then it is reasonable that the agent intends to do both actions.

In our system this is captured by the following property.

$$\textbf{(CINT3)} \quad \text{If } U \in It(l,\tau,a) \text{ and } V \in It(l,\tau,a) \text{ then } U \cap V \in It(l,\tau,a). \tag{8}$$

The following axiom (schema) is sound with respect to the above condition.

Proposition 3 *A BDIG model that satisfies condition (CINT3:8) validates the axiom*

$$\textbf{(INT3)} \qquad [t, Int_i\psi]\&[t, Int_i\varphi] \to [t, Int_i\psi\&\varphi]. \tag{9}$$

□

Adopting the axiom in the reverse direction, i.e. adopting $[t, Int_i(\varphi\&\psi)] \to [t, Int_i\ \varphi]\&[t, Int_i\psi]$ seems less reasonable. Suppose an agent promised to move Block A to location s and to put Block B on Block A. It is not clear that it intends to do each one of the actions separately. In many cases, there is no benefit in performing each action separately, but only doing them together is beneficial (see also [127]).

If one wouldn't adopt the axiom that requires splitting a conjunctive intention, one cannot require the agent's intentions to be closed under consequences. However, if such an axiom is appropriate in a specific application, the following axiom may also be sound:

$$\textbf{(INT4)} \qquad [t, Int_i\psi]\&[t, Int_i\psi \to \varphi] \to [t, Int_i\varphi] \tag{10}$$

Closure under consequence warrants, in addition to (CINT3:8), another restriction on It.

Proposition 4 *A BDIG model that satisfies condition*

$$\textbf{(CINT5)} \qquad \textit{If } U \in It(l,\tau,a), \quad \textit{and} \quad U \subseteq V, \quad \textit{then} \quad V \in It(l,\tau,a). \tag{11}$$

validates the axiom

$$\textbf{(INT5)} \qquad [t, Int_i(\varphi \& \psi)] \rightarrow [t, Int_i\ \varphi] \& [t, Int_i \psi.] \tag{12}$$

□

It is clear from Propositions 3 and 4 that closure under consequence of intention (INT4:10) is valid in models that satisfy condition (CINT3:8) and (CINT5:11).

(INT5:12), is a special case of the following inference rule which is also valid in models that satisfy condition (CINT5:11):

$$\textbf{(RINT5)} \qquad \text{From } \varphi \rightarrow \psi \text{ infer that } [t, Int_i\ \varphi] \rightarrow [t, Int_i \psi] \tag{13}$$

□

We will make similar restrictions on G and obtain similar properties for goals. Intentions and goals are consistent, since the agent needs to act and plan according to its intentions and goals.[11] However, desires may be inconsistent, as mentioned above. An agent may desire to dig on Mars, but also to conserve its battery power, and the two desires may lead to a contradiction (see also [152]). However, we do not want the agent to desire falsely.[12] Usually, an agent has some preferences among its contradicting desires.

We impose the following restrictions on the desires (D) operator:

(CD1) $\emptyset \notin D(l,\tau,a)$.
(CD2) If $U \in D(l,\tau,a)$ and $U \subseteq V$ then $V \in D(l,\tau,a)$.[13]

These restrictions yield axiom schemas similar to (INT1:6) and (INT4:10), where Int_i is replaced by $Desire_i$.

[11] Another good property for agents is that their intentions and goals be consistent with respect to their beliefs. We discuss this in Section 2.6.
[12] Note that there is a difference between $[t, Desire_i(\varphi, p_1)] \& [t, Desire_i(\neg\varphi, p_2)]$ and $[t, Desire_i(\varphi \& \neg\varphi, p)]$. We allow the first case, but not the second one.
[13] CD1 is similar to (CINT1:4) and (CD2) is similar to (CINT5:11).

We take a different approach concerning preferences and desires than [162]. We assume that the agent's preferences are over the sets of time lines, while [162]'s preferences are over single models. An agent prefers φ over ψ if it associates a higher value to the intension of φ ($\|\varphi\|$) than to the intension of ψ ($\|\psi\|$). In different models, different restrictions may be put on P.

The agent's desires are not derived from its preferences (see also [69]), but we make the following restriction on the model:

$$(\mathbf{CPD})\ \ \forall U, U' \in 2^L,\ \text{if } U \in D(l, a, \tau) \text{ and } P(l, \tau, a, U) < P(l, \tau, a, U') \text{ then } U' \in D(l, \tau, a). \tag{14}$$

Hence, in our model the following axiom is sound.

Proposition 5 A BDIG model that satisfies the condition (CPD:14) validates the following axiom:

$$\mathbf{(PD)}\ \ [t, Desire_i(\psi, p)]\&[t, Pref_i(\varphi, \psi)] \rightarrow (\exists p')((p' \geq p)\&[t, Desire_i(\varphi, p')]). \tag{15}$$

□

The property that desires may not be consistent plays a role in argumentation. In some situations, an agent tries to convince its opponent to perform an action that contradicts the opponent's current set of goals. However, it may contribute to one of the opponent's desires. We discuss this case in Section 3.7.

2.4 Agent Types

Within the general framework defined above, it is possible to define various types of agents. In the following subsections, we define the additional conditions on the models that these agents must satisfy in order to have a particular character. In addition, we define properties of the model associated with changes over time, as well as agent types that arise from different assumptions as to interrelations among modalities of the model. In Section 3.8, we discuss how agent types may be guiding factors in the selection of argument categories and the generation of arguments.

2.4.1 Properties Associated with Reasoning Power

The minimal properties we would like an agent to have with respect to its beliefs is that it will not believe in "false". However, as was discussed above, an agent whose beliefs are not closed under consequences may believe in contradiction, without being aware of it. This leads us to the definition of the following simple agent.

Bounded Agent

We would like a bounded agent to have the following axiom:

$$\textbf{(B1)} \qquad [t, \neg Bel_i \text{false}]. \tag{16}$$

So that an agent does not believe in "false", we need to impose additional restrictions on its belief-accessibility relation.

Proposition 6 A BDIG model that satisfies the condition

$$\textbf{(CB1)} \qquad \emptyset \notin B(l, \tau, a) \tag{17}$$

validates axiom (B1:16). □

We further assume that all the other types of agents do not believe in "false", either. The beliefs of a bounded agent are not closed under consequences, i.e., an agent may believe that φ and that $\varphi \rightarrow \psi$, but it may not believe in ψ. However, as we discussed in Section 2.3, it cannot distinguish between semantically equivalent formulas.

An Omniscient Agent

An agent whose beliefs are closed under inferences is said to be omniscient. For omniscience we impose the following additional conditions on the belief accessibility relation. These render its model equivalent to a Kripke Structure.

$$\textbf{(CB2)} \quad L \in B(l, \tau, a). \tag{18}$$

$$\textbf{(CB3)} \quad \text{If } \; U \in B(l, \tau, a) \; \text{ and } \; U \subseteq V \text{ then, } V \in B(l, \tau, a). \tag{19}$$

$$\textbf{(CB4)} \quad \text{If } \; U \in B(l, \tau, a) \; \text{ and } \; V \in (l, \tau, a) \text{ then, } \; U \cap V \in B(l, \tau, a). \tag{20}$$

A BDIG model that satisfies conditions (CB2:18)-(CB4:20) corresponds to a Kripke structure, and every Kripke structure corresponds to a BDIG model that satisfies conditions (CB2:18)-(CB4:20) [157], and thus such BDIG models validate the axioms of the modal logic K (e.g., [15]).

Proposition 7 BDIG models that satisfy conditions (CB2:18)-(CB4:20) validate the following axioms:

$$\textbf{(B2)} \qquad [t, Bel_i \text{true}] \tag{21}$$

$$\textbf{(B3)} \qquad [t, Bel_i \psi \& \varphi] \rightarrow [t, Bel_i \psi] \& [t, Bel_i \varphi] \tag{22}$$

$$\textbf{(B4)} \qquad [t, Bel_i \psi] \& [t, Bel_i \varphi] \rightarrow [t, Bel_i \psi \& \varphi]. \tag{23}$$

□

There may be other types of agents that may have only a partial set of the axioms of the omniscient agent, for example, an agent that does not believe in tautologies (without Axiom (B2:21)), but can reason using axioms (B3:22) and (B4:23).

While we assume that there are agents whose beliefs are not closed under consequences, we do assume that all agents' intentions and goals are closed under consequences. This is justifiable, since the set of intentions is much smaller than the set of beliefs. The agent is aware of its intentions, since it needs to search for plans to achieve them. Therefore, its intentions are under its scope, and it is reasonable to assume that the agent can compute their closure under consequence.

A Knowledgeable Agent

An agent is knowledgeable if its beliefs are correct. The corresponding axiom schema is:

$$\textbf{(B5)} \qquad [t, (Bel_i \varphi) \rightarrow \varphi]. \tag{24}$$

The related condition, which makes this axiom sound, is specified in the following proposition.

Proposition 8 BDIG models that satisfy the condition

$$\textbf{(CB5)} \quad \text{if } U \in B(l, \tau, a), \text{ then } l \in U, \tag{25}$$

validates axiom (B5:24). □

So far, the agent typology, namely omniscient and knowledgeable agents, has considered only properties local to the agent at a particular time interval. In an open world environment, however, agents' intentions can change over time. Change in intentions can be the result of knowledge that an agent keeps track of as time changes, or new knowledge and/or beliefs that an agent obtains by observing its environment, or through communications from other agents. The following agent typology takes into consideration how the passage of time interacts with an agent's beliefs and intentions.

An Unforgetful Agent

An agent who does not forget anything follows the following axiom:

$$\textbf{(BUF)} \qquad \forall t, t'((t' \geq t) \rightarrow [t, Bel_i\varphi] \rightarrow [t', Bel_i\varphi]) \tag{26}$$

An agent who does not forget anything can be characterized according to the following proposition.

Proposition 9 BDIG models that satisfy condition

$$\textbf{(CBUF)} \qquad \text{if } \ \tau \prec \tau', \ \text{ then } B(l,\tau,a) \subseteq B(l,\tau',a) \tag{27}$$

validates axiom (BUF:26). □

A Memoryless Agent

We would like to characterize agents that do not have memory and cannot reason about past events. An agent doesn't have a memory under the following condition[14] : if $U \in B(l,\tau,a)$ and $l' \in U$ then for every $\tau' \prec \tau$ (i) for every $P \in Pred$, $\Phi[P,l,\tau'] = \emptyset$; (ii) $B(l',\tau',a) = \emptyset$; (iii) $G(l',\tau'a) = \emptyset$; (iv) $It(l',\tau'a) = \emptyset$; (v) $D(l',\tau'a) = \emptyset$; (vi) for every $a_1, a_2 \in \mathcal{A}gents$, $RECEIVE(l',\tau',a_1,a_2) = \emptyset$ and $SEND(l',\tau',a_1,a_2) = \emptyset$.

A Non-observer

In some situations, it is useful to assume that an agent's beliefs change only

[14] Other reasonable restrictions can be chosen for characterizing a memoryless agent.

as a result of message evaluation; i.e. the agent doesn't observe things, and its only source of information is communication with other agents.

We make the following restriction on the model of such an agent: if an agent does not receive any message at a given time period, then if it believes in something, it will keep believing in it during the next time period, and the agent will not adopt new beliefs.

If $\forall b \in \mathcal{A}gents, RECEIVE(l, \tau, a, b) = \emptyset$, then $B(l, \tau, a) = B(l, \tau - 1, a)$.

Cooperative Agents

A group A of agents $A \subseteq \mathcal{A}gents$ is cooperative[15] in a BDIG model M, a time line l a time point t and an interpretation $\bar{v}$, if it shares common goals. This imposes the following condition: $\bigcap_{a \in A} G(l, t, a)) \neq \emptyset$.
Furthermore, we require that the goals are common belief.[16] That is, let Δ be the set of common goals, i.e., $\Delta = \{\psi \mid \|\psi\| \in \bigcap_{a \in A} G(l, t, a)\}$, and $A^c \subseteq AC$ are the "names" of the agents of A according to $\bar{v}$, then $M, l, \bar{v} \models [t, C \bigwedge_{\psi \in \Delta} \bigwedge_{j \in A} Goal_j \psi]$. It is easy to see that the set of common goals of cooperative agents is consistent.

Cooperative agents may have contradictory goals, e.g., $[t, G_i\psi]\&[t, G_j\neg\psi]$. These goals do not belong to the set of common goals. Our definition of the cooperativeness of agents may be time-dependent. A set of agents that are cooperative at a given time period may become adversaries at a later time period, when their common goals do not exist anymore. Our notion of cooperative agents is less restrictive than the notion of SharedPlans of [56], the notions of Cohen, Levesque, and Nunes [85] of joint persistent goals, and joint intentions or collaborative activity of [143]. However, as demonstrated in the example presented in Section 4.7, even this weak notion of cooperation may enhance the negotiation.

[15] Among cooperative agents, as among non-cooperative agents, conflicts may occur, and negotiation may be required. However, the argumentation is of a different nature, and we shall not dwell on that here.

[16] We denote by $E\psi$ the property that all the agents of A believe ψ. ψ is common belief at time t if everyone in A believes ψ, everyone believes that everyone believes ψ and so on [75]. $C\psi$ denotes that ψ is common knowledge [60].

So far we have presented axioms and semantics conditions to define properties of each modality. Now we will move to investigate inter-relations among the different modalities. First, every goal is also a desire. [17]

$$\textbf{(GD)} \qquad [t, Goal_i(\varphi)] \to [t, (\exists p) Desire_i(\varphi, p)]. \tag{28}$$

The correspondence restriction on the goal and on the desire-accessibility relations is described in the following proposition.

Proposition 10 A BDIG model that satisfies the condition

$$\textbf{(CGD)} \qquad \text{if } U \in G(l, \tau, a), \text{ then } U \in D(l, \tau, a) \tag{29}$$

validates axiom (GD:28). □

An agent adopts all its goals as intentions:

$$\textbf{(GINT)} \qquad [t, Goal_i \varphi] \to [t, Int_i \varphi]. \tag{30}$$

The correspondence restriction on the goal and on the intention-accessibility relations is similar to (CGD:29).

Proposition 11 A BDIG model that satisfies the condition

$$\textbf{(CGINT)} \qquad \text{if } U \in G(l, \tau, a), \text{ then } U \in It(l, \tau, a), \tag{31}$$

validates the axiom (GINT:30). □

However, there may be intentions that are not goals. An agent may hold an intention in response to a threat or promise for a reward. During the argumentation, the agent may come to have an intention to prevent the opponent from carrying out the threat, or to convince it to give a reward, which only indirectly contributes to one of the agent's goals. An agent is aware of its intentions and, moreover, its beliefs about its own intentions are correct:

$$\textbf{(INTB1)} \qquad [t, Int_i \varphi] \leftrightarrow [t, Bel_i [t, Int_i \varphi]]. \tag{32}$$

[17] In Cohen and Levesque's framework [16], all the agent's beliefs are also its goals. We do not have such a property. An agent may believe p, but may not desire p and may not adopt it as one of its goals.

The correspondence restriction on the structure is as stated in the following proposition.

Proposition 12 A BDIG model that satisfies the condition

$$\textbf{(CINTB1)} \qquad U \in It(l,\tau,a) \text{ iff } \{l' | U \in It(l',\tau,a)\} \in B(l,\tau,a) \qquad (33)$$

validates axiom (INTB1:32). □

We assume that an agent's intention doesn't contradict its beliefs:

$$\textbf{(INTB2)} \qquad [t, Int_i\varphi] \rightarrow [t, \neg Bel_i \neg\varphi]. \qquad (34)$$

The corresponding restriction on the It and B relations is as follows.

Proposition 13 A BDIG model that satisfies the condition,

$$\textbf{(CINTB2)} \qquad \text{if } U \in It(l,\tau,a), \text{ then } L \setminus U \notin B(l,\tau,a), \qquad (35)$$

validates the axiom (INTB2:34). □

In order to understand the intuition behind axiom (INTB2:34), we compare our notion of intentions with Cohen and Levesque [16,18]'s persistence goals (P-GOAL). [18] Cohen and Levesque assume that if an agent has a P-GOAL toward a proposition, then the agent believes that this proposition is not true now, but that it will be true at some time in the future. The agent will drop a persistent goal p only if it comes to believe that p is true or that p is impossible. In their logic, time doesn't explicitly appear in the proposition; thus, they cannot express P-GOAL toward propositions that will be true at some specific time in the future or consider situations where a proposition is true now, but which the agent believes will become false later and therefore has a P-GOAL to make it true again after it becomes false. Since time is explicit in our logic, we can express such intentions, in addition to expressing Cohen and Levesque's attitude P-GOAL. For example, $t < t_1 \& [t, Int_i[t_1, On(A,B)]]$ intuitively means that at time t, agent i intends that block A will be on block B at some later time t_1. In our framework, there is no relation between this intention and whether agent i believes, at time t, that $On(A,B)$ is not true at time t. Furthermore, we may express intentions toward propositions with different time points. In such intentions, the propositions may include the same predicates, e.g. $t < t_1 < t_2 \& [t, Int_i[t_1, On(A,B)] \& \neg[t_2, On(A,B)]]$, or different predicates, e.g., $t < t_1 < t_2 \& [t, Int_i[t_1, On(A,B)] \& [t_2, On(C,B)]]$. In both cases, $[t, Bel_i[t, On(A,B)]]$ may be true or may be false. However, we require,

[18] Cohen and Levesque's concept of intention is based on their notion of P-GOAL.

in axiom (INTB2:34), that at time t the agent will believe that it is possible that at time t_1 block A will be on block B, i.e. $\neg[t, Bel_i \neg[t_1, On(A,B)]]$.

This expressibility of our system, enables us to characterize different types of agents, according to their beliefs about their intentions. We may characterize an agent as **confident** if it believes that it will succeed in carrying out its intended actions.

$$\textbf{(Conf)} \qquad [t, Bel_i([t, Int_i\varphi \& Agent(\varphi, i)] \rightarrow \varphi)]. \tag{36}$$

The correspondence restriction on the model is as follows.

Proposition 14 A BDIG model that satisfies the condition

$$\textbf{(CConf)}\ \forall U \subseteq L, \{l' | U \notin It(l', \tau, a) \textit{ or } A(l', \tau, U) \neq a \textit{ or } l' \in U\} \in B(l, \tau, a) \tag{37}$$

validates axiom (CONF:36). □

For example, a confident agent may believe that if it intends to move block A on top of block B, then it will really do so, i.e.
$[t, Bel_i[t, Int_i([t_1, Do(i, move(A,B))]) \rightarrow [t_1, Do(i, move(A,B))]]]$. A confident agent which is omniscient (B1-B4) and aware of its intentions (IntB1) believes in its intended actions. Before presenting this proposition, we consider the agent's beliefs about the agent of a given action. We assume that an agent knows whether it is the agent of a proposition or not.

$$\textbf{(AGB1)} \qquad [t, Agent(\varphi, i)] \leftrightarrow [t, Bel_i[t, Agent\varphi]] \tag{38}$$

The correspondence restriction on the structure is as stated in the following proposition.

Proposition 15 A BDIG model that satisfies the condition

$$\textbf{(CAGB1)} \qquad A(l, \tau, U) = a \text{ iff } \{l' | A(l', \tau, U) = a\} \in B(l, \tau, a) \tag{39}$$

validates axiom (AGB1:38). □

We are now ready to present the proposition on confident agents.

Proposition 16 BDIG models that validates (INTB1: 32),(Conf:36), (AGB1:38) and (B1:16)-(B4:23) also validates

$$\textbf{(ConfB)} \qquad [t, Int_i\varphi] \& [t, Agent(\varphi, i)] \rightarrow [t, Bel_i\varphi]. \tag{40}$$

□

A confident agent that is not omniscient (i.e. does not satisfy axioms (B2:21)-(B4:23)) may not believe that its intended action will succeed. However, we can characterize such agents by imposing an additional restriction on It and B.

Proposition 17 A BDIG model that satisfies the condition

$$\textbf{(CConfB)} \text{ if } A(l,\tau,U)=a \text{ and } U\in It(l,\tau,a), \text{ then } U\in B(l,\tau,a) \quad (41)$$

validates axiom (ConfB:40). □

An agent that is sure that it will be able to satisfy all its intentions, including the ones that are not under its direct control, can be said to be **overconfident**.

$$\textbf{(OverConf)} \qquad [t, Bel_i([t, Int_i\varphi]\rightarrow\varphi)]. \qquad (42)$$

The correspondence restriction on the model is specified in the following proposition.

Proposition 18 A BDIG model that satisfies the condition

$$\textbf{(COver Conf)} \qquad \{l'|U\notin It(l',\tau,a) \text{ or } l'\in U\}\in B(l,\tau,a) \qquad (43)$$

validates axiom (OverConf:42). □

For example, an overconfident agent may believe that if it intends that block A will be on block B, then block A will actually be on block B, even if the agent is not capable of moving block A and does not intend to move it, i.e. $[t, Bel_i([t, Int_i[t_1, On(A,B)]]\rightarrow[t_1, On(A,B)])]$.

A proposition that is similar to Proposition 16 is true for an overconfident agent.

Proposition 19 BDIG models that validate (INTB1:32),(OverConf:42), and (B1:16)-(B4:23) also validate

$$\textbf{(OverConfB)} \qquad [t, Int_i\varphi]\rightarrow[t, Bel_i\varphi] \qquad (44)$$

□

An overconfident agent that is not omniscient may not believe in its intentions. However, (OverConfB:44) is sound in structures that have additional relations between It and B.

Proposition 20 A BDIG model that satisfies the condition

$$\textbf{(COverConfB)} \qquad \text{if } \; U \in It(l,\tau,a), \; \text{then} \; U \in B(l,\tau,a), \tag{45}$$

validates (OverConfB:44). $\square$

An omniscient agent which is overconfident will not adopt intentions that conflict according to its beliefs. That is, its beliefs about the side-effects of its intentions will influence its intentions. If $[t, Int_i\varphi]$, then i believes that φ. Therefore, if the agent is aware of some side-effects (e.g., $[t, Bel_i(\varphi \to \psi)]$), then it cannot intend the negation of the side-effects (e.g.,$[t, \neg Int_i \neg\psi]$).[19]

Proposition 21 BDIG models that validate (INTB1:32),(OverConf:42), (B1:16)-(B4:23) also validate

$$\textbf{(INTB3)} \qquad ([t, Int_i\varphi] \& [t, Bel_i(\varphi \to \psi)]) \to \neg[t, Int_i \neg\psi]. \tag{46}$$

$\square$

Proposition 21 is not true for non-omniscient agents. That is, the intentions of non-confident agents or (over)confident agents that are not omniscient may be contradictory according to their beliefs. Intuitively, this is because their beliefs are not closed under consequences and they do not realize that they believe in a contradiction, or because they do not check their intentions carefully, in light of their beliefs. This is quite common among humans. For example, a researcher may intend to finish a paper by a specific deadline, and believes that finishing the paper will prevent her from attending meetings on the day before the deadline. Nevertheless, she still intends to go to a faculty meeting on that day. To prevent such behavior, we may wish to restrict our attention to agents that satisfy axiom (INTB3:46), regardless of their reasoning power. This requires adding additional restrictions on the belief- and intention-accessibility relations.

Proposition 22 A BDIG model that satisfies the condition

$$\begin{aligned} \textbf{(CINTB3)} \;\; & \text{If } U_1 \in B(l,\tau,a), U_2, U_3 \in It(l,\tau,a) \;\; \text{and} \;\; U_1 \cap U_2 \neq \emptyset, \\ & \qquad\qquad \text{then} \;\; U_1 \cap U_2 \cap U_3 \neq \emptyset \end{aligned} \tag{47}$$

validates axiom (INTB3:46). $\square$

[19] Similarly, a confident agent will not adopt intentions that it believes contradict its own actions.

The above axioms and propositions put some constraints on when an agent can abandon its intentions. It is clear that according to axiom (INTB2:34), if an agent starts believing that one of its intentions is not possible, it must abandon this intention, i.e., $[t, Int_i\varphi]\&[t+1, Bel_i\neg\varphi] \rightarrow \neg[t+1, Int_i\varphi]$. Similarly Cohen and Levesque [16] also require that an agent will give up a persistent goal toward a proposition it considers impossible. However, Cohen and Levesque also assume that an agent will forgo its intention when it believes the intended proposition is true (unless it is a maintenance goal). Their requirement is based on the attribute of their system that there is no explicit time associated with their intended proposition (as we explained above). However, in our case an agent may believe that as a result of its own intended actions the intended proposition (that is associated with a specific time) is true. In particular, an overconfident agent always believes that its intentions are true. In such a case, it cannot forgo this intention. For example, suppose $t < t_1\&[t, Int_i[t_1, On(A,B)]]\&[t, Int_i([t_1-1, Do(i, Move(A,B))], [t_1, On(A,B)])]$ which intuitively means that, at time t, agent i intends that block A will be on block B at time t_1, and it intends to move A on top of B in order to achieve this intention. Suppose that agent i also believes that if it intends to do something, it will really do it, e.g., if it intends to move block A, it will really move it, and that moving block A on top of B will have the result that at the next time point $On(A,B)$ will be true. In such a case, agent i believes that, given its intentions and beliefs, $On(A,B)$ will be true at time t_1. However, its belief is based on its intended action to move A which is motivated by its intention that $On(A,B)$ will be true at time t_1. Thus, the agent must keep its intention $[t_1, On(A,B)]$ even though it believes $[t_1, On(A,B)]$ to be true.

Cohen and Levesque also require that an agent does not abandon a persistent goal unless it believes it is true or believes that it is impossible. We do not require that an agent keep its intention until it believes it is impossible. In our system an agent may abandon an intention due to other reasons, such as a request received from another agent or because of an observation. The ability for an agent to revise its intentions is very important in making argumentation effective. An agent may drop an intention because of a direct request from another agent, or because a request from another agent may conflict with the intention. Also, an agent may drop an intention as a result of a change in the environment. The commitment of an agent to its goals and intentions may greatly affect its performance and such a study is beyond the scope of this paper (see [68]). However, different types of agents may be characterized in our framework.

For example, an agent that does not change its intentions, unless it received a message from another agent may be characterized as follows:
if $\forall b \in \mathcal{A}gents, RECEIVE(l,\tau,a,b) = \emptyset$, then $It(l,\tau,a) = It(l,\tau-1,a)$.

It is also easy to see that if an agent adopts a new intention that it believes conflicts with its previous ones, it will drop the original intentions. This is a simple corollary of Axiom (INTB3:46).

Corollary 2.1 *BDIG models that validate Axiom (INTB3:46) also validate the following axiom*

$$\textbf{(INTB4)} \quad [t, Int_i\psi]\&[t+1, Int_i\varphi]\&[t+1, Bel_i(\varphi \rightarrow \neg\psi)]) \rightarrow \neg[t+1, Int_i\psi]. \quad (48)$$

Similarly, from Axiom (INTB1:32), it is easy to conclude that if an agent comes to believe that its intention is not possible, it will drop this intention.

Corollary 2.2 *BDIG models that validate Axiom (INTB2:34) also validate the following axiom*

$$\textbf{(INTB5)} \qquad [t, Int_i\psi]\&[t+1, Bel_i\neg\psi] \rightarrow \neg[t+1, Int_i\psi]. \qquad (49)$$

3 Axioms for Argumentation and for Argument Evaluation

The formal model can be used in two ways. One use is as a specification for agent design [56]. In this role, the model constrains certain planning and negotiation processes. It can also be used to check the agents' behavior. Another use of the model is by the agents themselves. In this section we demonstrate how the logic presented in the previous section can be used by a designer of an agent for the specification of the arguments it will use. In the next section we will describe an automated agent which uses our logic.

Arguments serve either to add an intention to the persuadee's set or to retract an intention or change the preferences of the persuadee. Below we present a list of several argument types which we use (a) to demonstrate the expressiveness of our logic, and (b) in the development of an automated negotiator. These argument types are not meant to constitute an exhaustive typology of arguments. Indeed, it has been pointed out [156] that it is not possible to present such an authoritative classification, since arguments must be interpreted and are effective within a particular context and domain. The six argument types that we present are ones that are commonly thought to have persuasive force in human negotiations [110,65,124]. Argumentations which were shown to be successful in human negotiation, may be also successful in automated agents' negotiations. Furthermore, we want our agents to be able to negotiate with humans, and therefore they need to be able to at least understand human argumentation. Moreover, the designers of the agents can follow the negotiation of the agents, if it is similar to human negotiation. The argument types we present are:

(1) Threats to produce goal adoption or goal abandonment on the part of the persuadee.
(2) Enticing the persuadee with a promise of a future reward.
(3) Appeal to past reward.
(4) Appeal to precedents as counterexamples to convey to the persuadee a contradiction between what she/he says and past actions.
(5) Appeal to "prevailing practice" to convey to the persuadee that the proposed action will further his/her goals since it has furthered others' goals in the past.
(6) Appeal to self-interest to convince a persuadee that taking this action will enable achievement of a high-importance goal.

Threats and promises are the most common arguments used in human negotiations [11]. An appeal to prevailing practice is the most common argument used in the legal system. Furthermore, it was found that presenting example instances (prevailing practice cases) is much more persuasive than presenting statistical summaries [70,151,107,61]. An "appeal to past promise" is supported by the cognitive dissonance theory [110] that assumes that a person seeks to maximize the internal psychological consistency of his/her cognition, and thus will be willing to keep his/her promises. This argument is also important in repeated interactions since agents prefer to maintain their credibility. The other two arguments, "an appeal to self interest" and "a counter example" are examples of arguments useful to persuade bounded rational agents which have limited inferential resources.

Each of the above arguments will be discussed in the following subsections. For each type we present examples that are borrowed from human argumentation, as well as examples of automated agents' interactions. Axioms for creation of such arguments will also be presented. Examples of argumentation among automated agents presented below are based on the following scenario. Agents with different spheres of expertise may need to negotiate with each other for the sake of requesting each others' services. Their expertise is also their bargaining power. As an example, consider a robot R_e who has a better "eye" (has a powerful camera) while R_h has a better "hand"(has skilled functions with dextrous fingers enabling it to isolate mineral chunks). Yet another agent R_m has specialized maps and terrain knowledge and is adroit at navigation. Imagine these three self-motivated robots with goals to obtain samples from Mars. R_e, R_h and R_m are looking for different mineral samples. We can imagine these three agents facing the need to argue with each other. [20]

[20] Similar needs for argumentation can be envisaged when automated agents seek different goals at the site of a nuclear accident, or at the site of a forest fire.

Suppose agent j intends that agent i should do α at time $\bar{t}$ and i refuses. Based on its own beliefs, j assumes that i refused to do α probably since α contradicts one of i's goals or intentions. If there is an action β that j can perform, that contradicts (as per j's beliefs) another goal of i, and this last goal is preferred by i (again according to j beliefs) over the first one, j threatens i that it will do β if i will not do α. This type of argument may appear in several different forms. For example, suppose agent j intends that agent i shouldn't do α, at time $\bar{t}$, and i insists to maintain its intention of doing α. Here, agent j threatens i that it will do β if i will *do* α.

A labor union insists on a wage increase. The management says it cannot afford it, and asks the union to withdraw its request. The management threatens that, if it grants this increase, it will have to lay off employees to compensate for the higher operational cost that the increase will entail. The outcome (i.e. whether the union succumbs to the threat or not) depends on the union's preferences. If preserving employment is more important than wage increases, the union will accept the argument (assuming it believes that the management will carry out the threat). If a wage increase is more important, then the union will not accept the argument and insist on a wage increase (here, whether or not it believes the management will carry out its threat is irrelevant to the union's decision.)

One of the questions related to generating a threat is: how does j choose β? If j wants the threat to be effective, carrying out β should be painful for i and conflict with one of its goals or intentions (as we stated above). However, the threat should be credible according to i's beliefs (see our discussion concerning the evaluation of threats below). First of all, doing β should be within the power of j (at least in i's view). Furthermore, usually, carrying out a threat may contradict some of j's intentions or goals. These intentions and goals should be less preferred by j than the goal that α contributes to (again, at least in i's view).

There may be several such βs that j may choose from. The β that is chosen depends on whether the persuader, j, wants to inflict a very strong threat (i.e., a β which contradicts a preferred goal or intention of i), or to start with a weaker threat (i.e., one which will contradict a less preferred goal of i) and, if i refuses it, to escalate with stronger threats (wearing i down). Argument evaluation is an important aspect of argumentation as discussed in the next section.

Here is an axiom scheme specifying the generation of a threat argument in the logic presented in Section 2.[21]

$$
\begin{aligned}
&(\forall t_1, t_2, t_3, t_4, \bar{t}, i, j, \alpha, \beta)\{\\
&\quad t_1 < t_2 < t_3 < \bar{t} < t_4 \;\&\; i \neq j \qquad\qquad \& \;[t_1, Send_{ji}Request([\bar{t}, Do(i,\alpha)])]\\
&\& \;[t_2, Receive_{ji}Reject([\bar{t}, Do(i,\alpha)]))] \;\&\; [t_3, Bel_j([t_3, Goal_i[\bar{t}, g_1]\&Goal_i[t_4, g_2]])]\\
&\& \;[t_3, Bel_j([t_3, Pref_i([t_4, g_2], [\bar{t}, g_1])])] \;\&\; [t_3, Bel_j[\bar{t}, Do(i,\alpha) \rightarrow \neg g_1]]\&[t_4, Do(j,\beta) \rightarrow \neg g_2]\\
&\& \;[t_3, Bel_j[t_3, Cred(\beta, \alpha, i,)]] \qquad\qquad \& \;[t_3, Bel_j[t_3, Appropriate(\beta, \alpha, i)]]\\
&\qquad\qquad \rightarrow [t_3, Send_{ji}Request([\bar{t}, Do(i,\alpha)], \neg[\bar{t}, Do(i,\alpha)] \rightarrow [t_4, Do(j,\beta)])]\}.
\end{aligned}
$$

$Cred$ is a meta-predicate which stands for an axiom that j will use for estimating whether β is a credible threat for i to do α. $Appropriate$ is a meta-predicate which stands for axioms that will specify how to choose β, when several such βs exist.

In the example of the robots on Mars agent R_h must explore in a dimly lit area while digging for its mineral. Some help from R_e in scanning the area with a high resolution camera would greatly contribute towards this goal. R_h requests from R_e the use of its camera. R_e refuses, since the time spent in furthering R_h's goals will interfere with its own goal– to dig for its own mineral. R_h then threatens that it will smash R_e's camera lens if R_e does not accede to this request.

3.2 Evaluation of Threats

In this section we demonstrate factors affecting the evaluation of a threat. Since we do not assume that agents are honest, the main problem in the evaluation is how to decide whether the threatening agent will carry out its threat. Usually, executing a threat will affect the agent that threatens to carry it out, and this has a bearing on the evaluation.

Suppose j had requested i to do α at a given time point $\bar{t}$, and it had threatened i that if it does not do α, j would do β. Now, i should consider several issues. First of all, how bad is the threat? If α contradicts one of i's goals and β contradicts another goal g, which goal does i prefer? But then again, i should evaluate whether j will carry out its threat at all. We may assume that β has negative side-effects on j as well. The question is whether j prefers the benefit

[21] All the axioms listed in this section are only suggested possibilities for producing the relevant argumentations. We demonstrate the expressibility of the logic in formalizing such axioms.

for itself from α over the losses resulting from the side-effects of β in case it carries out the threat. Another issue relevant here is how important is it for j to preserve its credibility and reputation. Another issue for i to consider is whether the threat is a *bounded* threat. A bounded threat is always credible since i is aware of prior arrangements made by j to execute the threat should i default. Usually, j will convey this information to i in a prior exchange. If i believes that j may carry out its threat and decides that it is worthwhile for it to do α, it still needs to update its goals and intentions. Here i will intend α in order to contribute to preventing j from doing β which contradicts g. Note that here i intends α, without β being a goal. Furthermore, since any goal is also an intention (GINT), i should abandon the goal that α contradicts, as well as the related intentions.

$$
\begin{aligned}
&(\forall t_1, t_2, t_3, \bar{t}, \alpha, \beta, i, j)\{t_1 < t_2 < \bar{t} < t_3 \ \&\ i \neq j \\
&\quad \& \quad [t_1, Receive_{ji}Request([\bar{t}, Do(i,\alpha)], \neg[\bar{t}, Do(i,\alpha)] \rightarrow [t_3, Do(j,\beta)])] \\
&\quad \& \ [t_2, Bel_i[\bar{t}, Do(i,\alpha) \rightarrow (\neg g_1^i \& g_2^j)]] \ \& \quad [t_2, Bel_i[\bar{t}, [Do(j,\beta) \rightarrow (\neg g_3^i \& \neg g_4^j)]] \\
&\quad \& \ [t_2, Goal_i[\bar{t}, g_1^i] \& Goal_i[\bar{t}, g_3^i]] \quad \& \quad [t_2, Pref_i([\bar{t}, g_3^i], [\bar{t}, g_1^i])] \\
&\quad \& \ [t_2, Bel_i[t_2, Goal_j[\bar{t}, g_2^j]]] \\
&\quad \& \ [t_2, Bel_i[t_2, \neg Goal_i[\bar{t}, g_4^j] \vee (Goal_i[\bar{t}, g_4^j] \& Pref_j([\bar{t}, g_2^j], [\bar{t}, g_4^j]))]] \\
&\rightarrow [t_2, Int_i([\bar{t}, Do(i,\alpha)], [t_3, \neg Do(j,\beta)]) \& Int_i([t_3, \neg Do(j,\beta)], g_3^i) \& \neg Goal_i[\bar{t}, g_1^i]] \ \& \\
&\qquad [t_2, Send_{ij}Accept([\bar{t}, Do(i,\alpha)])]\}.
\end{aligned}
$$

In this axiom we have listed one way to evaluate whether a threat is credible. Here, i believes that if j carries out threat β, this will contradict i's goal (g_3^i) as well as some possible goals of j (g_4^j). If i believes that g_4^j is one of j's goals, and if it believes that j prefers the goal that α contributes to over g_4^j then it will believe that β is a credible threat. A similar axiom can be specified which considers cases where j asks i not to do α.

3.3 Promise of a future reward

Agent j entices agent i to do action α (alternately, avoid doing α) at time t by offering to do an action β at a future time, as a reward. Agent j believes β to contribute to the desires of i.

An example is a sales agent trying to persuade a customer to buy a VCR by offering a free servicing plan and a set of blank cassettes.

$$
\begin{aligned}
&(\forall t_1, t_2, t_3, t_4, \bar{t}, i, j, \alpha, \beta)\{\\
&\quad t_1 < t_2 < t_3 < \bar{t} < t_4 \;\&\; i \neq j \qquad \& \; [t_1, Send_{ji}Request([\bar{t}, Do(i,\alpha)])]\\
&\;\& \; [t_2, Receive_{ji}Reject([\bar{t}, Do(i,\alpha)]))] \;\&\; [t_3, Bel_j([t_3, Goal_i[\bar{t}, g_1]\&Goal_i[t_4, g_2]])]\\
&\;\& \; [t_3, Bel_j([t_3, Pref([\bar{t}, g_2], [t_4, g_1])])] \;\&\; [t_3, Bel_j[\bar{t}, Do(i,\alpha) \rightarrow \neg g_1]\&[t_4, Do(j,\beta) \rightarrow g_2]]\\
&\;\& \; [t_3, Bel_j \; [t_3, Cred(\beta, \alpha, i,)]] \qquad \& \; [t_3, Bel_j[t_3, Appropriate(\beta, \alpha, i)]]\\
&\qquad\qquad \rightarrow [t_3, Send_{ji}Request([\bar{t}, Do(i,\alpha)], [\bar{t}, Do(i,\alpha)] \rightarrow [t_4, Do(j,\beta)])]\}.
\end{aligned}
$$

Consider the scenario described in the threat example above involving robots. Instead of responding with a threat, R_h could offer to contribute towards R_e's goal by helping it to isolate its samples from the debris better by using its sorting skills by means of skilled fingers. This would reduce the weight of the samples that R_e now plans to collect, and greatly increase the ratio of mineral to debris for R_e.

3.4 Appeal to past promise

In this case agent j had requested i to do an action α based on a past promise. If i refuses, j reminds him of the past promise. For example, a child has promised her parent to clean the house, in order to convince them to buy her something. When she is later asked by the parents to clean the house and refuses, the parents may remind her of the promise.

$$
\begin{aligned}
&(\forall t_0, t_1, t_2, t_3, t_0', \bar{t}, i, j, \alpha, \beta)\{ \quad t_0 < t_1 < t_2 < t_3 < \bar{t} < t_4 \& i \neq j\\
&\;\& \quad [t_1, Send_{ji}Request([\bar{t}, Do(i,\alpha)])]\\
&\;\& \quad [t_2, Receive_{ji}Reject([\bar{t}, Do(i,\alpha)]))]\\
&\;\& \quad [t_3, Bel_j[t_0, Received_{ji}Request([t_0', Do(j,\beta)], [t_0', Do(j,\beta)] \rightarrow [\bar{t}, Do(i,\alpha)])]]\\
&\;\rightarrow [t_3, Send_{ji}Request([\bar{t}, Do(i,\alpha)], [t_0, Send_{ij}Request([t_0', Do(j,\beta)], [t_0', Do(j,\beta)] \rightarrow [\bar{t}, Do(i,\alpha)])]])]\}.
\end{aligned}
$$

For example, if as in the previous case, R_h offered to contribute to R_e's goal by helping it to isolate its samples later, when the time for sample isolation arrives, R_e may use this argument as an argument for a request from R_h to help it in isolating samples.

Here, agent j had intended that i do α at time $\bar{t}$, requested it from i, but i refused. Now j believes that the reason i refused is that α contradicts one of its goals or intentions. However, j believes, that in the past, i had done another action β that also contradicted the same goal or similar intention, and brings it up as a counterexample.

As an example, consider a parent trying to persuade a teenager to stay up until midnight to study for an exam. The teenager refuses on the grounds that she may suffer bad health from staying up late. The parent points out that the teenager had stayed up until 2 a.m. for a party the previous week, without suffering any ill-effects, and brings it up as a counterexample to the argument.

$$
\begin{aligned}
&(\forall t', t_1, t_2, t_3, \bar{t}, i, j, \alpha, \beta)\{\ t' < t_1 < t_2 < t_3 < \bar{t} \& i \neq j \quad \& \\
&\quad [t_1, Send_{ji}Request([\bar{t}, Do(i,\alpha)])] \quad \& \\
&\quad [t_2, Receive_{ji}Reject([\bar{t}, Do(i,\alpha)], [\bar{t}, Do(i,\alpha) \rightarrow \neg g])] \quad \& \\
&\quad [t_3, Bel_j([t', Do(i,\beta) \& Do(i,\beta) \rightarrow \neg g])] \\
&\qquad\qquad \rightarrow [t3, Send_{ji}Request([\bar{t}, Do(i,\alpha)], [t', Do(i,\beta) \& Do(i,\beta) \rightarrow \neg g])]\}.
\end{aligned}
$$

The following is an example from the robots on Mars. Suppose, R_h requests R_m to survey the terrain using its navigation skills. R_m's temperature sensors indicate that in some areas there may be high temperature pockets and these may harm its electronic circuitry. R_m refuses on these grounds. R_h points out that in the past two days, R_m has withstood much higher temperatures created during the explosions used in the digging process, without any evidence of harm to its circuitry. R_h brings this up as a counterexample to convince R_m to undertake the survey.

3.6 Appeal to "Prevailing Practice"

In this case, j receives a refusal from i to do α on the grounds that it contradicts goal g of i. If j believes that another agent k had done the same α and it did not contradict the same goal g held by k at the time, it uses it as an argument. For example, a teacher intends that a student talented in baseball should stay after school for extra curricular activity. This will contribute to the teacher's desire to build a good baseball team at school. He asks the student to do so, but the student refuses on the grounds that this will adversely affect his academic performance. The teacher points out that last year the star baseball player of the class was also an "A" student, and that several good players are also good students, and encourages the student to take up the activity.

$$(\forall t', t, t_1, t_2, t_3, \bar{t}, i, j, h, \alpha)\{t' < t_1 < t_2 < t_3 < \bar{t} \;\&\; i \neq j \neq h$$
$$\& \; [t_1, Send_{ji}Request([\bar{t}, Do(i,\alpha)])]$$
$$\& \; [t_2, Receive_{ji}Reject([\bar{t}, Do(i,\alpha)], [\bar{t}, Do(i,\alpha) \rightarrow \neg g])]$$
$$\& \; [t_3, Bel_j([t', Do(h,\alpha) \& \neg(Do(h,\alpha) \rightarrow \neg g)])]$$
$$\rightarrow [t3, Send_{ji}Request([\bar{t}, Do(i,\alpha)], [t', Do(h,\alpha) \& \neg(Do(h,\alpha) \rightarrow \neg g)])]\}.$$

With the robots on Mars, consider the following mention of prevailing practice in an argument. As in the counterexample scenario, R_h requests R_m to survey the terrain using its navigation skills. R_m's temperature sensors indicate that in some areas there may be high temperature pockets and these may bring harm to its electronic circuitry. R_m refuses on these grounds. R_h points out that both R_e and itself were exposed to much higher temperatures two days ago, and had withstood them quite well.

3.7 Appeal to Self Interest

In this case j believes that α implies one of i's desires and uses it as an argument. This is a useful argument when j believes that i is not aware of the implication. For example, an employee has a goal to study Japanese, but wants to save money as well. She intends for her company to pay for the Japanese lessons and asks the company. The company refuses. The employee points out that having an employee with knowledge of Japanese is a great asset to the company, especially in the coming years when the company will face stiff competition from the Japanese.

In another setting, agent j requests agent i to give up its intention to do α by pointing out that either α or its side-effect β contradicts one of i's desires.

$$(\forall t_1, t_2, t_3, \bar{t}, i, j, k)\{t_1 < t_2 < t_3 < \bar{t} \;\&\; i \neq j$$
$$\& \; [t_1, Send_{ji}Request([\bar{t}, Do(i,\alpha)])]$$
$$\& \; [t_2, Receive_{ji}Reject([\bar{t}, Do(i,\alpha)]))] \;\& \quad [t_3, Bel_j([\bar{t}, Desire_i(d_1, p)])]$$
$$\& \; [t_3, Bel_j([\bar{t}, Do(i,\alpha) \rightarrow d_1])]$$
$$\rightarrow [t3, Send_{ji}Request([\bar{t}, Do(i,\alpha)], [\bar{t}, Do(i,\alpha) \rightarrow d_1 \& Desire_i(d_1, p)])]\}.$$

For example, suppose R_e and R_h both plan to dig at site X on Tuesday. If they both dig at the same site, clearly, there will be destructive interference, leading to a malfunctioning of the procedures of either. R_e makes a proposal to R_h to divide their digging time so that R_e digs in the morning while R_h digs in the evening. R_h refuses, since obviously, this proposal reduces its product. However, R_e points out, that if R_h refuses, it will result in near zero product for R_h and R_e. Instead, sharing the time will further R_h's self-interest much better, since getting half the work done is better than getting no work done.

3.8 Selecting Arguments by an Agent's Type

A persuading agent, faced with a particular situation, must decide which argument to use. In order for an argument to be effective, it must address beliefs and intentions of the persuadee. Therefore, the beliefs of an agent about the persuadee can be used as guidelines for argument generation. One type of belief pertaining to a persuadee is the type of agent the persuadee is believed to be. This can influence the argumentation.

For example, non-bounded threats or future rewards are not applicable if an agent is memoryless [22] . Suppose agent j asks a memoryless agent k to do α and threatens it with β. Let us assume that β is expensive to j. If k will not do α, there is no benefit to j to carry out the threat (i.e., do β). The only reason that j may do β is to maintain its credibility. However, if agent k doesn't have a memory of past encounters, there is no notion of credibility. In future encounters, k will not remember whether j carried out its past threats or not. But then again, if it is clear that j will not carry out its threat (or will keep its promise for future reward), there is no sense in making threats to begin with. It seems that in case of memoryless agents only bounded threats or rewards are applicable.

On the other hand, the counterexample argument is appropriate in the case of a memoryless agent. In this case the agent doesn't remember the counterexample, and the purpose of the argument is to bring it to its notice. Of course, a "memoryless" agent may evaluate a counterexample type argument as non-credible exactly because the agent is memoryless.

Counterexamples may also be useful as an argument for an agent that is not omniscient. This agent may not have realized in the past that its action contradicted its goal. However, the non-omniscient agent may respond with a counter-argument that had it realized the implication, it wouldn't have taken

[22] Note, that this discussion is specific to automated agents. Human negotiators always have at least some memory of the past.

the action in the past either.

Appeal to self-interest is more appropriate in cases where the agent is not omniscient or in cases when the agent's beliefs are incomplete. In both situations the agent may not be aware of its self-interest, and such an argument may change its intentions.

It seems that the arguments used among cooperative agents would tend to be different from those used among non-cooperative self-motivated agents. Here, we have concentrated mainly on arguments that are appropriate in non-cooperative environments. All the arguments consider goals and desires of self-motivated agents. In a cooperative environment additional arguments may be made, such as "appeal to a universal principle" (e.g., fairness) or "appeal to authority" [148]. There is no sense in talking about fairness with a "self-motivated" agent. However, cooperative agents that were built by the same designer, or that belong to the same organization, may be influenced by such an argument. On the other hand, threats seem inappropriate in cooperative environments. If the agents have common goals, standing in the way of another agent may cause damage to both agents.

3.9 An example: Labor Union vs. Management Negotiation

We will consider the labor union example we presented in Section 3.1. The union insists on a wage increase. The management says it cannot afford it, and asks the union to withdraw its request. The management claims that if it grants this increase, it will have to lay off employees to compensate for the higher operational cost that the increase will entail.

We will simplify the issues and describe only part of the negotiation process here. Let us suppose that it is October 2nd, the union (u) has the goals to have an increase in wages (*wage.increase*) become effective December 1, and also to prevent causing unemployment ($\neg$*unemployment*). We assume that the management (m) wants to save money (*save*). We also assume that in the exchange that took place between the union and the management on the previous day, i.e. on October 1, the union had received a message from the management requesting it to call off asking for a December 1st wage increase (*ask.wage.increase*), with a threat that, otherwise, the management will lay off employees (*lay.off*). Evidently, if there will not be a request for a wage increase on December 1, then there will not be a wage increase and the management will save money. On the other hand, laying off employees will cause unemployment. We also assume that the union believes that the management has a goal to save money on December 1, but it doesn't have the goal to preserve employment. The union prefers to preserve employment over obtaining a wage increase.

From these assumptions, one can conclude, using the evaluation of the threats axioms, that the union will intend not to ask for a wage increase. The union will revise its goals, and will send the management an appropriate message. Note, that the union may still continue to hold the desire for a wage increase.

Prior to Oct 1 (in the recent past): Union requests a wage increase.

Oct 1 (yesterday):
$[Oct1, Receive_{um} Request([Dec1, \neg Do(u, ask.wage.increase)],$
$\quad [Dec1, Do(u, ask.wage.increase)] \to [Dec1, Do(m, lay.off)])]$

Oct 2 (today): *At start of argument:*
$[Oct2, Goal_u[Dec1, wage.increase]\&Goal_u[Dec1, \neg unemployment]]$
$[Oct2, Goal_m[Dec1, save]]$
$[Oct2, Bel_u[Oct2, Goal_m[Dec1, save]\&\neg Goal_m[Dec1, \neg unemployment]]]$
$[Oct2, Pref_u([Dec1, \neg unemployment], [Dec1, wage.increase])]$
$[Oct2, Bel_u[Dec1, \neg Do(u, ask.wage.increase) \to \neg wage.increase\&save]]$
$[Oct2, Bel_u[Dec1, Do(u, ask.wage.increase) \to wage.increase\&\neg save]]$
$[Oct2, Bel_u[Dec1, Do(m, lay.off) \to unemployment]]$

On evaluation of argument:
$[Oct2, Int_u([Dec1, \neg Do(u, ask.wage.increase)], [Dec1, \neg Do(m, lay.off)]]$
$\neg[Oct2, Goal_u[Dec1, wage.increase]].$

3.10 Contract Net Example

Assume a contract net kind of situation [142] where agents make their bids independently, but also communicate to resolve any conflicts, in case two agents a and b opt to do a particular task. An argument from one agent to another (or of an arbitrator agent to one of the contestants) could be: agent a says to agent b, if you insist on doing the task ($task.b$), then the overall system goal will suffer (e.g., the overall task will not be on time ($ontime$)). Formally, the following message will be sent, using the axiom "Appeal to self interest":
$[t, Send_{ab}(Request([t', \neg Do(b, task.b)], [t', task.b \to \neg ontime]))]$.

We note that there is a difference between this example and the previous one in that agent a is not threatening to *do* an action such as delaying the overall task, on account of being upset over b doing the task. Rather, agent a simply explains to agent b, the deleterious consequences of agent b's action that presumably agent b could not deduce itself (due to lack of knowledge or inferential power). This is a case of appealing to self interest.

There is also a similarity to the previous examples in that, if agent b prefers his goal of the "overall task being on time" than doing the particular subtask itself (which could bring him some reward), then agent b will accept agent a's argument (assuming b trusts a).

4 Automated Negotiation Agent (ANA)

In the previous section we demonstrated how the formal model can be used as a specification for agent design. In particular, we demonstrated how axioms specifying the generation of arguments can be expressed. Another use of the formal model, which we present in this section, is that the agents themselves use the formal model and the axioms. For example, if agent i derives "$Do(\alpha, i)$", it would try to perform α. Similarly, the agents would use axioms and rules to evaluate messages, to send arguments and to update their knowledge bases. In order to demonstrate this aspect of our formal model, we implemented an automated negotiator agent (ANA) that acts in a simulated environment. [23] The simulation system complies with the definition of an Agent Oriented Programming (AOP) system [138]. This term denotes several ideas: (i) The agent is represented using notions of mental states; (ii) The agent's actions depend on these mental states; (iii) The agent's mental state may change over time; (iv) Mental state changes are driven by inference rules.

ANA gives an additional layer of design flexibility. The system infrastructure gives the user the ability to define the agents and to set a different mechanism as its mental state engine for each agent. Therefore, the user can control the agents' mental state behavior. In addition, the user can test different methods of negotiation between the agents and evaluate the effectiveness of various arguments in the negotiation. We demonstrate the properties of the system in a Blocks World environment. The ability to provide infrastructure for controlling agent's mental state behavior could be useful for constructing "believable agents" [7,62].

4.1 The structure of an agent and its life cycle

The general structure of an agent consists of the following main parts (see Figure 1):

- Mental state (beliefs, desires, goals, intentions).
- Characteristics (agent type, capabilities, belief verification capabilities).

[23] ANA was implemented in LPA Prolog for Windows. See [37,144] for a user guide.

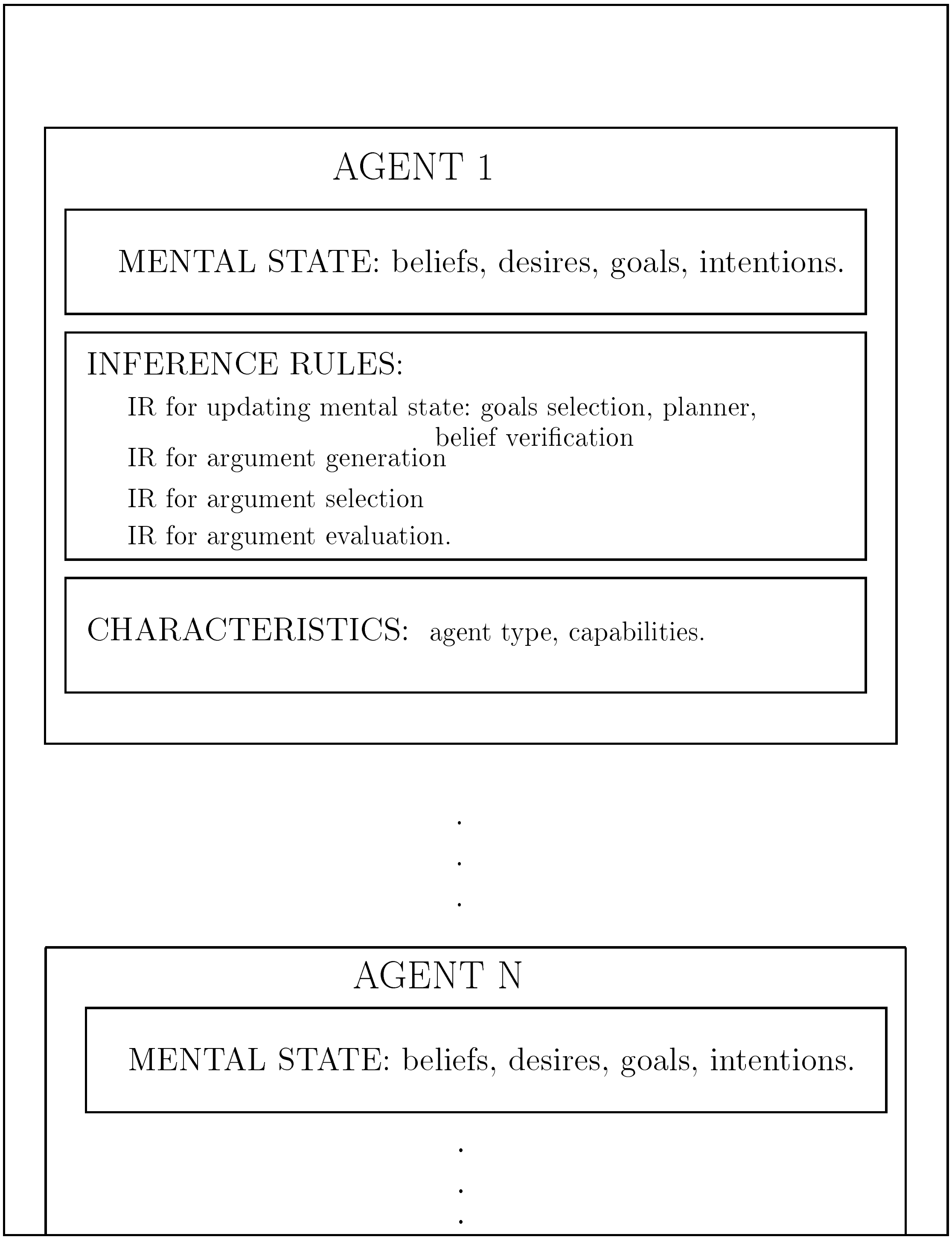

Fig. 1. ANA Structure.

- Inference rules (mental state update, argument generation, argument selection, request evaluation).

The designer of a specific agent can influence its general behavior by providing the following input at agent creation time:

(1) Mental state update rules: belief verification rules, goal selection rules and intention generation mechanism (the planner of actions).
(2) Argument generation and selection mechanism.
(3) Request evaluation rules.

In addition, when the user creates an agent, he/she assigns it initial beliefs and desires. The general inference rules and the specific beliefs and desires are the driving force behind the agent behavior. Once an agent is created, it operates in a loop (like an operating system). We call this loop the agent *life cycle*.

The agent life cycle includes the following steps:

(1) Generating goals for the current time interval.
(2) Generating intentions for selected goals.
(3) Performing all possible intentions-to.
(4) Generating requests (with or without arguments) for an intention-that.
(5) Selecting one best request from those generated.
(6) Sending that request.
(7) Reading incoming messages.
(8) Evaluating incoming messages.
(9) Updating beliefs, goals and intentions.
(10) Responding to incoming messages.
(11) Continuing from step 3.

As we said above, when the user creates an agent, he/she assigns it initial beliefs and desires. The agent associates a utility with each of its desires and uses its ranked desires and its beliefs to select its goals. These are the initial goals for the current time interval. From these goals the agent generates its intentions for the current time interval using a planning mechanism. Each of these intentions is categorized as *intention-that* or *intention-to*, according to the capabilities or lack of capabilities of the agent to perform the actions defined in the intentions.

The generation of an agent's intentions completes the first stage in the agent's life cycle. The agent will try to fulfill each and every one of its intentions. If there are intentions that should be performed and can be performed by the agent, then it will execute them, unless these actions can be used by the agent in future negotiations as a bargaining card. If the next intention to be performed cannot be executed by the agent (it is an intention-that), then the agent will generate a request message to one of the other agents that, according to the agent's beliefs, can execute it. The agent will now wait for a response. Note that the agent will remember that it issued the request. Later on, the agent will be able to use this information.[24]

According to the response, the agent will continue handling other intentions or will have to generate another request, this time using some type of reasoning. Actually, the two agents will start negotiation over this request. If argumentation is needed, the agent will generate all kinds of arguments that can be used in the negotiation. It will then select one of these arguments, which it determines to be most suitable, and will send it in a message to that other agent. Again, the agent will wait for a response. If the response is positive, the agent will wait for its execution and continue from that point on. Otherwise, the agent will try to generate a more persuasive argument and use it in a new request. This cycle will continue until there are no more arguments to be used, in which case, the agent will have to revise its intentions and possibly its goals.

This is a typical scenario for an agent life cycle. However, there might be cases in which the agent will generate its goals from scratch. Assume that a second agent issues a request for help which contradicts one of the agent's goals. If as a consequence of some argument, e.g., a threat, the first agent agrees to help the second agent, then the agent should select a new set of goals from its desires. This time, the selected set of goals will not contradict that second agent's request.

As we discussed in the theoretical section of this paper, time increments play a major role in the process of negotiation. As time passes, agents become more aware of their environment and can reason about each other's credibility. In the implementation, we divide the time line into several intervals. The number of time intervals is not known to the agents. We believe that the stages of negotiation are more relevant to argumentation than the length of each negotiation phase. We make this restriction, under the assumption that the agents have sufficient time to negotiate efficiently and take action in the same time interval. However, during negotiation, the agents can take into consideration past and future time intervals. This is based on the agents' ability to remember past events.

Thus, assume that during the negotiation phase of the first time interval, the time interval expires. The agent will have to generate its goals and intentions for the next time interval and start all over again. As mentioned above, starting a new time interval is not the same as starting the whole scenario from scratch. This time the agent has more information and knowledge about other agents and about the world. Moreover, the agent usually remembers previous promises, threats, and actions. Now, it will be better able to evaluate requests

[24] Given the general framework described in this paper, more sophisticated agents can be constructed (see [144]). Here we describe the simplest agent which we developed.

in the new time interval. Promises that were not fulfilled will damage the other agents' credibility in the eyes of the agent. The next time the agent will have to consider such a promise, it will not be that naive. The idea is to be able to conduct a more efficient negotiation as time progresses.

We have described the life cycle of the agents and their capabilities in conducting negotiation with each other. Note that our model places no restrictions on the agents' behavior during the negotiation. The logical negotiation protocol between the agents has a loose definition, as it adapts itself to the situation in which the agents are operating. An agent has no obligation to respond to any of the messages it receives. On the other hand, an agent can send as many requests as it wishes for one action. Although the model does not impose the restriction that an agent has to wait for a response before sending any new requests, an agent presumably will want to wait for and evaluate how the reply to a response fits into its current plan before sending another request. In our implementation we allow the agent to send one argument in each message. It is possible to relax this restriction, but it will require the development of rules for deciding when to send more than one argument, which arguments to send in such situations, and how to evaluate messages which consist of more than one argument. However, sending one argument at a time is considered a good negotiation policy [165]: arguments reveal information about the sender and may lead to commitments (e.g., the need for keeping promises). Also, in our environment communications take significantly less time than computations. In particular, finding a good argument is complex, and thus, it is worth trying one argument and only if the opponent is not convinced, more time should be spent on finding and selecting a better argument, which can be sent in a new message.

4.2 Inference Rules for Mental State Changes

As described earlier, every agent has inference rules, provided by its designer, for changing and updating its mental states according to various inputs. The details of the syntax of these rules can be found in [37]. In the following subsections, we present the form of these rules and the general rules we implemented which specify the behavior the agents exhibit. We also demonstrate the instantiation of theses rules in the Blocks World environment in Section 4.6 and Section 4.7.

4.2.1 Belief Verification Rules

The agent is assigned an initial set of beliefs by the user of the system. These beliefs change over time, according to new information the agent receives from the environment or from other agents. The agent uses its belief verification rules to conclude a new belief according to existing beliefs. The syntax of these rules is as follows:

$$agent\_believes(AgentName, Belief, TimeInterval, TruthValue) : -Condition$$

where *AgentName* is the name of the agent. *Belief* is any data, information or a fact statement that is being generated (or evaluated) by the inference rule. $TimeInterval$ is the time interval for which the Belief's $TruthValue$ is relevant. $TruthValue$ is the truth value that is being generated by the inference rule. $Condition$ is the body of the inference rule (Prolog clause). Once the $Condition$ is found to be true, then the agent considers the belief to be correct or incorrect at time $TimeInterval$, according to the $TruthValue$ generated by the Condition.

Currently, we have defined and implemented two types of negotiating agents, *reasonable* and *knowledgeable*, that are omniscient. In the implementation of these agents we distinguish between the agent's explicit and implicit beliefs [84]. The agent's explicit beliefs are the ones that are specified in its knowledge base. The implicit beliefs of a reasonable or knowledgeable agent are the beliefs it can conclude using its inference rules. That is, the reasonable agent's implicit beliefs satisfy axioms (B1:16)–(B4:23) of Section 2.4 and the beliefs of a knowledgeable agent also satisfy axiom (B5:24). The agent's beliefs are changing over time. However if the agent believes that a sentence is true, it will not change this truth value unless it receives some new relevant information.

In addition, at any given time point, we make the closed world assumption, i.e. an agent always implicitly believes a sentence or its negation, but this implicit belief may change over time. [25] When a reasonable or knowledgeable agent needs to check whether it believes a sentence to be true at a given time point, it will try to infer this sentence with respect to the given time point. If it fails, it will conclude that this sentence is false and that its negation is true. However, when considering a different time period, it may reach a different conclusion. Furthermore, the agent can revise its past and future beliefs after observations or after evaluating messages received from another agent. For example, suppose, during the negotiation, a reasonable agent needs to find out whether its opponent desires, at time t_1, that block A will be on block B at time t_2. It will check its knowledge-base and if this desire of the opponent is not explicitly stated and it cannot be inferred from other explicit beliefs, it will conclude that its opponent does not desire that A will be on B at the relevant time. However, if, for example, later its opponent asks the agent to put A on B, the agent will revise its beliefs.

[25] The appropriate semantic restriction on the structure of Section 2.3 is the following: $\forall l \in L, a \in \mathcal{A}gents, \tau \in T, U \subseteq L, (U \in B(l,\tau,a) \ or \ L \setminus U \in B(l,\tau,a))$.

4.2.2 Goal Selection

An agent's desires may conflict with one another (see Section 2.3.1.) When a new time interval is reached (or just after the agent has been created), the agent will look for a subset of its desires in order to set these as its goals for the specific time interval. In order to do so, the agent may use the preference values that is assigned to each and every desire. The selected desires (called goals) should not conflict with each other. The syntax for goal generating rules is as follows:

$$generate\_goals(AgentName, ListOfGoals,$$
$$TimeInterval/AgentName/GoalState/GoalsTimeInterval) : -Condition.$$

where $AgentName$ is the name of the agent; $ListOfGoals$ is the list of the agent's goals, which is a subset of its desires; $TimeInterval$ is the current time interval; *GoalState* is a different representation of the selected goals. This representation depends on the world environment example as it is used by the user. This will later be used to generate the agent's intentions. $GoalsTimeInterval$ is the time interval in which the goals being selected should be achieved. *Condition* is the body of the rule which defines the way to generate the agent's $ListOfGoals$.

In our case example of the Blocks World environment, we used the simplest way to generate the agent's goals. We generated all possible subsets of the agent's non-conflicting desires. Then we selected the subset which has the maximum sum of preference value for all of the agent's desires, regardless of the number of desires in that subset.

4.2.3 Intention Generation – Planning

A set of rules should be assigned to an agent to be used to generate intentions for the agent's selected goals. Performance of the intended actions will lead to satisfying the agent's goal.

$$search(AgentName, GoalState, Path, Intentions) : -Condition.$$

where $AgentName$ is the name of the agent. *GoalState* is the output from the goal generation rule that was previously executed by the agent. $Path$ is a set of steps that lead to the intentions. $Intentions$ is a list of actions that are to be executed. Each action is accompanied by its generating cause. The following structure is formed: Action / Source. In addition, each intention is assigned an $IntentionID$ which is a numeric ID of the intention. We added this ID for easier reference to the associated intention. Finally, each intention is associated with $Precondition$ which is a statement that should be verified before that intention is carried out. Only if this precondition is found to be correct will the agent try to fulfill that intention. Here again, *Condition* is the body of the rule which defines the way to generate the agent's Intentions. Once created, the system infrastructure will check each of the intentions to verify whether the agent is capable of carrying out that Action. According to the agent's capabilities, each intention is classified as intention-to or intention-that.

4.3 Argument production and evaluation

In order that an agent be able to negotiate with other agents, it must be provided with three sets of rules: (1) Argument Generation Rules. (2) Argument Selection Rules. (3) Request Evaluation Rules. We discuss them below.

4.3.1 Argument Generation Rules

Each agent in the system is assigned a list of possible argument types. Each argument type defines preconditions for its usage. Only if all of these preconditions are met, will the agent be allowed to use that argument type. These preconditions are verified against the agent's mental state. An agent cannot be certain of another agent's goals. It simply holds beliefs about the other agent's goals. Such information can be accumulated by the agent during negotiation. The rules require that the agent be able to use historical data and to evaluate the relationship between actions.

The argument generation rule is used when an agent identifies an intention that it cannot execute (it is an intention-that). The syntax for argument generation rules is as follows:

$$generate\_one\_argument\_per\_intention(AgentName, Argument,$$
$$IntentionID, CurrentTimeInterval, Precondition, Action, GoalTime, Source)$$
$$:-Condition.$$

where *AgentName* is the name of the agent; *Argument* is the generated argument resulting from use of the rule (as defined above); *IntentionID* is the numeric ID of the intention which motivates the search for an argument; *Action* is the intended activity that should be performed according to the intention and *Precondition* is a condition state that should be satisfied before the action is executed; *GoalTime* is the time interval in which the intention is to be achieved and *Source* is the source cause for that intention; *CurrentTimeInterval* is the current time interval. As before, *Condition* is the body of the rule which defines the way to generate the *Argument*. We have developed specific simple rules for the six types of arguments identified in Section 3. These simple rules can be replaced easily by more complex rules (see [144]), but we present them here to demonstrate the main mechanisms of ANA. We will demonstrate their use in a specific scenario of the Blocks World environment in Section 4.7.

An Appeal to Past Promise: As was discussed in Section 3.4 in this case, the agent expects the opponent agent to perform an action based on a past promise. This type of argument should not be used with a memory-less agent since it cannot remember past promises. We assume that before generating an argument the agent has checked that the opponent can execute the intended action. The main steps in finding such an argument are as follows:

(1) Check whether the opponent is a "memoryless" agent according to your beliefs.[26] If so, then this kind of argument cannot be used against it, since it does not remember past events.

(2) Check that you received a request from the opponent in the past, which included a future reward argument. If that reward was the intended action right now, then this argument type can be used.

An Appeal to Self Interest: As was discussed in Section 3.7, in this case, the agent believes that the requested action will serve one of its opponent's desires. This is a useful argument when the agent believes that the opponent agent is not aware of the implications. Therefore, this should not be used with a knowledgeable or reasonable agent, since the inferences of such agents are closed under consequences.

The main steps in finding such arguments are as follows:

(1) Check whether the other agent is a 'reasonable' or 'knowledgeable' agent (according to your beliefs). If so, then this kind of argument cannot be used against it, since it is already aware of its own interests.

(2) Find one desire that you believe the opponent has.

(3) Generate the list of actions, the plan, which will lead from the current world state to a state which satisfies the opponent agent's selected desire. This can be done using your own planning procedure.[27]

(4) Check whether the action (which is to be executed as a result of the argument which is being generated here), appears in the plan generated in the previous step. If so, use the selected desire in your argument, since the opponent, once carrying out the requested action, will get closer to fulfilling its own desires. This is how the self-interest condition is being

[26] Currently, the agent does not learn about its opponent's type, but uses beliefs given to it. Future implementation could be that an agent, while conducting a long-range negotiation with another agent, will learn and guess whether that other agent is a "memoryless" agent or not.

[27] In the Blocks World scenario, we use a STRIPS-like planner (see section 4.6).

checked in this rule.

An Appeal to Prevailing Practice: In this case, the agent believes that the opponent agent refuses to perform the requested action since it contradicts one of its own goals. However, the agent gives a counterexample from a third agent's actions, hoping it will serve as a convincing method. The main stages are:

(1) Find a third agent which you believe has executed the same action in the past.

(2) Make sure that, according to your beliefs, this third agent had the same goals as the current persuadee. If so, use this third agent as an example.

A Counterexample This argument is the same as Appeal to Prevailing Practice; however, the counterexample is taken from the opponent agent's own history of activities. Here, it is assumed that the agent has had a chance to observe the actions of the current persuadee in the past, or somehow has access to the persuadee's past history.

A Promise of a Future Reward: In this case, the agent promises its opponent a future reward as a condition for the opponent agent to help it execute the requested action. This should not be used with a "memoryless" agent, since it cannot remember any promises. The basic steps to generate this type of argument are:

(1) Find one desire of the other agent for a future time interval. Consider first joint desires, i.e., a desire of both agents. Also, try to find a desire which can be satisfied through actions that you can perform while your opponent cannot.

(2) Perform step 3 as in the production of arguments of the type, "An Appeal to Self Interest."

(3) This step results in a set of actions. Out of this set select an action which you can perform but your opponent cannot and that will cause you minimal cost. This action will be offered to the other agent as a reward in the future time interval, if it executes the action needed right now.

A Threat: In this case, the agent threatens to execute an action which will conflict with its opponent's plans, if the opponent agent will not help the agent in executing the requested action. The basic steps to generate a threat are:

(1) Find one desire of the opponent agent for a future time interval. Consider first desires with the highest preference value for the opponent which are not included in your desires set. Also, try to find a desire which involves actions that you can perform while your opponent cannot.

(2) Find a contradicting action to the desire [28] .

[28] For example, a contradicting action in the Blocks World, is an action which results in a block being placed in a different position than it was desired. So, for example, if according to the agent's desire BlockA should be in loc1, then putting BlockA

(3) If you are not able to find a contradicting action:
   (a) Perform step 3 as in the production of arguments of the type, "An Appeal to Self Interest." This results in a list of actions that can lead from the current state to a state believed to be desired by the opponent.
   (b) Select one action from the many actions that were generated in the previous step.
   (c) Choose a threatening action with respect to the selected action. A threatening action is one that undoes the effects of actions that would bring about a world state believed to be desired by the opponent. [29]

   The chosen action (either in step (2) or step (3) above) will be presented to the opponent agent as a threat for a future time interval, if the opponent refuses to execute the requested action right now.

4.4 Argument Selection Rules

The agent may generate several arguments for any specific situation. Only one of these should be used for each step of the negotiation. Therefore, it is required that an agent be supplied with a means for choosing one of the arguments. The syntax for such rules is as follows:

$$select\_and\_send\_one\_argument(AgentName, ArgumentList) :- Condition.$$

where *AgentName* is the name of the agent; *ArgumentList* is a list of all arguments that were generated using (the previously introduced) rules and from which one argument is to be selected; *Condition* is the body of the rule which defines the way to select and send one argument from the list of *ArgumentList*.

Currently, we implemented only one rule of selecting an argument. We order all argument types by their severity. That is, we order all of the argument types on a continuum from the weaker ones to the most aggressive ones, following the argument severity ordering of [148]. Our mechanism of choosing an argument is simple. The agent will first try to use the weakest argument and if it does not succeed, it will follow with stronger arguments (see [52]). The order is set

in loc2 is a contradicting action and putting BlockB in loc1 is also a contradicting action.

[29] For example, in the Blocks World, suppose the opponent's desired state is for BlockA to be in location loc2. Suppose, that now BlockA is in location loc1. An action that can bring about the opponent's desired state is "Move BlockA from loc1 to loc2". A threatening action then could be any one of the following actions: (i) move BlockA away from loc2, (ii) move another block to loc2, or (iii) move another block on top of BlockA.

as follows:

(1) An Appeal to Prevailing Practice.
(2) A Counterexample.
(3) An Appeal to Past Promise.
(4) An Appeal to Self Interest.
(5) A Promise of a Future Reward.
(6) A Threat.

We choose this progression since a negotiation usually begins with a simple request. Once a rejection is received, then an argument should be used. Such an argument should take into account two things: the immediate efficiency and the long term efficiency. Certainly a threat has the best immediate effectiveness. However, it may be costly for the agent to carry it out. If threats are not carried out, in the long run, they lose their effectiveness. On the other hand, appeals to an agent's common sense (i.e., an Appeal to Past Promise, an Appeal to Prevailing Practice, and a counterexample) have the least immediate effectiveness (these are too naive). However, these kind of arguments are not costly and can be used regularly. Among them, appeal to past promise is the most convincing, since refusing may reduce the opponent's credibility. The agent first tries a promise for future reward and only in case it fails, it threatens its opponent, since we believe that rewards contribute to cooperation (both for short and long run) and that, in general, all parties gain from cooperation [165].

4.5 Request Evaluation Rules

Upon receipt of a request (or a counter-request), an agent should be able to evaluate it. That is, the agent should be able to analyze a request and decide whether to accept or reject it. Moreover, the agent should be able to update its mental state resulting from new requests received and subsequent actions taken.

The syntax of such rules is as follows:

$$evaluate\_message(AgentName, Message) :- Condition.$$

where *AgentName* is the name of the agent; *Message* is the request message sent from the other agent; *Condition* is the body of the rule which defines the way to evaluate and react upon receiving the *Message*.

In our implemented agent, we defined a set of rules for evaluating the request messages. First, the agent checks if the requested action can be done according to its capabilities.[30] If not, it generates an appropriate rejection message. Next, it checks if the requested action can be done according to its beliefs about the world state and the domain. If not, it generates an appropriate rejection message. If the action can be done, the agent should evaluate the request and its argument. Since taking any action has a cost, an agent is not likely to agree to every request. For this reason, we have defined three parameters that assist in deciding whether to accept or reject a request as follows:

A Collision_Flag indicating whether the results of the requested action conflict with the agent's current goals. Possible values: TRUE or FALSE (calculation of this value is obvious).

A Convincing_Factor indicating how convincing, if at all, is the argument given for the requested action. Possible values: any integer value (positive or negative). The way to calculate this value will be shown later on.

An Acceptance_Value indicating the overall preference of the results of the requested action as opposed to all the other desires of the agent. Possible values: any integer value (positive or negative). The way to calculate this value will be shown later on.

When the agent needs to decide on the response to the other agent's request, it will first try to make a decision using the first two parameters. Computing the third parameter is time consuming and will be used only when the first two are not helpful. Table 1 specifies under which conditions a request is to be accepted by an agent and when it should be rejected. In general, the Collision_Flag and the Convincing_Factor are enough to determine the response in two extreme cases. The first case is when there is a conflict between the results of the requested action and the agent's current goals, and the argument is not convincing (first column of Table 1). In such a case the request is refused. The second case is when there is no conflict and the argument is convincing (last column of Table 1). However, for the other cases, the agent will use the third parameter– the Acceptance_Value.

The Convincing_Factor is calculated as follows. A more convincing argument should increase the Convincing_Factor and vice versa. A "do not care" argument or a missing argument will be given the score of 0. In the implementation we used the following simple heuristics to determine the Convincing_Factor.

- If the argument is an appeal to past promise, then the agent checks the past events. If a reward was indeed promised by the agent then the Convincing_Factor calculated is equal to 1.[31] If there was no past promise, then

[30] An important side effect is that the agent learns that the requesting agent is not capable of performing the requested action.

[31] Meaning, that the agent will accept the request if it does not conflict with its current goals, and if the request conflicts with its goals, it will need the Accep-

Collision_Flag = TRUE		Collision_Flag = FALSE	
Convincing_F < 1	Convincing_F ≥ 1	Convincing_F < 1	Convincing_F ≥ 1
Request rejected. with the explanation "contradicts a current goal."	If Acceptance_Value $>$ Performance_Threshold, request accepted. Add the requested action as an intention. Else, request rejected with the explanation "not worth performing."		Request accepted. Add the requested action as an intention.

Table 1
Request evaluation criteria.

the value given is 0, meaning that the agent is not convinced.
- If the request is an appeal to self interest, then the agent checks whether it is indeed beneficial for it to carry out the requested action. If so, a value of 1 is given to the Convincing_Factor, otherwise zero is given.
- Similar checks for verifying the truthfulness of the argument are done in the case of prevailing practice argument and a counter example argument. If the agent believes the argument is valid, it assigns 1 to the Convincing_Factor, and otherwise 0.
- For the future reward argument and for threats, the Acceptance_Value factor is always used. Thus, Convincing_Factor is set at 1 when the Collision_Flag is true and to less than 1 when the Collision_Flag is false.

Calculating the Acceptance_Value is needed when there is no clear cut decision using the Collision_Flag and the Convincing_Factor. Accepting or rejecting a request can influence two things. First, the total preference value that the agent will gain. Second, the number of actions that should be performed by the agent (referred to below as the intention list length). Accepting a request means doing another action in addition to the planned ones (as was determined using the rule of Section 4.2.3). It also means regenerating a completely new, and probably longer intentions list, which is time consuming. Such a list is also likely to achieve a new subset of goals, with a smaller preference value than the original. Rejecting a request seems to save the agent future unnecessary actions but it may cause the loss of a reward action offered by the other agent or lead to punishment by the other agent's threat execution. In both cases, the agent may have to go a longer route towards reaching its goals, in terms of time and number of intentions. This, too, might conclude with a smaller preference value than the original. Here is a list of parameters that the agent considers when computing the Acceptance_Value. Not all the parameters apply to all

tance_Value to make a final decision.

argument types:

(1) DL (Doing Length): The number of total intentions needed if the agent will accept the request.
(2) NDL (Not Doing Length): The current number of total intentions needed.
(3) DTL (Doing That Length): The number of "intention-that" needed if the agent will accept the request.
(4) NDTL (Not Doing That Length): The current number of "intention-that" needed.
(5) PL (Punishment Length): The number of total intentions needed if the agent will reject the request and therefore, the other agent will execute its threat.
(6) PTL (Punish That Length): The number of "intention-that" needed if the agent will reject the request and the other agent will carry out its threat.
(7) DP (Doing Preference): The preference value that the agent will gain, if it accepts the request, i.e., the sum of the preference values of the goals which will be achieved.
(8) NDP (Not Doing Preference): The preference value that the agent will gain, if it does not accept the request.
(9) PP (Punishment Preference): The preference value that the agent will gain, if it does not accept the request and the opponent will carry out its threat.

The idea is to add ratios of parameters pairs such as NDL/DL which should equal to one when there is no effect and increase when it pays to yield to the request. There are some characteristic agent parameters that are also taken into consideration:

- RL: The agent's reliability.
- ORL: The other agent's reliability for keeping promises.
- OTE: The other agent's percentage of threat executing.

The RL parameter is given to the agent by its designer when it is created. ORL and OTE are computed by the agent based on its beliefs about its opponent. Below we provide the formulas computing the acceptance value for several argument types. The acceptance value must always be more than a pre-defined limit, which is defined as the Performance_Threshold agent's parameter, in order for the agent to accept the request. This parameter belongs to the characteristics which are determined by the designer when he/she creates the agent.

4.5.1 The basic formula for Acceptance_Value

The basic formula of this section is used as the basis for calculating the Acceptance_Value of all the argument types. In cases discussed below, additional factors are added to the following basic formula.

$$Basic\_Acceptance\_Value = \Big(\frac{NDL+1}{DL+1} + 2\frac{NDTL+1}{DTL+1}\Big) \cdot \frac{DP+1}{NDP+1}.$$

The agent adds ratios of not performing the request over performing it. The addition of 1 in the formulas prevents dividing by zero. As was mentioned, accepting or rejecting a request may generate a new set of intentions with more intentions than the original. When considering the number of intentions it will eventually have to perform, the agent should differentiate between "intention-that" and "intention-to" . The reason for this is that another "intention-that" – instead of an "intention-to" – will complicate the situation. "Intention-that" means that the agent has to persuade another agent to perform that action. This task takes time and does not always succeed. Therefore, the agent gives twice as much importance to the "intention-that" ratio. [32] It then multiplies the result by the ratio of DP over NDP in order to consider the preference value gained in each way. We denote the result by Basic_Acceptance_Value.

In the case of an appeal to past promise, the reliability parameter of the agent (RL) is taken into consideration, and it is added to the Basic_Acceptance_Value to get the Acceptance_Value.

4.5.2 A promise of a future reward

The other agent's reliability should be taken into consideration in this case. In particular, the number of intentions that the agent will have, if the agent accepts the opposing agent's request, depends on the other agent's reward-keeping promises.

If the other agent fulfills its promise of reward, our agent can then subtract (at least) one intention from its overall set of intentions, because that intention will be performed by the other agent. Since the agent cannot be certain that the opposing agent will perform the reward action, the agent subtracts one intention multiplied by the opposing agent's reliability parameter (ORL).

[32] The exact form of the formulas were determined by a trial and error process.

$$Acceptance\_Value = \left(\frac{NDL+1-1\cdot ORL\cdot RD+1}{DL+1} + 2\frac{NDTL+1}{DTL+1-1\cdot ORL\cdot RDTA+1}\right)\cdot\frac{DP+1}{NDP+1}.$$

In the formula, RD is a flag which is equal to 1, if the action proposed by the opponent is really a reward, and 0 if not. $RDTA$ indicates whether the reward is toward satisfying the intention-that, which is preferred by the agent. An action is considered to be a reward (i.e., RD is set at 1) if the action is in the agent's original intentions list, or establishes a precondition for one of the agent's intentions or has the same result as one of the actions in the agent's original intentions list. It is considered $RDTA$ if the agent is not capable of performing that intention.

For example, consider a situation where BlockA is in loc1 and BlockB is in loc2. Suppose that Agent 1 desires to have BlockB at loc1 and is capable of moving BlockB, but is not capable of moving BlockA. Its intentions list may include moving BlockA to loc3 and moving BlockB to loc1. A reward could be moving BlockA to loc3, but can also be moving BlockA to another location, say loc4. Moving BlockB to loc3 won't be considered as a reward.

An agent's (AgentY) reliability in keeping promises (ORL) is calculated by another agent (AgentX) in the following way: AgentX looks for the percentage of AgentY's history of keeping its promises as is recorded in AgentX's knowledge-base. ORL is set to be this percentage.

4.5.3 Acceptance_Value of a threat

If the opponent will carry out its threat, the number of actions that the agent will need to perform will possibly increase and its preference value will decrease, given that the opponent will carry out its threat.

$$Acceptance\_Value = \left(\frac{NDL+(PL-NDL)OTE+1}{DL+1} + 2\frac{NDTL+(PTL-NDTL)OTE+1}{DTL+1}\right)\cdot\frac{DP+1}{NDP-OTE(NDP-PP)+1}.$$

The agent adds to the "not doing" portion of the ratio the number of intentions that will be added if the other agent carries out its threat. This addition is multiplied by the probability of the threat execution (OTE). That is, $(PL - NDL)OTE$ is added in the "intention-to" ratio and $(PTL - NDTL)OTE$ in the "intention-that" ratio. We calculate the preference of rejecting the request by taking into account the loss if the opponent will carry out and the probability of the threat execution (OTE), i.e., $NDP - OTE(NDP - PP)$.

The other agent's percentage of threat execution (OTE) is calculated in a way similar to the calculation of ORL, but with respect to threats.

4.6 The Blocks World Environment

The Blocks World (see for example [104,170,35]) has been selected as an example for our implementation. Consider a table with unlimited size and a set of blocks. All blocks have the same size. A world state is one of all possible combinations of blocks placed on the table or on each other. A block must be placed on the table or on another block. A block cannot be taken from the table or from another block unless immediately placed back on the table or on another block. No other action can be performed simultaneously by any of the agents. Some blocks are initially placed on the table.

We will use the notation $< blockname >, < horizontalposition > / < verticalposition >$ to specify a block position. We will omit the *blockname* when it is clear from the context. A state of the world will be a sequence of block positions between squared brackets. We will use the predicate $world\_state(s)$ to indicate that the world is in state s. For example, $world\_state([BlockA/5/1, BlockB/6/1])$ indicates that $BlockA$ is in horizontal position 5 and is on the table, and that $BlockB$ is also on the table, near $BlockA$ in location 6. When clear from the context, we will drop the predicate name and write $[BlockA/5/1, BlockB/6/1]$.

Each agent is given a set of desires. Each desire represents a subset of all possible world states which the agent will try to reach. In the Blocks World environment, we define two desires to be conflicting in the same time interval, if one of the two following cases is valid: (a) if two different blocks are to be placed in the same position according to the two desires, or (b) if a block should be placed in two different positions according to the two desires. In these cases the desires are considered to be conflicting and cannot be achieved simultaneously.

An intention, in the Blocks World case, is equal to an atomic change to the world state, in which one block changes its position and thereby creates a new world state. [33] In our example of the Blocks World environment, planning, i.e. the generation of the list of intentions is performed using a simple depth-first STRIPS-like algorithm (described in [43]). We supply the algorithm with three input parameters. The first is the current world state of blocks (see Figure 3). The second is the desired world state that was generated by the goal generating procedure. Last, we supply the algorithm with actions that change the world state. The algorithm will generate a list of intentions. These intentions, step

[33] In [144] we allow more abstract intentions, similar to the ones in our general model.

by step, change the state of the world from its current state to the desired one according to the selected goals. In our implementation, for simplicity, the STRIPS-like planning algorithm uses two very simple rules when needed:

(1) Pick a block which is not placed in its desired position. If it can be moved to its desired position, and this action will not conflict with some other block's position, then change it. If not, place that block in any position on the table which is not being used and should not be used by any other block in the desired world state.
(2) Pick a block which blocks the movement of another block which is not in its place and move it to a neutral position, as described above.

Each of the generated intentions is now categorized into one of two values: intention-to or intention-that. This is done according to the ability of the agent to perform this intention, as described above.

Consider a situation of four blocks as presented in Figure 2. Suppose that Agent 1 has the following goals: It would like BlockA to be in 3/1, BlockB in 2/1 BlockC in 4/1 and BlockD in 1/1. Agent 1 is capable of moving BlockB and BlockC but cannot move BlockA or BlockD.

Agent 1 generates the following intentions:

(1) move BlockB from 1/2 to 2/1 (intention-to),
(2) move BlockA from 1/1 to 5/1 (intention-that),
(3) move BlockD from 3/2 to 1/1 (intention-that),
(4) move BlockC from 3/1 to 4/1 (intention-to),
(5) move BlockA from 5/1 to 3/1 (intention-that).

The most well-known weakness of the STRIPS algorithm is its tendency to get into an endless loop while trying to achieve two goals at the same time. This occurs when the two goals do not conflict, while the way to achieve them does. However, the action rules provided by the algorithm in our system ensure that each new state in the world which is generated by the algorithm is different from any of the previous steps and that no one step will be generated twice. This ensures that the STRIPS search will not enter into an infinite loop. In the next section ANA's behavior will be demonstrated using a specific scenario of the Blocks World environment.

4.7 Simulation of a Blocks World Scenario

As an example, suppose there are two agents in the Blocks World environment, each with different capabilities. We assume that the agents hold beliefs about each other's desires and that each agent has desires for the next two time

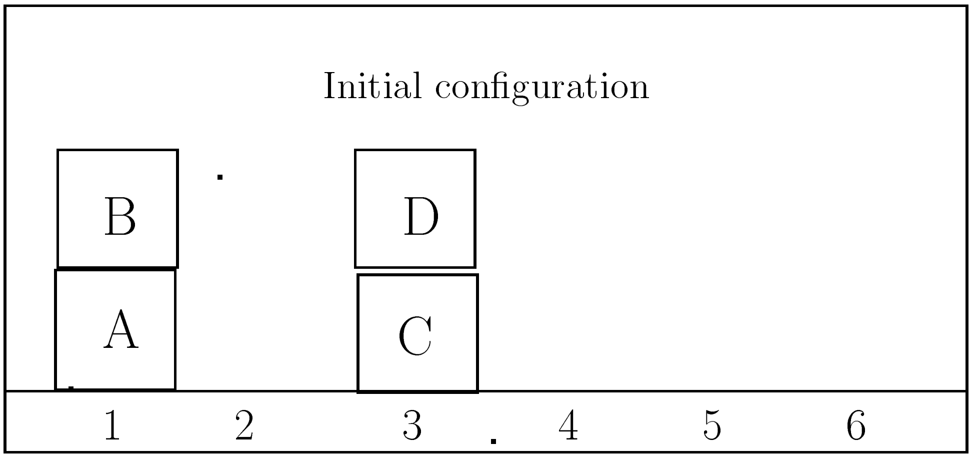

Fig. 2. The initial world state for the example in Section 4.7.

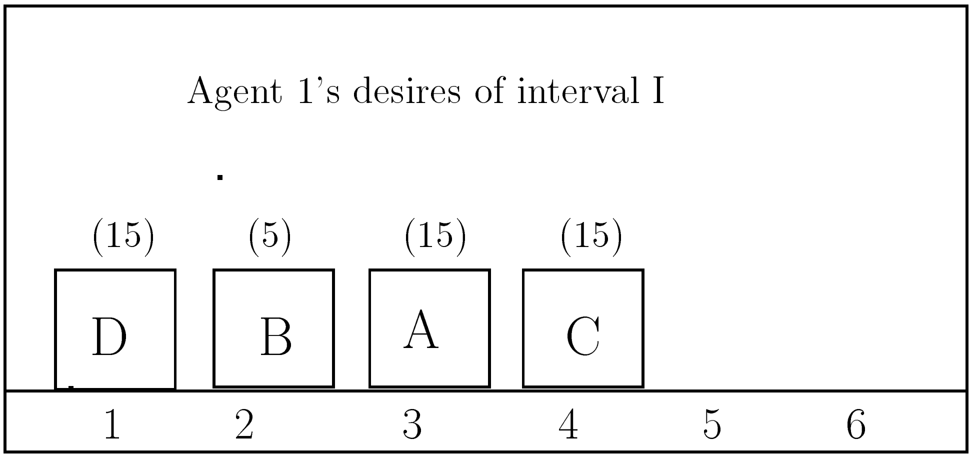

Fig. 3. The desired world state for Agent 1 interval I. The numbers above the blocks indicate the desires' preferences.

intervals.

In this scenario there are four blocks: BlockA, BlockB, BlockC and BlockD. The initial state is described in Figure 2, where BlockA and BlockC are on the table in locations 1 and 3 respectively, BlockB is on BlockA and BlockD is on BlockC. Agent 1 is capable of moving Block B and Block C and Agent 2 is capable of moving Block A and Block D.

For time interval I, Agent 1 has desires, as shown in Figure 3, and Agent 2 has desires, as shown in Figure 4. For time interval II, Agent 1 has desires, as shown in Figure 5 and Agent 2 has several desires, as shown in Figure 6.

As presented in Figure 3, Agent 1's desires for the first time interval are

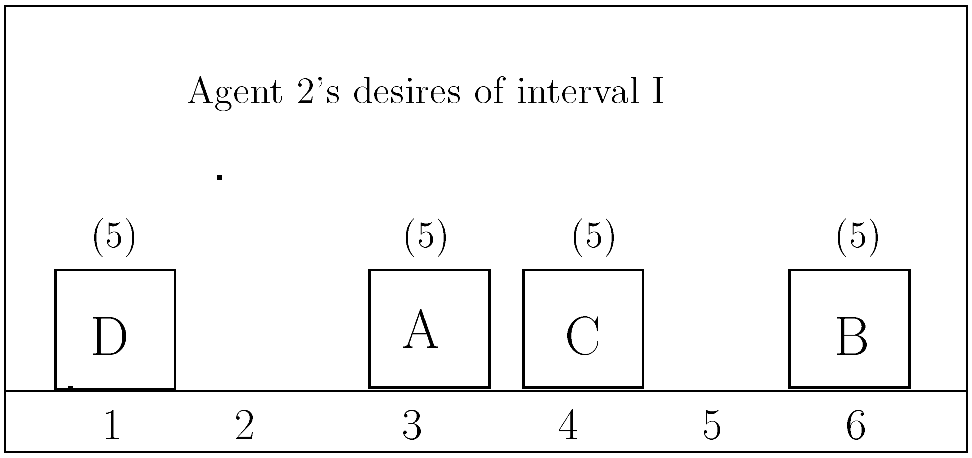

Fig. 4. The desired world state for agent 2, interval I.

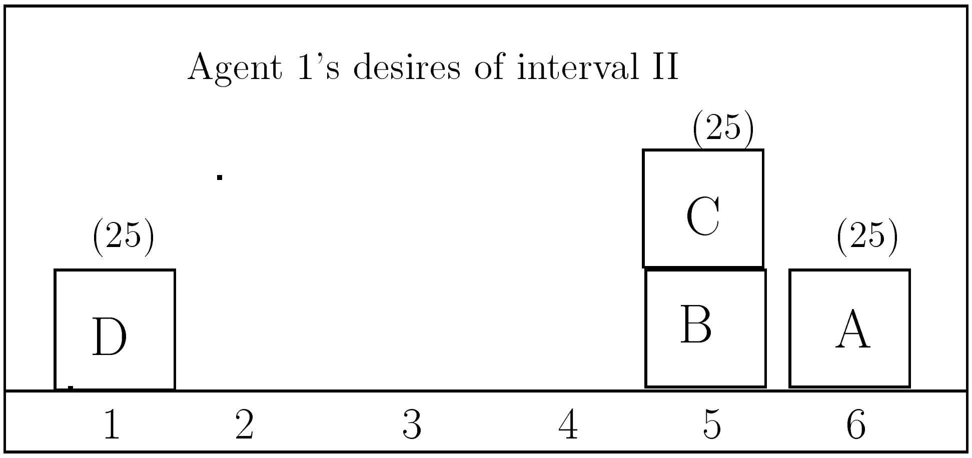

Fig. 5. The desired world state for Agent 1, interval II.

that BlockA will be in location 3, BlockB in location 2, BlockC in location 4, and BlockD in location 1, all on the table. Agent 1's preferences for its desires concerning BlockA, BlockC, and BlockD is 15, and its preference for BlockB being in location 2 is only 5. Since this set of desires is consistent, Agent 1 adopts all of them as its goals. These goals require the movement of all the blocks; however Agent 1 is capable of moving only BlockB and BlockC. Therefore, it cannot fulfill its goals without the help of the other agent.

Agent 2 also cannot fulfill its goals without the help of the other agent. It would like BlockA, BlockC and BlockD to be in the same locations as Agent 1 would like (i.e., locations 3, 4 and 1 respectively), but would like BlockB to be in location 6 (Figure 4). As in Agent 1's case, all these desires are consistent, thus Agent 2 adopts them as its goals. Since Agent 2 is capable of moving only BlockA and BlockD, it must get help in moving BlockB and BlockC. Also,

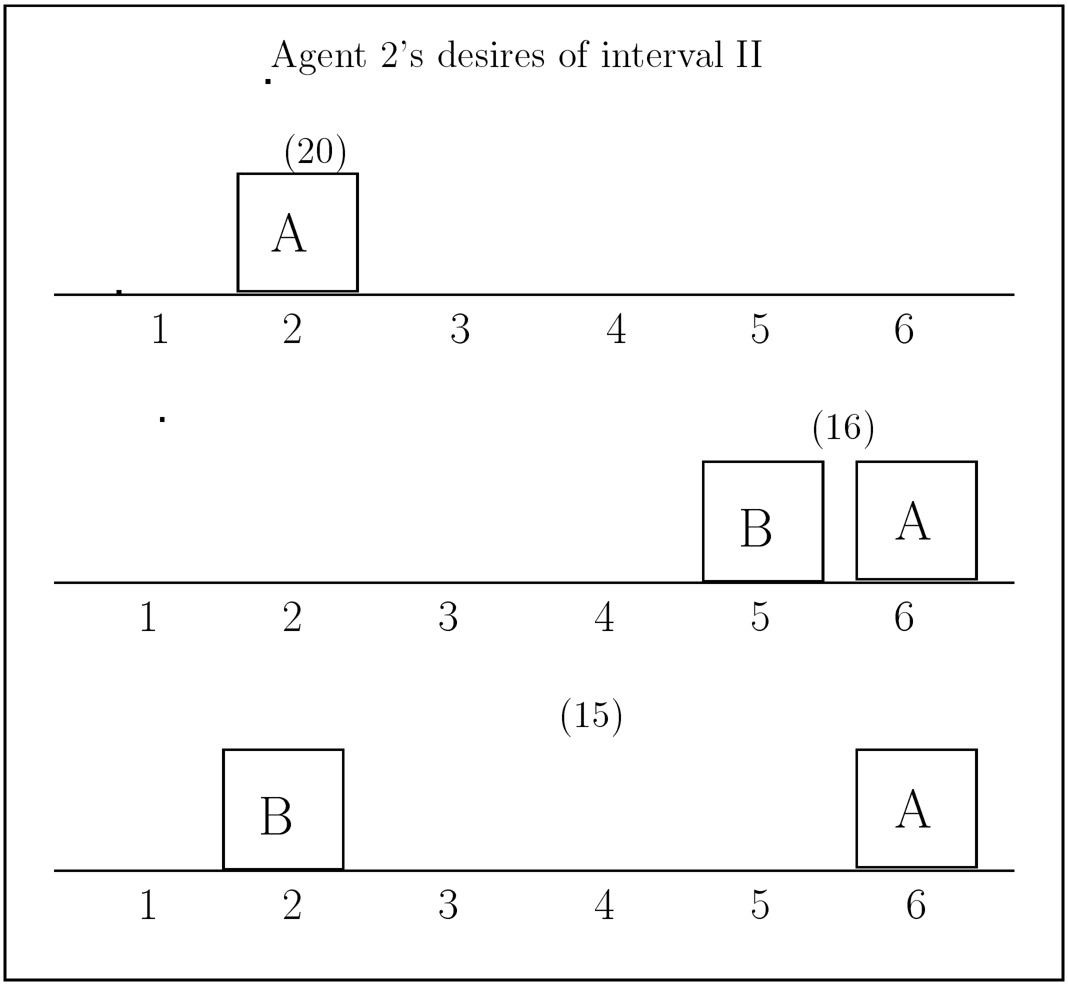

Fig. 6. The desired world states for Agent 2, interval II.

the agents will need to reach an agreement where to put BlockB.

The negotiation in the first time interval is affected by the agents' desires in the second time interval. Agent 1 would like blocks A, B and D to be in locations 6, 5 and 1 respectively, on the table and BlockC to be on BlockB (Figure 5). The preferences of all these desires is 25. Again, this is a consistent set of desires, but agent 1 may need help in moving blocks A and D.

Agent 2's desires for the second time period is more complicated (Figure 6.) They concern only blocks A and B, but conflict with each other. Agent 2 has a desire that BlockA will be in location 2 on the table with preference 20. Its desire to have BlockA at location 6 and BlockB at location 5 (on the table) has preference 16. Another desire is that BlockA will be at location 6 and BlockB at location 2, and its preference is 15. The set of desires conflicts, and thus Agent 2 will need to choose only one of them. At first, it chose the first desire (BlockA in 2/1) with the higher preference to be its goal, which conflicts with Agent 1's goals. Only the second desire of Agent 2 is compatible with Agent 1's goals.

In the beginning of the first time interval, both agents searched for plans to satisfy their chosen goals. Agent 1 found the following plan:

(1) move BlockB from 1/2 to 2/1,
(2) move BlockA from 1/1 to 5/1 (temporarily),
(3) move BlockD from 3/2 to 1/1,
(4) move BlockC from 3/1 to 4/1,
(5) move BlockA from 5/1 to 3/1.

As we mentioned above, Agent 1 was not capable of moving blocks A and, D, and therefore, the intentions with respect to steps 2, 3 and 5 of its plan were of the type intention-that.

Agent 2 found the following plan and adopted the relevant intentions:

(1) move BlockB from 1/2 to 6/1,
(2) move BlockA from 1/1 to 2/1 (temporarily),
(3) move BlockD from 3/2 to 1/1,
(4) move BlockC from 3/1 to 4/1,
(5) move BlockA from 2/1 to 3/1.

Since Agent 2 couldn't move blocks B and C, the intentions in steps 1 and 4 of its plan were intention-that.

Figures 7 and 8 describe the main steps of the negotiations and activities of the first time interval. The steps of the negotiation, which are indexed by the numbers in the first column, are only to note the order in which the agents sent their messages and took their actions. If, on the same step, both agents have an entry, that means that Agent 1 sent a message and, just after, Agent 2 sent its message. If one of the agents is missing from a step, that means that this agent did not send a message while the other agent sent two messages in a row.

We highlight some interesting points of these steps. After step 2 Agent 2 revised its plan, since whenever there is a change in the world state which the agent did not expect, then the agent re-considers its plans and intentions for the current time interval. Therefore, after Agent 1 at step 2 moves BlockB to 2/1, Agent 2 re-generates its intentions and changes its plan, so that it intends to move BlockA to position 5/1 and not to 2/1, as in its original plan (since location 2/1 is not empty any more). Also, Agent 2 intends-that BlockB be moved from location 2/1 and not from 1/2, as it originally planned (and also requested at Step 1).

In step 3, Agent 1 asked Agent 2 to move BlockA to its temporary location

	Agent 1 Activities	**Agent 2 Activities**
1		Requests Agent 1 to move BlockB from position 1/2 to position 6/1.
2	Moves BlockB from position 1/2 to position 2/1.	
3	Requests Agent 2 to move BlockA from position 1/1 to position 5/1.	
4	Rejects Agent 2 request to move BlockB, since it conflicts with its desire of locating BlockB to position 2/1.	Accepts Agent 1's request to move BlockA.
5		Moves BlockA from position 1/1 to position 5/1.
6	Requests Agent 2 to move BlockD from position 3/2 to position 1/1.	Moves BlockD from position 3/2 to position 1/1.
7		Requests Agent 1 to move BlockC from position 3/1 to position 4/1.
8		Accepts Agent 1's request to move BlockD from position 3/2 to to position 1/1 (since it has already moved BlockD to 1/1).
9	Accepts Agent 2's request to move BlockC from position 3/1 to position 4/1.	
10	Moves BlockC from position 3/1 to position 4/1.	
11	Requests Agent 2 to move BlockA from position 5/1 to position 3/1.	Accepts Agent 1's request to move BlockA from position 5/1 to position 3/1.
12		Moves BlockA from position 5/1 to position 3/1.

Fig. 7. The main steps of the negotiation in interval I.

	Agent 1 Activities	**Agent 2 Activities**
13		Requests Agent 1 to move BlockB from position 2/1 to position 6/1.
14	Rejects Agent 2's request to move BlockB since it conflicts with its desire of locating BlockB at position 2/1.	Requests Agent 1 to move BlockB from position 2/1 to position 6/1. The request is justified with a promise for future reward in the next time interval, of moving BlockA from position 3/1 to position 6/1
15	Accepts Agent 2's request to move BlockB from position 2/1 to position 6/1, since it evaluates the argument and finds it convincing.	
16	Moves BlockB from position 2/1 to position 6/1.	

Fig. 8. The main steps of the negotiation in interval I (Cont.)

5/1 so that BlockD will be moved to A's old position at 1/1 (see Agent 1's plan above).

In Step 6, Agent 2 moved BlockD to 5/1 before it read the request sent to it by Agent 1 in the same step. Thus, after it has finished the movement, and sent a request to Agent 1 (step 7), it read its message concerning BlockD, and, of course, sent an acceptance message (step 8).

Until Step 13, no argumentation was needed, since the requests were exactly according to the plans of the active agent. One request was rejected, at step 4, which concerned the conflict block, i.e., BlockB. Agent 1 moved BlockB to its desired location 2/1 and rejected Agent 2's request to move it to 6/1. In Step 13, Agent 2 requests again to move BlockB (now from its new location 2/1) to 6/1. Agent 1 rejects the request again. Therefore, Agent 2 generates a promise of future reward. Note that for keeping this promise, Agent 2 will need to give up a desire of preference 20 in the next time interval, and instead will be able to fulfill a desire of preference 16. In return, it fulfills a desire of preference 5 in the first time interval.

Agent 1 evaluates this argument and is convinced that getting help from Agent 2 to fulfill a desire of preference 25 in the next time interval, is worth not fulfilling a desire of the current time interval of preference 15. Agent 1 also believes that Agent 2 will keep its promise. Thus in Step 16, Agent 1 moves BlockB to 6/1.

	Agent 1 Activities	**Agent 2 Activities**
17	Moves BlockB from position 6/1 to position 5/1.	
18	Requests Agent 2 to move BlockA from position 3/1 to position 6/1.	Moves BlockA from position 3/1 to position 2/1.
19	Requests Agent 2 to move BlockA from position 2/1 to position 6/1.	
20		Rejects Agent 1's request to move BlockA, from position 6/1 to position 5/1, since it conflicts with its preferred desire of having BlockA in position 2/1.
21	Requests Agent 2 to move BlockA from position 2/1 to position 6/1. The request is justified by the promise made by Agent 2 to Agent 1 in the previous time interval.	Evaluates the request and accept Agent 1's request to move BlockA from position 2/1 to position 6/1.
23		Moves BlockA from position 2/1 to position 6/1.

Fig. 9. The main steps of time interval II.

.

The first time period ends after Step 16. The state of the world after Step 16 is as desired by Agent 2 (Figure 4). At the beginning of the second time interval, both agents consider their desires, generate goals, and generate plans for the new time interval. Agent 1's desires are consistent; thus it chooses them all as its goals and generates the following plan:

(1) move BlockB from 6/1 to 5/1,
(2) move BlockA from 3/1 to 6/1,
(3) move BlockC from 4/1 to 5/2.

Since Agent 1 cannot move BlockA, the second move turns into an intention-that. Agent 1 must get some help from Agent 2.

Agent 2's plan is as follows:

(1) move BlockA from 3/1 to 2/1,
(2) move BlockB from 6/1 to 5/1.

The main steps of the negotiations are listed in Figure 9. In Step 18, Agent 2 moves BlockA to position 2/1 before reading the request of Agent 1 to move BlockA to position 5/1. After reading this message, Agent 2 rejects the request, since it conflicts with its goal that BlockA be in 2/1. At Step 21, Agent 1 sends the request again, with an argument reminding Agent 2 of its promise in the previous step. After receiving the argument, Agent 2 decides to keep its promise and accept the request. It re-generates its goals for the time interval and selects the second desire (see Figure 6). It re-generates a plan for achieving the new goal which consists of moving BlockA from 2/1 to 6/1. It performs this action at step 24.

We will demonstrate the use of the argumentation rules described in Section 4.3 using the above scenario. An argument of an appeal to past promise is used in Step 21 above. An appeal to self interest could be used in Step 6 above. In this step, Agent 1 requests that Agent 2 move BlockD from 3/2 to 1/1. It could have used an argument that this act is in the interest of Agent 2, since it follows from its desires (Figure 4).

An appeal to prevailing practice could have been used if the scenario had been expanded to include additional agents Agent 3 and Agent 4 and an additional time interval. Suppose this expansion is done and suppose that Agent 3 has a desire to have BlockA at location 2/1, and Agent 4 has a desire to have BlockA at 6/1. Furthermore, Agent 3 is the only agent that is capable of moving BlockA, where the initial world state is: BlockA at 1/1. A similar situation occurred in the first interval. In the first interval, Agent 1 was convinced by Agent 2 to move BlockA to 6/1, even though it conflicted with its desires. Agent 4 can now use the activities of Interval I, as prevailing practice to convince Agent 3 to place Block A on 6/1 instead of 2/1.

A counterexample argument could be used in an example similar to the previous one, where Agent 2 (rather than Agent 3) has a desire to have BlockA at location 2/1 and Agent 4 has a desire to have BlockA at 6/1. Agent 4 can use the activities of Interval I as a counterexample.

A promise for future reward is used in Step 14 above. A threat could be used by Agent 2 in the scenario above. Agent 2 could threaten Agent 1 that in the next time interval it will move BlockD from position 1/1 to position 3/1. This threat is credible, since it will not conflict with any of Agent 2's desires in the second time interval, but will prevent Agent 1, who cannot move BlockD and wishes it to stay at position 1/1, from satisfying one of its desires.

5 Related Work

Our work on negotiation and argumentation combines a formal model, an implemented testbed for developing general negotiation agents, and an instantiation of the model in the Blocks World environment. Therefore, the work shares common research issues with three main Distributed Artificial Intelligence (DAI) areas: formal models of mental state of an agent, agent-oriented languages, multi-agent planning, and negotiation research. Other related areas of research are defeasible reasoning and computational dialectics, negotiation models in game theory, and social psychology research on persuasion. In the following subsections, we present related work in these areas and situate our research in the relevant literatures. Reader interested in detailed surveys of these areas may consult papers such as [10,167,14].

5.1 Mental State

Numerous works in artificial intelligence research try to formalize a logical axiomatization for rational agents (see [167] for a good survey). This is accomplished by formalizing a model for agent behavior in terms of beliefs, desires, goals etc. These works are known as BDI (belief, desire, intention) systems (see [128].) Many similar definitions are presented in the research community for BDI systems.

The first difference which is noted when comparing existing research is the varying usage of attitudes and pro-attitudes. Cohen and Levesque [16,18] use only two attitudes, beliefs and goals, and define other attitudes, such as intentions, using these attitudes only. Rao and Georgeff [126,128] use a wider definition of attitudes: beliefs, goals and intentions. Both Shoham [138] and Thomas [153] use the same set of mental states: beliefs, commitments and knowledge. In [152], they consider an extended set which also includes desires. In all of these cases, the definitions are not suitable for our needs of describing a more complex behavior of the agents which is required for addressing the issues of producing and evaluating arguments. We use four basic attitudes: beliefs, desires, goals and intentions. Desires, which are originally given to the agent, may be inconsistent. The goal set is a consistent subset of its desires, which the agent would like to satisfy. The intentions are formed to make the goals true.

Most of the BDI research adopts Kripke possible worlds semantics (e.g., [59,75,16].) This yields the known problem of logical omniscience for belief and knowledge and the side-effect problem for intentions that we discussed in Section 2.1. Considering agents that are not omniscient is important in the context of negotiation. Non-omniscient agents may use arguments, such as appeal to self interest, that are not useful for omniscient agents. In order to allow considering a wide variety of agents, we adopted the approach of minimal structures [15] for all our modalities. We still have the problem, in our logic, that if an agent believes p, and q is logically equivalent to it, then the agent also believes q. However, the agent may believe p but not q, even if $p \rightarrow q$.

Other approaches that attempt to solve the problems associated with omniscience include [84,38,71,33,39]. Works for appropriate semantics for intentions include [69,28,126] and [73] which also take the minimal structure approach. The advantage of our approach is that we deal with all the modalities in the same manner, which addresses both issues (omniscience and side effects for intentions), and also allows for contradicting desires, a condition that is critical for negotiating agents.

In addition, since time plays an important role in negotiation, we do not use simple minimal structures as in [73], but rather our possible worlds are time-lines [152]. This enables us to consider agents' attitudes toward past and future events and their change over time. Georgeff and Rao [126,129] consider complex possible worlds semantics (but not minimal structure models). In their formalism, the belief-desire-intention accessible worlds are branching time structures. We preferred time lines that also allow uncertainty about the past, not only about the future, as in their model.

In our framework, an agent can reason about other agents' mental states in terms of its beliefs, intentions, and goals. Not all systems allow this behavior (for example, see [58,16,18]). However, most of the latest systems do allow such reasoning (for example, see [138,153,128]). This is the exact difference between first-order intentional systems (where agents reason only about their own mental states) and second-order intentional systems (where agents reason about other agents' mental states, as well) as stated in [25]. In our system, reasoning about other agents is a necessity in order to address the needs of argument generation. In addition, in our logic, every goal of an agent is also one of its desires. This is a unique relationship between the two attitudes, since in most other systems agents do not hold both mental states.

In our logic, every goal is also an intention. Additional intentions are steps to satisfy goals. However, there may be some intentions that are not motivated by an agent's own goal, but rather by a request from another agent. Researchers who didn't consider multi-agent environments with negotiation (e.g., [126]) assumed that every intention is also a goal.

Another difference concerns the relations between intentions and beliefs. As discussed in Section 2.6, Cohen and Levesque [16,18] assume that if an agent holds an intention toward a proposition, then the agent believes that this proposition is not true, but that it will be true some time in the future. According to their logic, time doesn't explicitly appear in the proposition. In our logic, we only require that the agent believes that its intentions are possible (Axiom (INTB2:34) of Section 2.6). Since time is expressed explicitly in our logic, we are able to present different types of agents with different levels of self-confidence and characterize the appropriate semantic constraints on their accessible relations. Georgeff and Rao [126] also consider interesting relations between an agent's beliefs about possible histories and its intentions.

As described in section 4.1, in ANA we distinguish two types of intentions (intention-to and intention-that). Many other systems, such as that of Cohen and Levesque, do not make this distinction. In Cohen and Levesque's system, once an agent adopts an intention, the agent will look for ways to achieve it by itself. Moreover, the agent believes that the intention can be achieved. In contrast, in our case, the agent, once creating an intention, knows whether it is capable of executing it on its own or not. However, the agent is uncertain whether there is another agent which is able to successfully execute that intention. Here, we follow the logic as presented in [55–57].

Since our agents are self-interested, we do not try to provide formal specification for agents working together on a joint goal [85], team activity [17,143], or SharedPlan [55,56]. We concentrate on agents that have their own desires and negotiate to obtain help or to resolve conflicts in order to maximize their own utility.

5.2 Agent Oriented Languages

The idea that agents are modeled in terms of their mental states led to Shoham's [138] definition of an Agent Oriented Programming paradigm. Shoham first implemented an AOP language called the Agent0 system. It enables modules to process knowledge and beliefs about others and about the world. In order to be considered an AOP language, a program must allow its user to define agents and to assign to each of them a set of capabilities, initial beliefs, initial mental states, and some mental-change rules (to update their mental states over time). A few years later Thomas presented a second language which is the descendant of Agent0, called *PLACA* [153]. Its main contribution is the ability of its agents to plan their activities.

Our system ANA can be viewed as an AOP language as well. Our main concern when building ANA is to allow its user to define the environment, set of rules for negotiation, and inference on different stages of the negotiation. As in the case of PLACA where a planning mechanism is added to the Agent0 language, our contribution is to enable a user to define a planner with the ability of conducting a negotiation between the agents. Our aim is to allow argumentation in order to fulfill each of the agents' desires and resolve conflicts during planning so that successful plans can be found.

In the Agent0 system, the agents act according to their commitment rules. Each commitment rule contains a message condition, a mental condition, and an action. The agent acts only when these two conditions are met. More specifically, the agent acts only when it receives a message from other agents. This is not the case in ANA. Our agents are motivated by their desires, and their activities are almost solely caused by their own wishes and desires. The more obvious difference between ANA and the two systems Agent0 and PLACA is the negotiation and argumentation ability, and the concept of agent life cycle being supported by our system.

The Concurrent Metatem language by Fisher [44] presents a system in which multiple agents can work simultaneously. Each agent is assigned a unique behavior specification and it acts according to these behavior rules. ANA is similar to the Concurrent Metatem system with respect to these two features. However, there are two major conceptual differences between ANA and Concurrent Metatem. The first is that Concurrent Metatem allows and supports grouping of several agents together. Agents can be grouped together and form a group, agents can be added (removed) to (from) an existing group, and messages can be sent to all members of a specific group. We do not support such a feature in our system since we are interested in examining single agents that handle negotiation on their own. The second difference is the fact that the behavior rules in Concurrent Metatem are based on on general condition-result rules and not on BDI logic.

Wavish [161] presents the Agent Behavior LanguagE - ABLE system which is a two-level agent behavior system that supports user definition of agents. The first level defines behavior rules for an agent, while the second level can be used to define behavior for a group of agents. These rules of behavior are called licenses, schemas, and functions, and can correspond to different kinds of forward chaining production rules. In ANA such rules do not exist. Since we are interested in negotiation, we present rules for behavior that relate to negotiation only. Another difference between the two systems, is the fact that the ABLE system can execute several rules for the same agent in parallel. These rules can be nested and can also be limited in time. These features and capabilities are missing from our system, but are not needed at this stage of the research. As in the case of the Concurrent Metatem system, the ABLE system is not a BDI system.

Rao, Ramamohanarao and Weerasooriya [130] present a distributed autonomous system, called AgentSpeak. In its environment, agents are organized in families, offering services to other agents, as well as data sharing functionality.

This language supports concurrent object oriented languages and well developed communication capabilities. Although called AOP, strictly speaking it is not, since it does not support well defined BDI features for its agents.

The Agent PRocess Interaction Language - April [98] is a multi-agent system. Its main purpose is to construct a means for facilitating pure multi-tasking and communication, in parallel to pattern matching and symbolic processing capabilities. Trying to cover all of these issues produces difficulties in specifying and implementing agents, since the user must use only basic April primitives. ANA uses the Prolog programming language in order to get the underlying logic part of the system working, while the multi-agent part was developed on top of the logic infrastructure. This enables the user of ANA to define the agents in a more intuitive way. Yet another difference between the two systems is the ability of an ANA agent to plan its actions in advance, an ability lacking in the April agents.

In order to demonstrate the usage of ANA, we implemented a simple planner for the Blocks world environment. This planner can be easily replaced by other planners. It is important to remember that the aim of the planning activity executed by the agents in ANA is to seek ways for reaching their goals. An agent only tries to satisfy its desires by generating a list of actions which will lead it to its goals. This definition of planning is different from many other DAI systems in which the term planning indicates planning for multiple agents' activities, that is, planning several agents' actions in order to reach some kind of agreement or to schedule tasks between the agents (see Section 5.3 below.) Such tasks can be seen in ANA as one of the goals of the argumentation process, but the planning mechanism and process which is carried out by each of the agents is meant to generate a list of steps which the agent will use to satisfy its own goals.

Rao and Georgeff [128] also consider the problem of finding for their system. In their case an agent was given a set of predefined plans which can be used to satisfy several goals at the same time. Their agent chooses one of the plans and in that way assigns itself a set of actions to be performed. The selection is based on the agent's beliefs about the side affect of the plan, its desires, and intentions. Their planning activity fulfills a purpose similar to ours. However, they also impose on the agent the belief that its goals are achievable if all agents act in appropriate ways. This is a very strong assumption (or limitation) that we do not hold in our system.

As mentioned above we implemented a very simple mechanism for planning an agent's intentions from its goals. This is accomplished by executing the STRIPS-like algorithm (as described in [43]). The algorithm is adjusted for our needs and world example. Similar usage is presented in [1]. Since planning is not a main task of our system, we will not present a thorough overview on the subject. For a well presented survey on planning, the reader is advised to see Allen, Hendler and Tate [2].

5.3 Multi-agent Planning

Multi-agent planning in DAI research has evolved along a number of different dimensions. One dimension focuses on agents that cooperate to solve subproblems of a given problem and integrate the results [21,20,30,24,114,36]. Other lines of inquiry concerns planning for task allocation, so that effective execution will result [141,95], centralized planning for multi-agent execution [113,67], or centralized planning to avoid execution time action conflicts [50,51]. Another area of research has concentrated on multiple agents' mental attitudes for coordinating their activities [16,150,55,154].

Another area of investigation has focused on multiple agents each of which is self-motivated, i.e. has its own goals, which could be in conflict with the goals and/or actions of others [134,136]. This literature has not concentrated on deriving the plan steps; instead, it assumes that the agents are capable of deriving plans, and only concentrates on resolving the conflicts in goals and utilities through forms of coordinated negotiation [171,20].

Our work combines plan generation with negotiation and argumentation during planning as an explicit mechanism for plan adjustment and conflict resolution during planning.

Argumentation has been used as a method to represent interactions in a multi-agent plan development process [42] in a mixed-initiative context. In particular, Ferguson and Allen's work [42] aims to define plans as arguments, in the sense of [90], so the agents can reason defeasibly whether a certain course of action under certain explicit conditions will achieve certain goals. In our work instead, we use arguments as a mechanism to influence the intentions of other agents so that effective plans can be produced.

5.4 Automated Negotiation

Negotiation is usually referred to as a communication process used to achieve coordination and resolve conflicts (see discussion in [147]). Two main approaches are introduced in previous research. The first approach suggests that by communicating, agents can influence each other's goals and intentions. This influence should lead to resolution of goal and plan conflicts and better cooperation (see [148]). The agents exchange proposals, counterproposals and arguments and incrementally reach an agreed upon solution. Our work follows this line. The agents simulated in our system try to convince each other using argumentation to change their intentions and perform actions which are beneficial to the other negotiating party. The PERSUADER system by Sycara (see [148,147,146]) is a similar system which involves a multi-agent program that operates in the domain of labor negotiation. It contains three agents: a company, its union, and a mediator. The negotiation model of the PERSUADER has also been applied to the domain of concurrent engineering [149]. Our paper tries to create a more general solution for any domain, without limiting the number of agents which can perform negotiation. Our system inherited many of the ideas presented in PERSUADER. Some of these are the use of utilities to change the agent's beliefs and behavior, several types of arguments, setting levels of strength for each argument type, and the argumentation strategies.

The second approach suggests that incremental suggestions performed by the agents can find a plan that will satisfy all agents (refer to [78,74,136]). Here one of the main issues is time and how it influences the negotiation. As time passes, some agents may benefit since they may learn more about their opponent. However, other agents may lose, since each of their goals is aimed for a certain time. Therefore, each agent will have a different approach to the negotiation process. In [78], a distributed negotiation mechanism is introduced which is used by the agents. Using this mechanism, the agents conduct negotiation and can reach efficient agreements without delays. Moreover, it is shown that the individual approach of each agent towards the negotiation time, affects and even determines the final agreement that is reached. The main conclusion is that delaying agreements causes inefficiency in the negotiation. Our current paper does not consider the overall negotiation time, since we are interested in creating a general system for automatization of the negotiation process. In ANA, we assume that there is enough time to perform the negotiation and that each agent's utility does not change over time. Therefore we do not analyze the impact of time on the negotiation.

The Diplomat system by Kraus and Lehmann [76] presents a general model for a negotiating agent and handles the issues of who to negotiate with, how to generate suggestions, how to evaluate counter suggestions, and how to form coalitions among the players. Yet this system was implemented for one specific domain - the game Diplomacy. Our system, though intended to be a more general system, did not address difficult issues such as coalition building. We focus on allowing the user to define his/her own set of rules that will perform his/her own kind of desired negotiation and argumentation.

Research by Lesser and Durfee and colleagues [31,23,82], addresses the issue of agents' communication in distributed problem solving systems. Here, the main purpose of the research is to combine information from several agents (usually sensor data) in order to reach a conclusion for the entire group. The information exchanged between the agents is usually based on partial solutions for various levels of the problem. Later these will contribute to the global solution. In other research, (e.g. [83,89,20]) negotiation is used as a metaphor for a group of heterogeneous agents that use different search operators to try to arrive at a global conflict-free solution. In other negotiation models (e.g. [163]), agents negotiate through an arbitrator who resolves the conflicts that arise. Our approach differs from this line of work. In these works it is assumed that the agents are designed as part of a global system and are working towards a global system wide goal. In our work, each agent is trying to achieve its own goals. We do not assume any common knowledge or goal, nor any immediate will to cooperate. On the contrary, we assume that the agents are self-motivated and that cooperation should be pursued and achieved via negotiation and argumentation.

Negoplan by Matwin, Szpakowicz and Koperczak [97] is a decision support system for conducting negotiation. Through simulating both parts of the negotiation process, its main task is to give one party of the negotiation an advantage. Its main model presents the current situation of the negotiation and by using a Goal Representation Tree (GRT) it is able to suggest paths for the negotiation process in which a better outcome could be achieved for the party using the tool. Although this is a general system, its main task is to support the user in negotiation and to suggest actions. It does not represent different negotiation strategies nor use argumentation. In contrast, our system allows its user to simulate several agents together, each of which uses different kinds of argumentation, resulting in an analysis of the negotiation and argumentation process.

Liu and Sycara [88] have modeled negotiation as a constraint relaxation process where the agents are self-interested in the sense that they would like to achieve an agreement that gives them highest utility, but are also cooperative in the sense that they would accept lower utility to facilitate reaching an agreement. The model does not use argumentation for mental-state revision.

Parsons and Jennings [112] have followed our formalism described in [77] to construct arguments to evaluate proposals and counterporposals in negotiation. A recipient agent evaluates a proposal by constructing arguments for and against it. Their notions of argument defeat are based on the work of [45,90,34,79].

Gasser [48] discusses the social aspects of action in multi-agent systems. In his view, different social mechanisms can dynamically emerge, resulting in changing the communication language between the agents, and forming different communities of agents. This approach is most effective when the agents' structure is continuously changing or no structure exists at all. In our system, this is not the case. Although we are interested in environments in which no pre-defined protocol exists (for solving the conflicts between the agents), we do assume a formal means for interfacing between the agents.

Zlotkin and Rosenschein [170,134,172,171] lay the ground for a domain theory of negotiation. They describe a way of classifying interactions between agents. This classification helps designers of agents to choose appropriate negotiation mechanisms and strategies. They use the Nash solution which maximizes the product of the agents' utilities and call it the Unified Negotiation protocol (UNP) so it can be used in different types of encounters. The main domain in which their research is concentrated is the Blocks World environment, which we also used as an example domain. However, there is a significant difference between the two works. We base the negotiation part of our work on a formal logic model, based on the agents' mental state. We did so, since we wanted to analyze scenarios in which there is no pre-defined protocol or mechanism for solving the conflicts between the agents. This is not the case in the work of Zlotkin and Rosenschein. In addition, they do not deal explicitly with the construction of a sequence of plan steps but assume that the agents have somehow constructed their plans and are choosing among them to satisfy conflicting notions of utility. In contrast, we show how explicit communication of arguments and the change in mental states of the agents are part of the plan construction process.

Recently, there has been increasing interest in integrating learning into the negotiation process (e.g., [123,32,111].) Zeng and Sycara [168,169] have developed an economic bargaining negotiation model, where the agents are self-interested. The model emphasizes the learning aspects. The agents keep histories of their interactions and update their beliefs, using Bayesian updating, after observing their environment and the behavior of the other negotiating agents.

5.5 Defeasible Reasoning and Computational Dialectics

Many argumentation logical systems have been proposed in defeasible reasoning. The main purpose of these logics is to construct "defeasible proofs", called arguments that can be partially ordered by relations expressing differences in conclusive force. Arguments are *prima facie* proofs that may make use of assertions that one sentence is a (defeasible) reason for another. They indicate support for a proposition, but do not establish warrant [156] once and for all; it matters what other counterarguments there may be. Arguments may have structure [90,116], or may just be collections of supporting sentences [122,49]. In general, this body of work has focused on formalizing the "strength of arguments" and criteria for determining difference in strengths among arguments so that undefeated arguments can be found and presented. This body of work assumes that an argument that is logically undefeated is the most persuasive.

In other words persuasion is viewed on only syntactic grounds with no explicit links and representation of agents' objectives, actions or preferences (utilities) [94].

A large number of formal argumentation systems exist. Pollock [116,117,119,118,120,121] developed the argumentation system OSCAR that can reason with *suppositional arguments*. Nute [108,109] developed the LDR system where adjudication among competing arguments is performed via so called *top-rules*. An argument defeats another if and only if the antecedent of the top-rule of the first argument is strictly more specific than the antecedent of the top-rule of the second. Loui [90–93] presented a system of argumentation where defeat among arguments is defined recursively in terms of interference, specificity, directness and evidence and introduced a new defeasible operator [139]. In addition, Loui developed an implementation for this rule system to compute defeat among arguments [92].

Horty and Thomason [63] presented a theory of mixed inheritance in non-monotonic proof nets that resembles symbolic argumentation, where arguments are called *paths*. Paths are one-dimensional lines of reasoning and are therefore simpler than argumentation systems where arguments are trees. Lin and Shoham [87] developed an argument system that captures some well-known non-monotonic logics (e.g. Reiter's default logic, McDermott and Doyle's non-monotonic logic etc). In their system, they do not have logical hierarchy among arguments. So, it is not possible to determine which argument is undefeated. Konolige [72] proposed a solution to the Yale Shooting Problem in the context of which he discussed important issues in defeasible argumentation. His formalism is based on situation calculus, where properties are attached to situations. Dung [29] presented a mathematical argumentation theory where an argument is accepted by S if and only if S attacks all attackers of that argument.

Vreeswijk [160] presents a thoughtful critique of existing argumentation systems. We present some of his critique. For example, in some cases, (e.g. [116]) fallacious arguments are produced that defeat lines of reasoning; or (e.g. [90]) cyclic sets of arguments can be constructed where each argument defeats its successor. In general, most systems have difficulty in situations where it cannot be unambiguously determined which argument should win. Various attempts to fix this have given rise to approaches such as credulous reasoning [131,96,99], skeptical reasoning [90,109,121] and others. Almost all systems include detailed specification of defeat. However, it is possible to construct arguments that are syntactically isomorphic but, for different semantic situation descriptions, an undefeated argument is correct for one situation but wrong for the other [159]. Vreeswijk [160] presents an *abstract argumentation system* where it is not attempted to prescribe how argumentation should be performed, what arguments are in force, or how defeasible information should be manipulated, but rather present a general framework in which basic notions of argumentation (e.g. defeat) take on well-defined meaning.

Our work differs from this body of work in that in our system we do not focus on the logical defeasibility or on which arguments to present in legal reasoning. Rather, concentrate on how argumentation can guide negotiation by supplying a mechanism to agents to influence the beliefs and actions of others and achieve coordination in situations where agents are self-interested. In this respect, our research also touches upon research on game theory and multi-agent planning.

AI work in legal reasoning has concerned itself with legal disputation, arguments and their refutations, based on rules or cases. Much of this work has been case-based [3,4,132,140]. In particular, this work is concerned with criteria for argument strength based on factors of a case, and when and how to combine rules and cases to support claims. Additional work in this vein includes [94] whose focus is reasoning by representing argument rationales, defining rationale types and providing a formal account of how they change legal disputation. Case-based argument is exactly our argument type of “Appeal to Prevailing Practice” and is the most frequently used argumentation method in legal reasoning.

5.6 Game Theory's Models of Negotiation

Game theory work [46,155,101] concerns itself mainly with determining conditions under which the outcomes of a game can be predicted. In particular, solutions in game theory consist of equilibrium strategies. The notion of Nash equilibrium [102] is heavily relied upon. A profile of strategies forms a Nash equilibrium if each player's strategy is an optimal response to the other players' strategies. In a Nash equilibrium, the players take their opponents' strategies as given and therefore do not consider the possibility of influencing them.

In games in which a player chooses some actions after observing some of his opponents' actions (dynamic games), this conjecture leads to some absurd Nash equilibria [155]. Selten [46] proposed a more restrictive equilibrium concept for dynamic games, the subgame perfect equilibrium. The basic idea of subgame perfect equilibrium is to select Nash equilibria that do not involve non-credible threats, i.e., a threat that would not be carried out if the player were put in the position to do so, since the threat move would give the player lower payoff than he would get by not doing the threatened action. In many games of complete information, this notion turns out to be very powerful [135]. However, it was limiting for games of complete information. In dynamic games of incomplete information a player who observes another's move can extract information. The inference process takes the form of Bayesian updating from the second player's supposed equilibrium strategy and the observed action. For such games, Selten [137] introduced the notion of perfect Bayesian equilibrium. The outcomes of such games depend on what one party believes the second will do in response to actions by the first. However, for information sets that are not reached, Bayes formula does not hold. Thus, along the equilibrium path, a move by a player can be designed to influence his opponent's beliefs, but a move off the equilibrium path is considered a zero-probability event, so it cannot be chosen. However, Selten's equilibrium notion could not adequately restrict out-of-equilibrium beliefs, thus enabling the players to establish credibility for "too many" threats. In other words, equilibrium could be supported for threatened behavior through incredible beliefs [54]. The unrestricted proliferation of equilibria also occurs in games with communication (e.g. signaling games).

To be able to pose restrictions on beliefs so that unreasonable equilibria could be eliminated, game theorists more closely examined notions of credibility of threats and consistency of beliefs (e.g. [100,54,41]). Grossman defines the notion of metastrategy which prescribes for each player's information set and his beliefs over the information set, the set of actions the player will follow. A metastrategy specifies a player's action when his beliefs are given by a probability distribution and he has observed a message sent by the other player. An updating rule specifies the belief that a player has at each of his information sets as a function of his belief in the past. Based on these notions, Grossman defines credible updating rule and consistent belief. Using these notions, he generalizes the idea of a game node to include a belief as well as a history. Using his mathematical definitions of these notions, he is able to show that he can restrict the set of resulting equilibria.

The main focus of game theoretical research is on defining appropriate notions of equilibria and the conditions under which they can be obtained. Communication is considered but is not explicitly and exogenously represented. In our framework, we do not focus on the presence of equilibria solutions, but rather concentrate on a logic framework for explicitly representing and evaluating communications (arguments) where arguments are connected to an agents' mental attitudes. Arguments are exogenous to the game. Myerson [100] put it well: "The theory of non-cooperative games with signaling and communication (based largely on Aumann's [5] correlated equilibrium and Kreps and Wilson's [80] sequential equilibrium) derives the meaning of all statements and signals from the equilibrium in which they are used. In fact if the mere act of saying something does not directly affect any payoffs, then there is always a "babbling" equilibrium in which every player randomizes over the set of his possible statements independently of his information and his payoff-relevant actions, and in which all other players ignore his meaningless statements. Such analysis suggests that communication can only increase the set of equilibria, and cannot provide a way to select among equilibria. To escape from this conclusion, we must introduce the assumption that negotiation statements have literal meanings that are exogenously defined". Myerson, then proceeds to define the notion of the credibility of negotiation statements in terms of "tenability", "reliability" and "plausibility". These are defined precisely and mathematically. He defines negotiation statements in terms of an allegation that describes a negotiator's private information, a promise that describes how the negotiator plans to choose his future actions and messages, and a request that describes strategies for the other players that the negotiator may want or expect them to use thereafter. We see a deep connection between our own work on arguments especially in viewing them as mechanisms effecting change in belief and behavior of players and Myerson's framework. Our logical framework provides a language and inference mechanism for modeling and deploying a broader range of argumentation strategies, however we do not explicitly provide a solution concept formulation. In addition, we integrate the argumentation framework within a process oriented model of agent interactions. It is interesting to note that Myerson's mathematical formulation of conditions for credibility of negotiation statements (although computational issues are not addressed in his work) could be integrated into our framework and used to determine the convincing power of an argument and its evaluation by the receiver agent.

5.7 Social Psychology

A wide large of literature in social psychology and in particular in social judgement theory deals with persuasion. The basic tenet of social judgment theory is that attitude change is mediated by judgmental processes and effects. Persuasion is seen as a two-step process in which the receiver assesses the position advocated in the message, and changes attitudes [110]. This literature presents theoretical hypotheses that are then tested through controlled experimentation with human subjects to investigate the factors that affect receiver's judgement and cognitive attitude change.

The research most relevant to our paper include a number of investigations. Experiments have found that threats ("fear appeal") are very effective [11,145]. Within this context, some researchers have examined the fear appeal content of the message, whether for example it contains gruesome pictures etc. But there is no conclusive evidence that such messages are persuasive. Another set of findings relates to presenting particular examples vs. statistical summaries. Robust experimental findings show that presenting examples is much more persuasive [70,151,107,61]). This relates to our argument type of "appeal to prevailing practice".

Other issues that social judgment theory examines are the notion of whether increased credibility of the source of the argument (the persuader) increases the persuasive force of an argument. This finding is robust [8,9]. In addition, research has been performed concerning argumentation strategies (presentation of sequences of arguments). Though the experimental evidence is not absolutely conclusive, it seems that the climax strategy (i.e. present the weakest argument first and follow with increasingly stronger arguments) works best [52]. This is the strategy we adopted for our model.

6 Conclusions

In a multi-agent environment, where self-motivated agents try to pursue their own goals, cooperation cannot be taken for granted. Persuasive argumentation has been advocated as a general mechanism for planning how to influence agents' intentions in order to increase their cooperativeness and reduce disparities and conflicts. In this paper we have presented a formal framework for argumentation and a simulation environment in which the user can develop and test various algorithms and mechanisms for establishing communication and negotiation between self-motivated agents. In addition, this explicit representation of and reasoning about argumentation has been interleaved in the process of constructing joint plans where the agents may have different goals and where plan steps may give agents differing amounts of utility.

A formal mental model of the agents based on minimal-structure of possible worlds (time lines) has been developed using modal operators for beliefs, desires, intentions and goals having an appropriate set of properties. Under different assumptions of agent properties and conditions on the agent models, different types of agents have been defined. A formal axiomatization scheme has been constructed for argument generation and evaluation based on argument types identified from human negotiation patterns. In addition, suitability of the various types of arguments for the different agent types has been described. A persuading agent uses its model of another agent, the persuadee, in order to generate persuasive arguments that are suitable for the negotiation situation and the particular persuadee type.

As distributed systems become more widespread, and as humans become part of human-agent systems, the need for persuasive argumentation formalisms will become apparent, especially when agents constrained by incomplete knowledge and bounded rationality will be forced to interact. One obvious and immediate application of such formalisms is agents that negotiate to provide services on the Internet-based environment.

In summary, the main contributions of this paper are:

- Formalizing to some extent the "semantics" of argumentation and linking to mental attitudes of the agents.
- Unlike game theory, we explicitly represent and reason about arguments as mechanisms for influencing others' beliefs and behavior in interactions of self-interested agents (in situations of incomplete information).
- The reasoning about argumentation is integrated in a multi-agent planning system where it is used to reconcile conflicting goals and plan steps and guide plan adjustment and adjudication.

One aspect of our future work includes a more detailed investigation of the relations between different modalities. Another future focus will be investigation of change in the modalities over time in the course of the argumentation process, and as the result of external events and observations from the environment. An analysis of the credibility and reputation of adversaries based on repeated encounters is currently being incorporated into the argumentation process.

Most importantly, future research will comparatively evaluate various arguments in different negotiation settings and for different types of agents. Creating new arguments and verifying their effectiveness under different conditions may lead to effective criteria for selecting one best argument in a specific situation, thereby getting the most out of the negotiation.

References

[1] J. F. Allen. Planning as temporal reasoning. In *Principles of Knowledge Representation and Reasoning: Proceeding of the Second International Conference (KR'91)*, pages 3–14, San Mateo, CA, 1991. Kaufmann.

[2] J. F. Allen, J. Hendler, and A. Tate. *Readings in Planning.* Morgan Kaufmann Publishers, San Mateo, CA, 1990.

[3] K. Ashley. Defining salience in case-based arguments. In *Proceedings of IJCAI-89*, pages 537–542, 1989.

[4] K. Ashley and V. Aleven. Toward an intelligent tutoring system for teaching law students to argue with cases. In *Proceedings of 1991 ACM International Conf. on AI and Law*, 1991.

[5] R. J. Aumann. Correlated equilibria as an expression of bayesian rationality. *Econometrica*, 55:1–18, 1987.

[6] C. Baral, S. Kraus, J. Minker, and V.S. Subrahmanian. Combining knowledge bases consisting of first order theories. *Computational Intelligence*, 8(1):45–71, 1992.

[7] J. Bates, A. Bryan Loyall, and W. Scott Reilly. An architecture for action, emotion, and social behaviour. In C. Castelfranchi and E. Werner, editors, *Artificial social systems*, pages 55–68. Springer-Verlag, 1994.

[8] M. J. Beatty and R. R. Behnke. Teacher credibility as a function of verbal content and paralinguistic cues. *Communication Quarterly*, 28(1):55–59, 1980.

[9] R. A. Bell, C. J. Zahn, and R. Hopper. Disclaiming: A test of two competing views. *Communication Quarterly*, 32:28–36, 1984.

[10] A. H. Bond and L. Gasser. An analysis of problems and research in DAI. In A. H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 3–35. Morgan Kaufmann Publishers, Inc., San Mateo, California, 1988.

[11] F. J. Boster and P. Mongeau. Fear-arousing persuasive messages. In R. N. Bostrom, editor, *Communication yearbook 8*. SAGE Publications, 1984.

[12] M. E. Bratman. What is intention? In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication*, pages 15–31. MIT Press, 1990.

[13] M. E. Bratman. Shared cooperative activity. *The Philosophical Review*, 101:327–341, 1992.

[14] B. Chaib-draa. Distributed artificial intelligence: An overview. *AI Review*, 6(1):35–66, 1992.

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

[16] P. Cohen and H. Levesque. Intention is choice with commitment. *Artificial Intelligence*, 42:263–310, 1990.

[17] P. Cohen and H. Levesque. Teamwork. *Noûs*, pages 487–512, 1991.

[18] P. R. Cohen and H. Levesque. Rational interaction as the basis for communication. In Philip R. Cohen, Jerry L. Morgan, and Martha E. Pollack, editors, *Intentions in Communication*, pages 221–256. MIT Press, Cambridge, MA, 1990.

[19] P. R. Cohen, J. Morgan, and M. E. Pollack (editors). *Intentions in Communication.* MIT Press, 1990.

[20] S.E. Conry, K. Kuwabara, V.R. Lesser, and R.A. Meyer. Multistage negotiation for distributed satisfaction. *IEEE Transactions on Systems, Man, and Cybernetics, Special Issue on Distributed Artificial Intelligence*, 21(6):1462–1477, December 1991.

[21] D. Corkill. Hierarchical planning in a distributed environment. In *Proceedings of the Sixth International Joint Conference on Artificial Intelligence*, pages 168–175, Tokyo, August 1979.

[22] M. Dalal. Investigations into a theory of knowledge base revision: Preliminary report. In *AAAI*, pages 475–479, 1988.

[23] K. Decker and V. Lesser. A one-shot dynamic coordination algorithm for distributed sensor networks. In *Proc. of AAAI-93*, pages 210–216, 1993.

[24] K. Decker and V. Lesser. Designing a family of coordination algorithms. In *Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95)*, pages 73–80, San Francisco, CA, June 1995.

[25] D. C. Dennett. *The Intentional Stance.* The MIT Press, Cambridge, MA, 1987.

[26] J. Doyle. Some theories of reasoned assumptions: an essay in rational psychology. Technical Report 83-125, Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 1983.

[27] J. Doyle. Rational belief revision. In *Proc. of KR-91*, pages 163–174, 1991.

[28] J. Doyle, Y. Shoham, and M. Wellman. A logic of relative desire. In *Proc. of the 6th International Symposium on Methodologies for Intelligent Systems*, 1991.

[29] P. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming, and human's social and economical affairs. *Artificial Intelligence*, 77:321–357, 1995.

[30] E. H. Durfee and V. Lesser. Negotiating task decomposition and allocation using partial global planning. In L. Gasser and M. Huhns, editors, *Distributed Artificial Intelligence Volume II*, pages 229–244. Pitman Publishing: London and Morgan Kaufmann: San Mateo, CA, 1989.

[31] E. H. Durfee and V. R. Lesser. Partial global planning: a coordination framework for distributed hypothesis formation. *IEEE Trans. on Systems Man and Cybernetics*, 21(5):1167–1183, 1991.

[32] G. Dworman, S. O. Kimbrough, and J. D. Laing. Bargaining by artificial agents in two coalition games: A study in genetic programming for electronic commerce. In *Proc. of AAAI Genetic Programming Conference*, 1996.

[33] J. Elgot-Drapkin and D. Perlis. Reasoning situated in time I: Basic concepts. *Journal of Experimental and Theoretical Artificial Intelligence*, 2(1):75–98, 1990.

[34] M. Elvang-Goransson, P. Krause, and J. Fox. Dialectic reasoning with inconsistent information. In *Proceddings of the 9th Conference on Uncertainty in Artificial Intelligence*, 1993.

[35] E. Ephrati and J. S. Rosenschein. The Clarke tax as a consensus mechanism among automated agents. In *Proc. of AAAI-91*, pages 173–178, California, 1991.

[36] E. Ephrati and J. S. Rosenschein. Distributed consensus mechanisms for self-interested heterogeneous agents. In *Proceedings of International Conference on Intelligent and Cooperative Information Systems*, Rotterdam, Netherlands, May 1993.

[37] A. Evenchik. Inference system for argumentation in negotiation between automatic agents. M.Sc. thesis, Dept. of Mathematics and Computer Science, Bar-Ilan University, 1995.

[38] R. Fagin and J. Halpern. Belief, awareness, and limited reasoning. *Artificial Intelligence*, 34:39—76, 1988.

[39] R. Fagin, J. Halpern, and M. Y. Vardi. A nonstandard approach to the logical omniscience problem. *Artificial Intelligence*, 79(2):203–240, 1995.

[40] R. Fagin and M. Vardi. Knowledge and implicit knowledge in distributed environment: Preliminary report. In J. Y. Halpern, editor, *Proceedings of the First Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 187–206. Morgan-Kaufmann, Monterey, CA, 1986.

[41] J. Farrell. Meaning and credibility in cheap-talk games. In M. Dempster, editor, *Mathematical Models in Economics*. Oxford University Press, 1988.

[42] G. Ferguson and J. F. Allen. Arguing about plans: Plan representation and reasoning for mixed-initiative planning. In *Proceedings of the Second International Conference on AI Planning Systems*, pages 43–48, 1994.

[43] R. E. Fikes and N. J. Nilsson. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence*, 2:189–208, 1971.

[44] M. Fisher. Representing and executing agent-based systems. In *Intelligent Agents*, Lecture Notes in Artificial Intelligence No. 890, pages 307–323. Springer-Verlag, 1995.

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

[46] D. Fudenberg and J. Tirole. *Game Theory*. The MIT Press, 1991.

[47] J. W. Garson. Quantification in modal logic. In D. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic II*, pages 249–307. D. Reidel Publishing Company, 1984.

[48] L. Gasser. Social concepts of knowledge and action: DAI foundations and open systems semantics. *Artificial Intelligence*, 47(1–3):107–138, 1991.

[49] H. Geffner and J. Pearl. A framework for reasoning with defaults. In H. E. Kyburg, Jr., R. Loui, and G. Carlson, editors, *Knowledge Representation and Defeasible Reasoning*, pages 245–26. Kluwer Academic Press, Dordrecht, Netherlands, 1990.

[50] M. Georgeff. Communication and interaction in multi-agent planning. In *Proceedings of IJCAI-83*, pages 125–129, Karlsruhe, West Germany, 1983.

[51] M. A. Georgeff. Theory of action for multi-agent planning. In *Proceedings of AAAI-84*, pages 121–125, Austin, TX, 1984. AAAI.

[52] H. Gilkinson, S. F. Paulson, and D. E. Sikkink. Effects of order and authority in an argumentative speech. *Quarterly Journal of Speech*, 40:183–192, 1954.

[53] M. L. Ginsberg. Counterfactuals. *Artificial Intelligence*, 30(1):35–79, 1986.

[54] S. Grossman and M. Perry. Perfect sequential equilibrium. *Journal of economic theory*, 39:97–119, 1986.

[55] B. Grosz and S. Kraus. Collaborative plans for group activities. In *IJCAI-93*, pages 367–373, Chambery, France, 1993.

[56] B. J. Grosz and S. Kraus. Collaborative plans for complex group activities. *Artificial Intelligence Journal*, 86(2):269–357, 1996.

[57] B. J. Grosz and S. Kraus. The evolution of SharedPlans. In A. Rao and M. Wooldridge, editors, *Foundations and Theories of Rational Agency.* Kluwer Academic Publishers, 1998. (to appear).

[58] S. Hagg and F. Ygge. An architecture for agent-oriented programming with a programmable model of interaction. In *Proceeding of AICS94*, Dublin, Ireland, 1994.

[59] J. Y. Halpern and Y. Moses. A guide to completeness and complexity for model logics of knowledge and belief. *Artificial Intelligence*, 54(3):319–379, 1992.

[60] J. Y. Halpern and Yoram Moses. Knowledge and common knowledge in a distributed environment. *Journal of the Association for Computing Machinery*, 37(3):549–587, 1990.

[61] R. Hamill, T. D. Wilson, and R. E. Nisbett. Insensitivity to sample bias: Generalizing from atypical cases. *Journal of Personality and Social Psychology*, 39:579–589, 1980.

[62] B. Hayes-Roth. Agents on stage: Advancing the state of the art in AI. In *Proc. of IJCAI95*, pages 967–971, Montreal, Canada, August 1995.

[63] F. Horty and R. Thomason. Mixing strict and defeasible inheritance. In *Proceedings AAAI-88*, pages 427–432, 1988.

[64] A. J. I. Jones. Toward a formal theory of communication and speach acts. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication*, pages 161–185. MIT Press, 1990.

[65] M. Karlins and H. I. Abelson. *Persuasion: How opinions and attitudes are changed.* Springer Publishing Company, Inc., second edition, 1970.

[66] H. Katsuno and A. Mendelzon. Knowledge Base Revision and Minimal Change. *Artificial Intelligence*, 52:263–294, 1991.

[67] M. J. Katz and J. S. Rosenschein. Verifying plans for multiple agents. *Journal of Experimental and Theoretical Artificial Intelligence*, 5:39–56, 1993.

[68] D. Kinny and M. Georgeff. Commitment and effectiveness of situated agents. In *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, IJCAI-91*, pages 82–88, Sydney, Australia, 1991.

[69] G. Kiss and H. Reichgelt. Towards a semantics of desires. In E. Werner and Y. Demazeau, editors, *Decentralized Artificial Intelligence, Volume 3*, pages 115–128, Germany, 1992. Elsevier Science Publishers.

[70] R. R. Jr. Koballa. Persuading teachers to reexamine the innovative elementary science programs of yesterday: The effect of anecdotal versus data-summary communications. *Journal of Research in Science Teaching*, 23:437–449, 1986.

[71] K. Konolige. *A Deduction Model of Belief.* Pitman, London, 1986.

[72] K. Konolige. Hierarchic autoepistemic theories for nonmonotonic reasoning. In *Proceedings of AAAI-88*, pages 42–59, Saint Paul, MN, 1988. AAAI.

[73] K. Konolige and M. E. Pollack. A representationalist theory of intention. In *Proc. of IJCAI-93*, pages 390–395, 1993.

[74] S. Kraus. Beliefs, time and incomplete information in multiple encounter negotiations among autonomous agents. *Annals of Mathematics and Artificial Intelligence*, 20(1-4):111–159, 1997.

[75] S. Kraus and D. Lehmann. Knowledge, belief and time. *Theoretical Computer Science*, 58:155–174, 1988.

[76] S. Kraus and D. Lehmann. Designing and building a negotiating automated agent. *Computational Intelligence*, 11(1):132–171, 1995.

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

[78] S. Kraus, J. Wilkenfeld, and G. Zlotkin. Multiagent negotiation under time constraints. *Artificial Intelligence*, 75(2):297–345, 1995.

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

[80] D. Kreps and R. Wilson. Sequential equilibria. *Econometrica*, 50:863–894, 1982.

[81] Saul Kripke. Semantical considerations on modal logic. *Acta Philosophica Fennica*, 16:83–94, 1963.

[82] B. Laasri, H. Laasri, and V. Lesser. Negotiation and its role in cooperative distributed problem solving. In *10th International Workshop on DAI*, Texas, 1990. Chapter 9.

[83] S. Lander and V. Lesser. Undertanding the role of negotiation in distributed search among heterogeneous agents. In *Proceedings of the 12th International workshop on DAI*, Hidden Valley, PA., 1993.

[84] H. Levesque. A logic of implicit and explicit belief. In *Proc. of AAAI-84*, pages 198–202, Austin, TX, 1984.

[85] H. Levesque, P. Cohen, and J. Nunes. On acting together. In *Proceedings of AAAI-90*, pages 94–99, Boston, MA, 1990.

[86] M. Lewis and K. Sycara. Reaching informed agreement in multi-specialist cooperation. *Group Decision and Negotiation*, 2(3):279–300, 1993.

[87] F. Lin and Y. Shoham. Argument systems: a uniform basis for nonmonotonic reasoning. In *Proceedings First International Conf. on Knowledge Representation and Reasoning*, pages 245–255, 1989.

[88] J. S. Liu and K. P. Sycara. Distributed meeting scheduling. In *Proceedings of the Sixteenth Annual conference of the Cognitive Science Society*, Atlanta, Ga, August 1994.

[89] J.S. Liu and K. Sycara. Coordination of multiple agents for production management. *Annals of Operations Research*, 75:235–289, 1997.

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

[91] R. Loui. Defeat among arguments II. Technical Report WUCS-89-06, Department of Computer Science, Washington University, St. Louis, MO, 1989.

[92] R. Loui. A design for reasoning with policies, precedents, and rationales. In *Proceedgins of 4th International Conf. on AI and Law*, pages 202–211, 1993.

[93] R. Loui. Argument and arbitration games, working notes of the workshop on computational dialectics. In *Proceedings of AAAI-94*, pages 72–83, 1994.

[94] R. Loui and J. Norman. Rationales and argument moves. *Artificial Intelligence and Law*, 3(3):159–189, 1995.

[95] T. W. Malone. Modeling coordination in organizations and markets. *Management Science*, 33:1317–1332, 1987.

[96] J. Martins and S. Shapiro. A model for belief revision. *Artificial Intelligence*, 35:25–79, 1988.

[97] S. Matwin, S. Szpakowicz, Z. Koperczak, G. E. Kersten, and W. Michalowski. Negoplan: An expert system shell for negotiation support. *IEEE Expert*, 4(4):50–62, 1989.

[98] F. McCabe and K. Clark. April: Agent PRocess Interaction Language. In *Intelligent Agents*, Lecture Notes in Artificial Intelligence No. 890, pages 324–340. Springer-Verlag, 1995.

[99] J.-J.C. Meyer. Modal logics for knowledge representation. In *Linguistic Instruments in Knowledge Engineering. Proceedings of the 1991 Workshop*, pages 251–275, 1992.

[100] R. B. Myerson. Credible negotiation statements and coherent plans. *Journal of economic theory*, 48:264–303, 1989.

[101] R. B. Myerson. *Game Theory: Analysis of Conflict.* Harvard University Press, 1991.

[102] J. F. Nash. Noncooperative games. *Annals of Mathematics*, 54:289–295, 1951.

[103] B. Nebel. A knowledge level analysis of belief revision. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 301–311. Morgan Kaufmann, May 1989.

[104] N. J. Nilsson. *Principles of Artificial Intelligence.* Morgan Kaufmann Publishers Inc., 1980.

[105] M. Nirkhe, S. Kraus, and D. Perlis. Situated reasoning within tight deadlines and realistic space and computation bounds. In *Proceeedings of the Second Symposium On Logical Formalizations Of Commonsense Reasoning*, 1993.

[106] M. Nirkhe, S. Kraus, and D. Perlis. Thinking takes time: A modal active-logic for reasoning in time. In *Proc. of BISFAI-95*, 1995.

[107] R. E. Nisbett, E. Borgida, R. Crandall, and H. Reed. Popular induction: information is not necessarily informative. In J. S. Carroll and J. W. Payne, editors, *Cognition and social behavior*. Lawrence Erlbaum, 1976.

[108] D. Nute. A non-monotonic logic based on conditional logic. Technical Report ACMC 01-0007, Advanced Computational Methods Center, University of Georgia, Athens, GA, 1986.

[109] D. Nute. Defeasible reasoning and decision support systems. *Decision Support Systems*, 4:97–110, 1988.

[110] Daniel J. O'Keefe. *Persuasion: Theory and Research.* SAGE Publications, 1990.

[111] J. Oliver. *An Automated Negotiation and Electronic Commerce.* PhD thesis, Univ. of Pennsylvania, Philadelphia, PA, 1996.

[112] S. Parsons and N. R. Jennings. Negotiation through argumentation–a preliminary report. In *Proceedings of Second International Conference on Multi-Agent Systems*, pages 267–274, 1996.

[113] E. Pednault. Formulating multi-agent dynamic world problems in the classical planning paradigm. In *Reasoning About Actions & Plans — Proceedings of the 1986 Workshop*, pages 47–82. Morgan Kaufmann Publishers: San Mateo, CA, 1986.

[114] M. Pollack. The uses of plans. *Artificial Intelligence*, 57(1):43–68, 1992.

[115] M. E. Pollack. Plans as complex mental attitudes. In P. R. Cohen, J. Morgan, and M. E. Pollack, editors, *Intentions in Communication*, pages 77–103. MIT Press, 1990.

[116] J. L. Pollock. Defeasible reasoning. *Cognitive Science*, 11(4):481–518, 1987.

[117] J. L. Pollock. Interest driven suppositional reasoning. *J. Autom. Reasoning*, 6:419–461, 1990.

[118] J. L. Pollock. Self-defeating arguments. *Minds Mach.*, 1:367–392, 1991.

[119] J. L. Pollock. A theory of defeasible reasoning. *International Journal of Intelligent Systems*, 6:33–54, 1991.

[120] J. L. Pollock. How to reason defeasibly. *Artificial Intelligence*, 57(1–42), 1992.

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

[122] D. Poole. On the comparison of theories: preferring the most specific explanation. In *Proceedings of IJCAI-85*, Proceedings of IJCAI-85, pages 144–147, 1985.

[123] M. V. N. Prasad, V. R. Lesser, and S. E. Lander. Learning organizational roles for negotiated search in a multi-agent system. *Special issue on Evolution and Learning in Multiagent Systems in the International Journal of Human-Computer Studies (IJHCS)*, 1997. To appear.

[124] D. G. Pruitt. *Negotiation Behavior.* Academic Press, New York, N.Y., 1981.

[125] A. Rao and N. Y. Foo. Formal theories of belief revision. In *Proceedings of the First International Conference on Principles of Knowledge Representation and Reasoning*, pages 369–380. Morgan Kaufmann, May 1989.

[126] A. Rao and M. Georgeff. Modeling rational agents within BDI architecture. In *Proc. of the Second International Conference of Knowledge Representation*, pages 473–484, San Mateo, 1991. Morgan Kaufman Publishers.

[127] A. Rao and M. Georgeff. A model-theoretic approach to the verification of situated reasoning systems. In *IJCAI-93*, pages 318–324, French, 1993.

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

[129] A. Rao and M. P Georgeff. Asymmetry thesis and side-effect problems in linear-time and branching-time intention logics. In *Proc. of IJCAI-91*, pages 498–504, Australia, 1991.

[130] A. Rao, K. Ramamohanarao, and D. Weerasooriya. Design of a concurrent agent-oriented language. In *Intelligent Agents*, Lecture Notes in Artificial Intelligence No. 890, pages 386–401. Springer-Verlag, 1995.

[131] R. Reiter. A logic for default reasoning. *Artifical Intelligence*, 13:81–132, 1980.

[132] E. Rissland and K. Ashley. A case-based system for trade secrets law. In *Proceedings of 1987 ACM International Conf. on AI and Law*, 1987.

[133] J. S. Rosenschein. Synchronization of multi-agent plans. In A. H. Bond and L. Gasser, editors, *Readings in Distributed Artificial Intelligence*, pages 187–191. Morgan Kaufmann Publishers, Inc., San Mateo, California, 1988.

[134] J. S. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiation Among Computers.* MIT Press, Boston, 1994.

[135] A. Rubinstein. Perfect equilibrium in a bargaining model. *Econometrica*, 50:97–109, 1982.

[136] R. Schwartz and S. Kraus. Negotiation on data allocation in multi-agent environments. In *Proc. of AAAI-97*, pages 29–35, Providence, Rhode Island, 1997.

[137] R. Selten. Reexamination of the perfectness concept for equilibrium points in extensive games. *Internatinoal Journal of Game Theory*, 4:25–55, 1975.

[138] Y. Shoham. Agent oriented programing. *Artificial Intelligence*, 60(1):51–92, 1993.

[139] G. Simari and R. Loui. A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, 53:125–157, 1992.

[140] D. Skalak and E. Rissland. Argument moves in a rule-guided domain. In *Proceedings of ACM International Conf. on AI and Law*, 1991.

[141] R. G. Smith. The Contract net: A formalism for the control of distributed problem solving. In *Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI-77)*, 1977.

[142] R.G. Smith and R. Davis. Negotiation as a metaphor for distributed problem solving. *Artificial Intelligence*, 20:63–109, 1983.

[143] E. Sonenberg, G. Tidhar, E. Werner, D. Kinny, M. Ljungberg, and A. Rao. Planned team activity. Technical Report 26, Australian Artificial Intelligence Institute, Australia, 1992.

[144] A. Stollman. Negotiation methods for automatic agents. Master's thesis, Bar-Ilan University, Ramat-Gan, Israel, 1997.

[145] S. R. Sutton. Fear-arousing communications: A critical examination of theory and research. In J. R. Eiser, editor, *Social psychology and behavioral medicine.* Wiley, 1982.

[146] K. P. Sycara. *Resolving Adversarial Conflicts: An Approach to Integrating Case-Based and Analytic Methods.* PhD thesis, School of Information and Computer Science, Georgia Institute of Technology, 1987.

[147] K. P. Sycara. Resolving goal conflicts via negotiation. In *Proceeding, AAAI-88*, pages 245–250, 1988.

[148] K. P. Sycara. Persuasive argumentation in negotiation. *Theory and Decision*, 28:203–242, 1990.

[149] K. P. Sycara and C. M. Lewis. Modeling group decision and negotiation in concurrent product design. *Systems Automation: Research and Applications*, 1(3), December 1991.

[150] M. Tambe. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7:83–124, 1997.

[151] S. E. Taylor and S. C. Thompsons. Staking the elusive vividness effect. *Psychological Review*, 89:155–181, 1982.

[152] B. Thomas, Y. Shoham, A. Schwartz, and S. Kraus. Preliminary thoughts on an agent description language. *International Journal of Intelligent Systems*, 6(5):497–508, August 1991.

[153] S. R. Thomas. *PLACA, An Agent Oriented Programming Language.* PhD thesis, Computer Science Department, Stanford University, Stanford, CA, 1993. Also available as Technical Report No. STAN-CS-93-1487, September 1993.

[154] G. Tidhar, A. Rao, and E. Sonenberg. Guided team selection. In *Proceedings of Second International Conference on Multi-Agent Systems*, pages 369–376, 1996.

[155] Jean Tirole. *The Theory of Industrial Organization.* The MIT Press, 1988.

[156] S. Toulmin, R. Rieke, and A. Janik. *An introduction to reasoning.* Macmillan Publishing Co., Inc., 1979.

[157] M. Vardi. On the complexity of epistemic reasoning. In *Proceedings of the 4th Annual Symposium on Logic in Computer Science*, 1989.

[158] B. Vermazen. Objects of intention. *Philosophical Studies*, 71:223–265, 1993.

[159] G. A. W. Vreeswijk. Abstract argumentation systems: preliminary report. In *Proceedings of First World Conference on the Fundamentals of Artificial Intelligence*, pages 501–510, Angkor, Paris, 1991.

[160] G. A. W. Vreeswijk. Abstract argumentation systems. *Artificial Intelligence*, 90(1–2):225–279, 1997.

[161] P. Wavish. Exploiting emergent behavior in multi-agent systems. In E. Werner and Y. Demazeau, editors, *Decentralized Artificial Intelligence, Volume 3*, pages 297–310. Elsevier Science Publishers, Germany, 1992.

[162] M. Wellman and J. Doyle. Preferential semantics for goals. In *Proc. of AAAI-91*, pages 698–703, California, 1991.

[163] K. Werkman. Multiple agent cooperative design evaluation using negotiation. In *Proceedings of the Second International Conference on AI in Design*, Pittsburgh, PA, June 1992.

[164] E. Werner. Toward a theory of communication and cooperation for multiagent planning. In *Proceedings of the Second Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 129–143, Pacific Grove, California, March 1988.

[165] G. R. Williams. Style and effectiveness in negotiation. In L. Hall, editor, *Negotiation : strategies for mutual gain : the basic seminar of the Harvard Program on Negotiation*. Sage, Newbury Park, California, 1993.

[166] M. Winslett. Is belief revision harder than you thought? In *Proceedings of AAAI-86*, pages 421–427, Philadelphia, 1986.

[167] M. J. Wooldridge and N. R. Jennings. Agent theories, architectures and languages: A survey. In *Intelligent Agents*, Lecture Notes in Artificial Intelligence No. 890, pages 1–39. Springer-Verlag, 1995.

[168] D. Zeng and K. P. Sycara. Bayesiam learning in negotiation. In *Proceedings of the AAAI Stanford Spring Symposium on Adaptation, Co-evolution and Learning in Multi-Agent Systems*, 1996.

[169] Dajun Zeng and Katia Sycara. Bayesian learning in negotiation. *International Journal of Human-Computer Studies*, 48:125–141, 1998.

[170] G. Zlotkin and J. S. Rosenschein. Cooperation and conflict resolution via negotiation among autonomous agents in noncooperative domains. *IEEE Transactions on Systems, Man, and Cybernetics, Special Issue on Distributed Artificial Intelligence*, 21(6):1317–1324, December 1991.

[171] G. Zlotkin and J. S. Rosenschein. Mechanism design for automated negotiation, and its application to task oriented domains. *Artificial Intelligence*, 86(2):195–244, 1996.

[172] G. Zlotkin and J. S. Rosenschein. Mechanisms for automated negotiation in state oriented domains. *Journal of Artificial Intelligence Research*, 5:163–238, 1996.

List of Figures

List of Tables