An Agent Communication Protocol for Resolving Conflicts (Extended Abstract)

Jamal Bentahar
CIISE, Concordia University
1515 Ste-Catherine Street
West, EV7.640
Montreal, Canada
bentahar@ciise.concordia.ca

ABSTRACT

This paper proposes an agent communication protocol specifie as a set of dialogue rules for resolving conflict using assumption-based argumentation. Arguments are built from a set of rules and assumptions using backward deduction. Beyond arguments agents can handle, we propose the notion of *partial arguments* along with *partial acceptability*. This protocol merges inquiry and persuasion stages and by building and reasoning about partial arguments, agents can jointly fin arguments supporting a new solution for their conflict which is not known by any of them individually.

Categories and Subject Descriptors

I.2.4 [Knowledge Representation Formalisms and Methods]

General Terms

Design, Languages, Theory

Keywords

Agent communication, argumentation

1. ARGUMENTATION MODEL

This paper focuses on using argumentation for persuasion and inquiry for agent communication. In multi-agent systems, some interesting argumentation-based protocols for these dialogues have been proposed [1, 2, 4, 6]. However, they are either pure persuasion or pure inquiry and are not complete in the sense of pre-determinism. Except a few proposals such as [4], the notion of agents' strategies on how to use these protocols is disregarded [3]. The purpose of this paper is to address these limitations and its main contribution is the proposition of a new sound and complete agent communication protocol combining persuasion and inquiry for conflic resolution.

In this paper, we use a formal language $\mathcal{L}$ to express agents' beliefs. This language consists of countably many sentences (or wffs). The language is associated with an abstract *contrary mapping* like the one used in [5] (the negation is an example of this mapping). We do not assume this mapping to be necessarily symmetric. $\bar{x}$ denotes an arbitrary contrary of a wff x. The set $Arg(\mathcal{L})$ contains all arguments, which will be denoted as (X, c) or simply by $a, b, d, \ldots$. An argument a supporting c will be denoted $a \uparrow c$.

Cite as: An Agent Communication Protocol for Resolving Conflict (Extended Abstract), Jamal Bentahar, *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Lespérance, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. 1557-1558

Definition 1. Acceptable Arguments Let $Ar \subseteq Arg(\mathcal{L})$ be a set of arguments, and G the grounded extension in the argumentation framework. An argument a over Ar is acceptable iff $a \in G$.

Definition 2. Preferred Semi Acceptable Arguments Let G be the grounded extension in the argumentation framework, and $E_1, \ldots, E_n$ the preferred extensions. An argument a is:

- Semi-acceptable iff 1) $a \notin G$ and $\exists\, E_i, E_j$ with $(1 \leq i, j \leq n)$ such that $a \in E_i \wedge a \notin E_j$; or 2) $E_1 = \cdots = E_n = \emptyset$.
- Preferred semi-acceptable iff $a \notin G$ and $\forall E_i$ $(1 \leq i \leq n)$ $a \in E_i$.

Definition 3. Eliminated Arguments An argument is eliminated iff it does not belong to any preferred extension in the argumentation framework.

Definition 4. Con icts Let α and β be two argumentative agents and AF_α and AF_β their respective argumentation frameworks. These two frameworks share the same contrary mapping and the same rules, but not necessarily the same assumptions. There is a conflic between α and β iff one of them (e.g., α) has an acceptable argument (X, c) relative to AF_α and the other (i.e., β) has an acceptable argument for a contrary of c $(X', \bar{c})$ relative to AF_β. We denote this conflic by $\alpha_c \ncong \beta_{\bar{c}}$ $(\bar{c} \in^- c)$.

2. PARTIAL ARGUMENTS

The outcome of an interaction aiming to resolve a conflic depends on the status of the formula representing the conflic topic. A wff has four statuses depending on the statuses of the arguments supporting it. Let St be the set of these statuses. We defin the following function that returns the status of a wff with respect to a set of arguments: $\Delta : \mathcal{L} \times 2^{Ar} \rightarrow St$

To resolve conflicts it happens that agents do not have complete information. In similar situations, they can build *partial arguments* for some conclusions out of their beliefs. The set $Arg(\mathcal{L})$ contains now not only all the arguments, but also all the partial arguments agents can build. We defin a partial argument as follows:

Definition 5. Partial Arguments Let $X \subseteq \mathcal{A}$ be a consistent subset of assumptions, and c a sentence in $\mathcal{L}$. A partial argument in favor of c is a pair denoted by $(X, c)_\partial$ such that $\exists\, Y \subseteq \mathcal{A}$ $(Y \neq \emptyset)$ and $(X \cup Y, c)$ is an argument.

As for arguments, we need to defin the *status of partial arguments*. The idea is that if, considering the information an agent has at the current moment, there is no chance for the partial argument to be acceptable or at least to change the status of already uttered arguments, then there is no need to try to build such a partial argument. When the internal structure of these partial arguments is not needed, we use the notations a_∂ and a_∂^Y to denote a partial argument and a partial argument that can be completed by the set Y of assumptions respectively. The argument obtained by adding the assumptions Y to the partial argument a_∂^Y is denoted by $a_\partial.Y$. The following definitio establishes the status of partial arguments.

Definition 6. Status of Partial Arguments Let $Ar \subseteq Arg(\mathcal{L})$ be a set of arguments and partial arguments over the argumentation framework. A partial argument a_∂ over Ar is acceptable ((preferred) semi-acceptable, eliminated) iff $\exists\, Y : a_\partial.Y \in Ar$ is acceptable ((preferred) semi-acceptable, eliminated).

3. PROTOCOL

Agents share the same set of rules and each agent has a possibly inconsistent belief base $\mathcal{A}_\alpha$ and $\mathcal{A}_\beta$ respectively containing assumptions. Agents strategies are based on their argumentation systems. Each agent α has a commitment store (CS_α). In general, commitment stores can contain arguments, partial arguments and sentences agents exchange during their interactions. Our protocol is to be used when a conflic is identified for example as an outcome of a previous interaction.

The set of arguments an agent can build from the union of commitment stores is denoted by $Arg(\cup CS)$. The possibility for an agent α to build an acceptable argument a (respectively an acceptable partial argument a_∂^Y) from its knowledge base and the commitment store of the receiver β is denoted by $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta) \rhd a$ (respectively $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta) \rhd a_\partial^Y$). $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta) \not\rhd a$ (respectively $\mathcal{AR}(\mathcal{A}_\alpha \cup CS_\beta) \not\rhd a_\partial^Y$) means that agent α cannot build an acceptable argument a (respectively an acceptable partial argument a_∂^Y) from $\mathcal{A}_\alpha \cup CS_\beta$. For simplificatio reason, we associate the same symbols ($\unrhd$ and $\not\unrhd$) with (partial) preferred semi-acceptable and (partial) semi-acceptable arguments. However, agents consider firs (partial) preferred semi-acceptable arguments. We assume that statuses are ordered using $\succ$ ordering relation.

Definition 7. Protocol A protocol is a pair $\langle \mathcal{C}, \mathcal{D} \rangle$ with $\mathcal{C}$ a finit set of allowed moves and $\mathcal{D}$ a set of dialogue rules.

The moves in $\mathcal{C}$ are of n different types ($n > 0$). We denote by $M^i(\alpha, \beta, a, t)$ a move of type i played by agent α and addressed to agent β at time t regarding a content a. We consider 5 types of moves: $Assert$, $Accept$, $Attack$, $Question$ ad $Stop$. Generally, in the persuasion protocol agents exchange arguments. Except the $Question$ move whose content is not an argument, the content of other moves is an argument a ($a \in Arg(\mathcal{L})$). When replying to a $Question$ move, the content of $Assert$ move can also be a partial argument or "?" when the agent does not know the answer.

Intuitively, a dialogue rule in $\mathcal{D}$ is a rule indicating the possible moves that an agent could play following a move done by a receiver. To make agents deterministic, we specify these rules using conditions that reflec the agents' strategies. Each condition C_k is associated with a single reply. This is specifie formally as follows:

Definition 8. Dialogue Rule It has either the form:

$$\bigwedge_{\substack{0<k\leq n_i \\ j\in J}} \big(M^i(\alpha, \beta, a, t) \wedge C_k \Rightarrow M_k^j(\beta, \alpha, a_k, t')\big)$$

where J is the set of move types, M^i and M^j are in $\mathcal{C}$ (M_k^j refers to the move of type j, which is related to the condition C_k), $t < t'$ and n_i is the number of allowed communicative acts that β could perform after receiving a move of type i from α; or of the form:

$$\bigwedge_{\substack{0<k\leq n \\ j\in J}} \big(C_k \Rightarrow M_k^j(\alpha, \beta, a_k, t_0)\big)$$

where t_0 is the initial time and n is the number of allowed moves that α could play initially.

Some examples of the dialogue rules of our protocol are:

1 Initial Rule

$C_{in_1} \Rightarrow Assert(\alpha, \beta, a)$

where:

$$C_{in_1} = \exists\, p, q \in \mathcal{L} : \alpha_p \not\cong \beta_q \wedge \mathcal{AR}(\mathcal{A}_\alpha) \rhd a \wedge a \uparrow p$$

2 Assertion Rule

$$\begin{aligned}
&Assert(\alpha, \beta, \mu) \wedge C_{as_1} \Rightarrow Attack(\beta, \alpha, b) && \wedge \\
&Assert(\alpha, \beta, \mu) \wedge C_{as_2} \Rightarrow Attack(\beta, \alpha, b') && \wedge \\
&Assert(\alpha, \beta, \nu) \wedge C_{as_3} \Rightarrow Question(\beta, \alpha, Y) && \wedge \\
&Assert(\alpha, \beta, \nu) \wedge C_{as_4} \Rightarrow Question(\beta, \alpha, Y') && \wedge \\
&Assert(\alpha, \beta, \nu) \wedge C_{as_5} \Rightarrow Accept(\beta, \alpha, a) && \wedge \\
&Assert(\alpha, \beta, \nu) \wedge C_{as_6} \Rightarrow Stop(\beta, \alpha)
\end{aligned}$$

where μ is an argument or partial argument, ν is an argument, partial argument, or "?" and:

$$\begin{aligned}
C_{as_1} = {} & \exists\, b \in Ar : \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd b \\
& \wedge \alpha_p \Rightarrow \Delta(p, Arg(\cup CS)) \succ \Delta(p, Arg(\cup CS \cup \{b\})) \\
& \wedge \beta_p \Rightarrow \Delta(p, Arg(\cup CS \cup \{b\})) \succ \Delta(p, Arg(\cup CS)) \\
C_{as_2} = {} & \neg C_{as_1} \wedge \exists\, b' \in Ar : \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \unrhd b' \\
& \wedge \alpha_p \Rightarrow \Delta(p, Arg(\cup CS)) \succ \Delta(p, Arg(\cup CS \cup \{b'\})) \\
& \wedge \beta_p \Rightarrow \Delta(p, Arg(\cup CS \cup \{b'\})) \succ \Delta(p, Arg(\cup CS)) \\
C_{as_3} = {} & \neg C_{as_1} \wedge \neg C_{as_2} \\
& \wedge \exists\, b_\partial, Y :\ b_\partial.Y \in Ar \wedge \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd b_\partial^Y \\
& \wedge \alpha_p \Rightarrow \\
& \quad \Delta(p, Arg(\cup CS)) \succ \Delta(p, Arg(\cup CS \cup \{b_\partial.Y\})) \\
& \wedge \beta_p \Rightarrow \\
& \quad \Delta(p, Arg(\cup CS)) \succ \Delta(p, Arg(\cup CS \cup \{b_\partial.Y\})) \\
C_{as_4} = {} & \neg C_{as_1} \wedge \neg C_{as_2} \wedge \neg C_{as_3} \\
& \wedge \exists\, b_\partial, Y' :\ b_\partial.Y' \in Ar \wedge \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \unrhd b_\partial^{Y'} \\
& \wedge \alpha_p \Rightarrow \\
& \quad \Delta(p, Arg(\cup CS)) \succ \Delta(p, Arg(\cup CS \cup \{b_\partial.Y'\})) \\
& \wedge \beta_p \Rightarrow \\
& \quad \Delta(p, Arg(\cup CS \cup \{b_\partial.Y'\})) \succ \Delta(p, Arg(\cup CS)) \\
C_{as_5} = {} & \exists\, a \in Ar : \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \rhd a \wedge a \uparrow p \\
& \wedge \neg C_{as_3} \wedge \neg C_{as_4} \\
C_{as_6} = {} & \neg C_{as_1} \wedge \neg C_{as_3} \wedge \neg C_{as_4} \wedge \neg C_{as_5} \\
& \wedge \forall\, b' \in Ar,\ \mathcal{AR}(\mathcal{A}_\beta \cup CS_\alpha) \unrhd b' \Rightarrow \\
& \qquad \Delta(p, Arg(\cup CS)) = \Delta(p, Arg(\cup CS \cup \{b'\}))
\end{aligned}$$

4. REFERENCES

[1] A. Artikis, M. Sergot, and J. Pitt. An executable specificatio of a formal argumentation protocol. *Artif. Intell.*, 171(10-15):776–804, 2007.

[2] K. Atkinson, T. Bench-Capon, and P. McBurney. A dialogue game protocol for multi-agent argument over proposals for action. *J. of Auton. Agents and Multi-Agent Sys.*, 11(2), 2005.

[3] J. Bentahar, B. Moulin, and M. Bélanger. A taxonomy of argumentation models used for knowledge representation. *Artif. Intell. Rev.*, DOI: 10.1007/s10462-010-9154-1, 2010.

[4] E. Black and A. Hunter. A generative inquiry dialogue system. In *The International Conference on Autonomous Agents and Multiagent Systems*. ACM Press, 2007.

[5] D. Gaertner and F. Toni. Computing arguments and attacks in assumption-based argumentation. *IEEE Int. Sys.*, 22(6):24–33, 2007.

[6] P. Pasquier, I. Rahwan, F. Dignum, and L. Sonenberg. Argumentation and Persuasion in the Cognitive Coherence Theory. In *The 1st Int. Conf. COMMA*. IOS Press, 2006.