한정된 자원을 가진 에이전트 환경에서의 충돌해결을 위한 협상기법

이명진[0], 김진상
계명대학교 컴퓨터공학부
mjlee@jinri.kmu.ac.kr[0], jskim@kmu.ac.kr

A Negotiation Mechanism for Resolving Conflicts in Resource-bounded Agents Environments

Myung-Jin Lee[0], Jin-Sang Kim
Faculty of Computer Engineering, Keimyung University

요 약

In most Multi-Agent Systems (MAS), agents are required to achieve their own goals. An agent's goals, however, can conflict with others either when agents compete with each other to achieve a common goal or when they have to use a set of limited resources to accomplish agents' divergent goals. In this paper, we consider that a BDI architecture, as a shorthand for belief, desire, and intention in any agent-based systems, is a core component of agents' mental attitudes and represent resource-bounded BDI agents in logic programming framework. We propose an algorithm in which BDI agents with different goals solve their problems through negotiation resolving goal conflicts.

1. Introduction

Research in MAS is concerned with the behavior of a collection of autonomous agents trying to solve given problems. Agents should ideally possess several abilities: perception and interpretation of incoming data and messages, reasoning based upon their beliefs, decision making, planning, and the ability to execute plans including message passing. Agents in MAS also need to interact with others since there exist inherent interdependencies among them, but inter-agent conflicts may arise because of two basic reasons: different agents have contrasting goals and they have inconsistent knowledge [6]. A conflict in belief may be the source of confusion in goal. *Negotiation* might be a promising mechanism for resolving these conflicts.

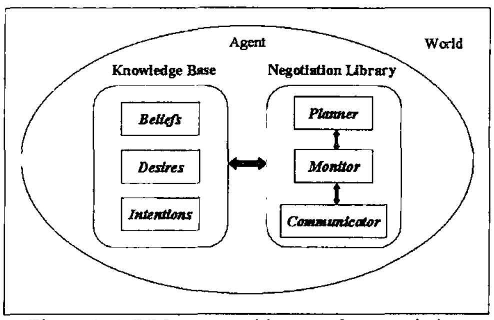
In this paper, we represent agents' *beliefs*, *desires*, and *intentions* (BDI) for MAS in logic programming and develop a negotiation mechanism for BDI agents. We then introduce an agent communication language (ACL) and a negotiation protocol for resource-bounded BDI agents and show how goal conflicts through negotiation in MAS are resolved using our ACL and negotiation mechanism in practice.

2. BDI Agent Architectures

An abstract characterization of BDI agents is assumed, in which some core capabilities of agents are described as *mental attitudes*. The FIPA ACL specification [4, 5] has characterized an agent as being able to be described as though it has mental attitudes of: *belief*, *uncertainty*, *intention*. Chalmers [1] has explored the declarative use of constraints within a BDI agent framework to represent knowledge as complex quantified constraints and to apply these techniques to a courier scenario where cooperation agents communicate, delegate, and exchange desires and information using generalized partial global planning mechanisms to solve a given set of tasks.

In general, an agent has a variety of knowledge to achieve its goal, to plan some task, and to communicate with other agents. Also, agents should have the capability of dealing with multi-interaction and communicating with others distributed by a network. In order to design such an agent, we consider a BDI agent architecture containing the following two major components: *knowledge base* and *negotiation library* including *planner*, *monitor*, and *communicator*.

The knowledge base is a set of logical sentences, which includes knowledge about the agent's capabilities and other agents', and rules for problem decomposition. On the other hand, the negotiation library is responsible for deciding how to solve each task, supervising the execution of tasks, and handling incoming and outgoing messages. Figure 1 illustrates our BDI agent architecture for negotiation.



<Figure 1. A BDI agent architecture for negotiation>

3. A Negotiation Model for Resource-bounded BDI Agents

We describe negotiation processes in MAS with resource-bounded BDI agents that do not have a complete state of knowledge about others and we propose a negotiation mechanism to resolve goal conflicts for BDI agents that can undertake negotiation by changing their beliefs.

3.1 An ACL for BDI Agents

Negotiation is achieved through the exchange of messages in a shared communication language. We use a variant of the FIPA ACL specification and the negotiation meta-language as our negotiation language.

The actual exchange of messages is driven by the participating agents' own needs, goals, or mental attitudes. In this case, we can define a communication language *CL*, communicative acts, for BDI agents as follows:

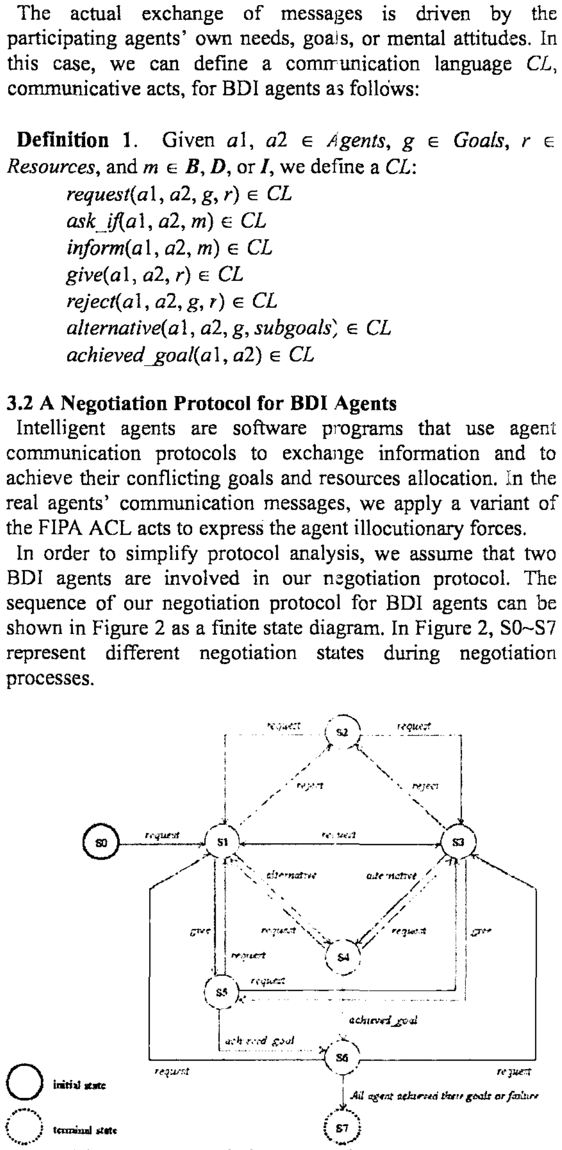
Definition 1. Given $a1$, $a2 \in Agents$, $g \in Goals$, $r \in Resources$, and $m \in \boldsymbol{B}, \boldsymbol{D}$, or $\boldsymbol{I}$, we define a *CL*:

request($a1$, $a2$, g, r) $\in CL$
ask_if($a1$, $a2$, m) $\in CL$
inform($a1$, $a2$, m) $\in CL$
give($a1$, $a2$, r) $\in CL$
reject($a1$, $a2$, g, r) $\in CL$
alternative($a1$, $a2$, g, *subgoals*) $\in CL$
achieved_goal($a1$, $a2$) $\in CL$

3.2 A Negotiation Protocol for BDI Agents

Intelligent agents are software programs that use agent communication protocols to exchange information and to achieve their conflicting goals and resources allocation. In the real agents' communication messages, we apply a variant of the FIPA ACL acts to express the agent illocutionary forces.

In order to simplify protocol analysis, we assume that two BDI agents are involved in our negotiation protocol. The sequence of our negotiation protocol for BDI agents can be shown in Figure 2 as a finite state diagram. In Figure 2, S0~S7 represent different negotiation states during negotiation processes.



<Figure 2. A negotiation protocol for BDI agents>

3.3 Interpretation and Generation of Messages

How an agent chooses which message to utter depends upon many factors: agent's theory, the active goals of the agent, or the history of the negotiation; and it also depends upon the way that particular agent interprets those messages. We define a message generation function *G* under the condition of negotiating agent about deficient resources.

Definition 2. We suppose that agent $a2$ has received a message from agent $a1$. Given an appropriate *CL*, $a2$ generates a message depending upon the following message generation function *G*:

If goal g satisfies the unacceptable conditions and there exists an alternative to achieving g,
G(*request*($a1$, $a2$, g, r)) = *alternative*($a2$, $a1$, g, *subgoals*)
If g satisfies the unacceptable conditions and there does not exist an alternative,
G(*request*($a1$, $a2$, g, r)) = *reject*($a2$, $a1$, g', r) or *make a counter-request*
If g does not satisfy the unacceptable conditions,
G(*request*($a1$, $a2$, g, r)) = *give*($a2$, $a1$, r)
If $a2$ has received an alternative,
G(*alternative*($a1$, $a2$, g, *subgoals*)) = *make a replanning*
If $a2$ has received resources r,
G(*give*($a1$, $a2$, r)) = *continue planning*
If a request was rejected by $a1$,
G(*reject*($a1$, $a2$, g, r)) = *search an alternative*
If $a1$ has notified its goal achievement,
G(*achieved_goal*($a1$, $a2$)) = *make a replanning*
If $a2$ was asked whether p is true or not,
G(*ask_if*($a1$, $a2$, p)) = *inform*($a2$, $a1$, p) or *inform* ($a2$, $a1$, *not p*)
If $a1$ informs $a2$ of the truth of p,
G(*inform*($a1$, $a2$, p)) = *continue planning*

4. Experiments and Comparisons

4.1 Experiments

We take an experiment platform which has Intel Pentium 4 1GHz processor and Windows Millennium Edition version 4.90. Our simple BDI agents' negotiation mechanism is tested on this platform using InterProlog supporting Java 2 SDK version 1.3.1 and XSB Prolog version 2.4.

To show how our negotiation mechanism might work in practice, we consider three home improvement BDI agents with different objectives and resources. Agent $a1$ has the intention of hanging a picture, *intend*(*do*($a1$, *hang_picture*)), and believes that it has in its possession a picture, a screw, a hammer, a hanger nail, and a screwdriver. It also believes that its name is $a1$ and agent $a3$ has a hanger.

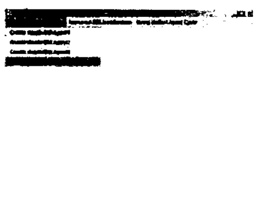
Agent $a2$ has the intention of hanging a mirror, *intend*(*do*($a2$, *hang_mirror*)), and believes that it has a mirror and a nail. It also believes that its name is $a2$ and agent $a1$ has a screw, a hammer, and a screwdriver. Finally, agent $a3$ has the intention of hanging a clock, *intend*(*do*($a3$, *hang_clock*)), and believes that it has a clock and a hanger. It also believes that its name is $a3$ and agent $a1$ has a hanger nail.

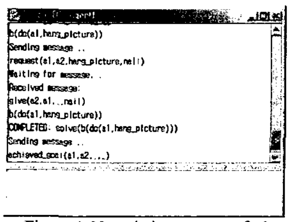
In this paper, we try to confine intentions to commit when the agent believes that it can take the action, i.e., $intend(a) \wedge can(a) \rightarrow does(a)$.

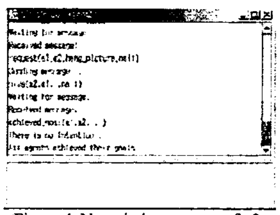
We construct BDI agents and allow them to negotiate with each other from the top-level window in Figure 3. `NegotiationWindow` class is responsible for creating this window, constructing the BDI agents, and starting a negotiation with the agents.

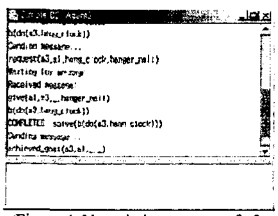
When we click *Create simple BDI Agent*1 item in Figure 3, `Agent1Window` class displays *Simple BDI agent*1 window with two panes. `Agent1Window` class then consults a routine of initializing agent *a*1 using *sendAndFlush* method in `PrologEngine` class. `Agent2Window` class and `Agent3Window` class perform similar functions to `Agent1Window` class.

When we click *Start simple BDI agents negotiation* item in Figure 3, negotiation among agent *a*1, *a*2, and a3 starts. At first, *a*1 tries to achieve its goal, *intend*(*do*(*a*1, *hang_picture*)). Thus, it tries to solve the query, ?-*solve*(*b*(*do*(*a*1, *hang_picture*))). This query is transferred to the monitor of *a*1 and again passed on to the planner of *a*1. The planner decomposes the query and comes to know that *a*1 needs a nail. *a*1 however has no knowledge about nail so that it asks *a*2 and *a*3 if they have a nail, *ask_if*(*a*1, *a*2, *b*(*have*(*a*2, *nail*))) and *ask_if*(*a*1, *a*3, *b*(*have*(*a*3, *nail*))), and then it waits for a reply through the communicator. Now each agent negotiates with one another and Figure 4, 5, and 6 show these negotiation processes.


<Figure 3. The top-level windows>


<Figure 4. Negotiation process of *a*1>


<Figure 4. Negotiation process of *a*2>


<Figure 4. Negotiation process of *a*3>

4.2 Comparisons

The argumentation system [7] goes through the process that it evaluates counter-agent's arguments included in received messages in its position. Through this process, the argumentation system evaluates the received message, making an argument for or against the received one. In this paper, if agent *a*2 receives a message, *request*(*a*1, *a*2, *g*, *r*), *a*2 knows the reason why *a*1 requests the resources *r*, i.e., because of the goal *g*. This fact helps an agent to reason about others' mental attitudes so that the agent may plan its goal more effectively.

Kasbah agents [2] may have conflicting goals and negotiate, changing suggested price using three functions of price decay or raise over time while our negotiation system negotiates changing agents' beliefs.

The CNP [3] does not allow agents to cause goal conflicts and a manager agent is responsible for all negotiations. Unlike the CNP, our negotiation mechanism for BDI agents supposes that agents may have conflicts. These conflicts occur when each agent has conflicting beliefs, desires, or intentions and when one agent has erroneous assumptions about another agent's knowledge. In this state, we show the capability of negotiation among agents without an agent playing manager role.

5. Conclusions and Future Research

In this paper, we have represented agents' beliefs, desires, and intentions for MAS in logic programming environments and we have introduced a negotiation mechanism for BDI agents. We have shown how goal conflicts through negotiation in MAS are resolved using our ACL and negotiation mechanism in practice.

This paper has raised the necessity of many types of specialized negotiation strategies. Existing communication languages consider additional communication actions such as *critique*, *threaten*, *reward*, *persuade*, and so on. Finally, we have assumed that different agents share the same communication language. Indeed, translator agents could be defined, acting as mediators between agents with different communication languages.

References

[1] S. Chalmers. "BDI Agents & Constraints". In *Proceedings of the Artificial Intelligence and the Simulation of Behavior Symposium on Information Agents for Electronic Commerce*, pages 61-71, 2001.

[2] A. Chavez and P. Maes. "Kasbah: An Agent Marketplace for Buying and Selling Goods". In *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agents Technology*, pages 75-90, 1996.

[3] R. Davis and R. G. Smith. Frameworks for Cooperation in Distributed Problem Solving. *Readings in Distributed Artificial Intelligence*, pages 61-70, 1980.

[4] FIPA Specification Part 2: Agent Communication Language. http://www.fipa.org/. *The Foundation for Intelligent Physical Agents* (FIPA), 1999.

[5] FIPA Communicative Act Library Specification. Http://www.fipa.org/specs/fipa00037/. *The Foundation for Intelligent Physical Agents* (FIPA), 2000.

[6] M. Garagnani, M. Fox, and D. P. Long. "Belief Systems for Conflict Resolution". In *Proceedings of the Thirteenth European Conference on Artificial Intelligence Workshop on Conflicts among Agents*, pages 55-60, 1998.

[7] S. Parsons, C. Sierra, and N. R. Jennings. "Agents that Reason and Negotiate by Arguing". *Journal of Logic and Computation*, 8(3):261-292, 1998.