퍼지 및 지능시스템학회 논문지 2003, Vol. 13, No. 2, pp. 192-199

협동적인 분산 환경에서 BDI 에이전트를 위한 협상 기법

A Negotiation Mechanism for BDI Agents in Distributed Cooperative Environments

이 명 진, 김 진 상
Myung-Jin Lee, Jin-Sang Kim

아시아대학교 인터넷비즈니스학과, 계명대학교 컴퓨터공학부

요 약

멀티에이전트 시스템에서 에이전트는 각자 달성해야 할 목표를 가지고 있다. 그러나 여러 에이전트들이 하나의 공통된 목표를 달성하기 위해 서로 경쟁을 하는 경우거나 혹은 각자의 서로 다른 목표를 달성하기 위해 제한된 자원을 사용해야 하는 경우 에이전트들은 서로 충돌할 수 있다. 충돌이 발생할 때 에이전트들은 각자의 목표를 달성하기 위해 다른 에이전트들과의 협상을 통해 목표의 충돌이 없는 일치 상황에 도달하도록 설계되어야 한다. 본 논문은 멀티에이전트 시스템에서 믿음(belief), 소망(desire), 그리고 의도(intention)로 설명되는 BDI 구조가 각 에이전트가 가져야 할 정신적 태도의 핵심 요소라고 가정하고, 이러한 구조를 가지는 BDI 에이전트를 논리 프로그래밍의 틀에서 표현한다. 또한 서로 다른 목표를 가진 BDI 에이전트들이 자원이 제한된 협동적인 분산 환경에서 상호간의 목표 충돌을 해결하기 위해 협상을 통해 각자의 문제를 해결하는 알고리즘을 제시한다. 마지막으로 본 논문에서 제안한 협상 알고리즘의 효과성을 검증하기 위하여 협상 메타 언어로 구현한 예제 문제의 실험 결과를 기술한다.

Abstract

Agents in multi-agent systems (MAS) are required to achieve their own goals. An agent's goal, 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 either case, agents need to be designed to reach a mutual acceptable state where they can avoid any goal conflicts through negotiation with others to achieve their goals. In this paper, we consider a BDI agent architecture where belief, desire, and intention are the three major components for agents' mental attitudes and represent resource-bounded BDI agents in logic programming framework. We propose a negotiation algorithm for BDI agents solving their problems without goal conflicts in distributed cooperative environments. Finally, we describe a simple scenario to show the effectiveness of the negotiation algorithm implemented in a negotiation meta-language.

Key Words : Multi-Agent Systems, Distributed cooperative environments, BDI Agent, Negotiation, Agent Communication Language, Logic Programming

1. Introduction

Research in Artificial Intelligence (AI) aims at developing software to simulate intelligent capabilities of human beings. MAS, as a subfield of AI, is concerned with the behaviors of a collection of autonomous agents trying to solve given problems that are beyond their individual capabilities. These agents are autonomous and may be heterogeneous in nature and need to interact with others since there exist inherent interdependencies among them, but inter-agent conflicts may arise. These conflicts concern which mental attitudes of the agents are in conflict. That is, two agents may hold beliefs that are in conflict and conflicting agents may have different mental constructs. A conflict in belief may be the source of confusion in goal. Negotiation might be a promising mechanism for resolving these conflicts.

Negotiation plays a fundamental role in human cooperative activities, allowing people to resolve conflicts that could interfere with cooperative behaviors. Durfee *et al.* [5] have defined negotiation as a process of reducing inconsistency and uncertainty and improving agreement on common viewpoints or plans through the structured exchanges of relevant information. The general aims of negotiation are modification of local agent plans, in the case of negative or harmful interactions, and identification of situations where potential interactions are possible [11]. Modification and identification situations trigger a process of negotiation in the sense that agents communicate in a certain way to reach a mutual agree-

접수일자 : 2002년 7월 19일
완료일자 : 2003년 1월 2일

ment state.

Contract Net Protocol (CNP) of David and Smith [4] views negotiation as an organizing principle used to effectively match tasks and agents. To start the CNP, however, there must not be a conflict at all among agents. Rosenshein and Zlotkin [15] have studied negotiation based on the game theory, but they have assumed that agents have complete knowledge about the payoff matrices. This means that agents have complete knowledge about the preferences of other agents. On the other hand, Parsons *et al.* [12] have proposed a formal framework, based on a system of argumentation, which permits agents to negotiate to establish acceptable ways to solve problems. However, the size of communication messages in this system is reasonably large since communication messages contain sender, receiver, message content, and arguments as message parameters.

In this paper, we consider rational agents with mental attitudes of belief, desire, and intention (BDI) on the logical point of view. We also consider a negotiation algorithm that has BDI predicates and two definite unacceptable conditions. These unacceptable conditions might occur when each agent has conflicting beliefs, desires, or intentions or when an agent has erroneous assumptions about another agent's mental attitudes. We present a logic programming framework for negotiation in which each resource-bounded agent is represented on the basis of belief, desire, and intention. Further, we suggest a negotiation algorithm to resolve goal conflicts accompanied with each agent's problem solving activities in cooperative MAS. We show the applicability of the proposed negotiation algorithm in the context of a simple example. It is tested on InterProlog [9], a Java front-end and functional enhancement for Prolog, based on a variant of the Foundation for Intelligent Physical Agents (FIFA) ACL specification [6, 8], FIPA interaction protocols [7], and negotiation meta-language [19].

2. Negotiation Techniques

MAS coordinate intelligent behavior among a collection of autonomous intelligent agents. Each one may have different goals and different interests, which may conflict with the interests of other agents in the system. These conflicts are antagonistic states or actions. Chang and Woo [2] have classified conflicts into four different types: conflicts of interest when agents compete for scarce resources, conflicts of value, cognitive conflicts when agents differ in their thought of processes or perceptions, and goal conflicts when the desired outcomes of agents differ.

CNP is the first work to use a negotiation process involving a mutual selection by both managers and contractors [16]. CNP, however, has a major shortcoming. That is, to start CNP, there must not be a conflict at all among agents. A contractor does not communicate its minimal condition, nor do bidders have a second choice. Moreover, a mutual agreement is eventually given by the decision of an offer. Hence, CNP is more like a standardized coordination method than a negotiation principle.

A large part of the argumentation is concerned with the development of formal models for reasoning with imperfect information. An argument is rather like an endorsement, though it is more accurate to think of it as a tentative proof the proof is tentative because argumentation allows the proof of a fact being true to co-exist with the proof of a fact being false, a state of affairs that is becoming acceptable amongst logicians. The argumentation systems [17, 18] is based on a well-ground framework for describing the reasoning process of negotiation agents. An originating agent puts forward an initial proposal. The recipient agents evaluate the proposal by constructing arguments for and against it. This system uses BDI modalities and argumentation consequence relations.

Zeng and Sycara [20] have considered negotiation in a marketing environment with a learning process in which the buyer and the seller update their beliefs about the opponent's reservation price, using the Bayesian rule. However, they have used a simple model, where there are only two parties in the negotiation and where the subject of the negotiation is the price of the item. Lomuscio *et al.* [10] have defined the negotiation space for electronic commerce. That is, they have identified the possible parameters that can be used to classify any negotiation mechanism for electronic commerce. Such a classification is an important step for the development of more sophisticated shopping assistants because it defines and delimits the design space for agent interactions.

On the other hand, Anthony *et al.* [1] have presented the design of autonomous agent that can participate across multiple online auctions, in particular, English auction and Dutch auction. For example, in English auction in Figure 1, the auctioneer seeks to find the

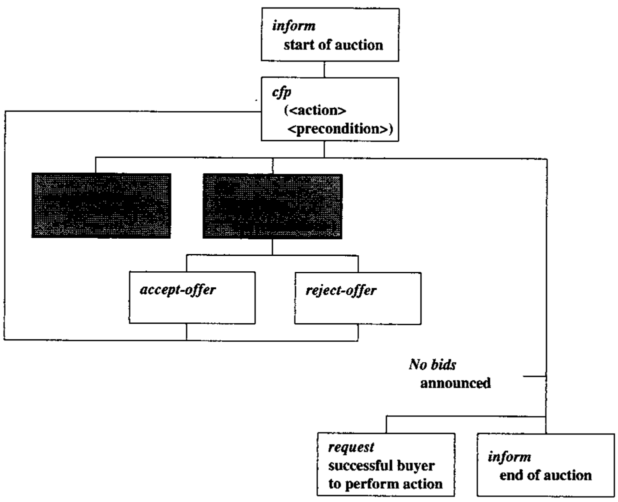

Figure 1. English auction protocol

market price of a good by initially proposing a price below that of the supposed market value, and then gradually raising the price. Each time the price is announced, the auctioneer waits to see if any buyers will signal their willingness to pay the proposed price. As soon as one buyer indicates that it will accept the price, the auctioneer issues a new call for bids with an incremented price. The auction continues until no buyers are prepared to pay the proposed price, at which point the auction ends.

3. A BDI Agent Architecture for Negotiation

Rao and Georgeff [14] have presented the possible-worlds formalism for BDI agent architectures, called the direct interpretation. However, this is not the interpretation we use in our work, we prefer to build a multi-agent system with an indirect interpretation in which B, D, and I are taken as meta-predicates. This means that predicates can be modeled in first-order logic and meta-predicates cannot be modeled in first-order logic.

3.1 Axioms for Resource-bounded BDI

In this paper, we consider that the axiomatization for beliefs is the standard $KD45$, D and K axioms for desires and intentions, and Modus Ponens inference rule. We also consider axioms for resource-bounded BDI. Firstly, when an agent X is the owner of resources Z and it gives Z to another agent Y, Y becomes its new owner:

$$\mathrm{B}_i(have(X, Z) \wedge give(X, Y, Z)) \rightarrow \mathrm{B}_i(have(Y, Z)). \quad (3\text{-}1)$$

When an agent j is not the owner of resources Z and another agent X gives it to j, j believes itself a new owner of Z:

$$\mathrm{B}_j(\neg have(j, Z) \wedge give(X, j, Z)) \rightarrow \mathrm{B}_j(have(j, Z)). \quad (3\text{-}2)$$

Secondly, when an agent X gives resources Z away, it no longer owns Z:

$$\mathrm{B}_i(have(X, Z) \wedge give(X, Y, Z)) \rightarrow \mathrm{B}_i(\neg have(X, Z)). \quad (3\text{-}3)$$

Thirdly, when an agent i has something Z that it does not believe to hold on and is requested to give it to another agent X, i adopts the intention of giving Z to X. Naturally more complex cooperative strategies can be defined if desired:

$$\mathrm{B}_i(have(i, Z)) \wedge \neg \mathrm{B}_i(holdon(i, Z)) \wedge request(X, i, give(i, X, Z)) \rightarrow give(i, X, Z). \quad (3\text{-}4)$$

When an agent i has resources Z that it does believe to hold on and is requested to give it to another agent X, i does not give Z to X:

$$\mathrm{B}_i(have(i, Z)) \wedge \mathrm{B}_i(holdon(i, Z)) \wedge request(X, i, give(i, X, Z)) \rightarrow \neg give(i, X, Z). \quad (3\text{-}5)$$

We also try to follow Rao and Georgeff axiomatization in which if an agent adopts a formula α as an intention, the agent should adopt α as a goal to be achieved, called the *goal-intention compatibility*, i.e., $intend(\alpha) \rightarrow goal(\alpha)$. And if an agent has an intention to do a particular action, the agent will do that action, called the *intentions leading to actions*, i.e., $intend(do(a)) \rightarrow do(a)$. In addition to this axiomatization, we try to confine intentions to commit when the agent believes that it can do the action i.e., $(intend(do(a)) \wedge can(a)) \wedge do(a)$.

We need to give both agents a domain dependent rule to link inter-agent communication and the agent's mental attitudes. When an agent i needs something Z from another agent X, it requests for Z:

$$\mathrm{I}_i(give(X, i, Z)) \rightarrow request(i, X, give(X, i, Z)). \quad (3\text{-}6)$$

3.2 A BDI Agent Architecture

In general, an agent has a variety of knowledge to achieve its goal, to plan some task, and to communicate with other agents. For example, agents are required to possess knowledge that can be represented as a set of sentences. These sentences describe knowledge about their beliefs or capabilities, other interactive agents, how to communicate with others, and a specific application domain. 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. The factors in the knowledge base are represented by predicates which mean mental attitudes of the agent. 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 2 and 3 illustrates our BDI agent architecture for negotiation and the planner structure, respectively.

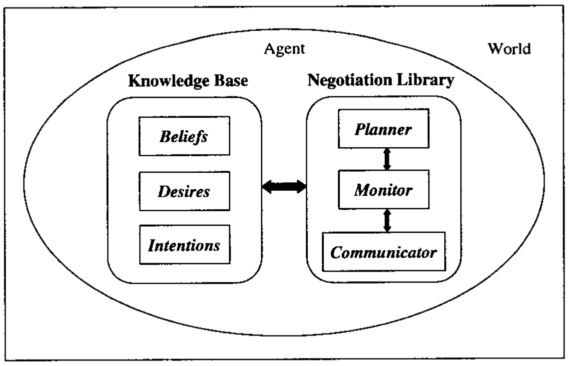

Figure 2. A BDI agent architecture for negotiation

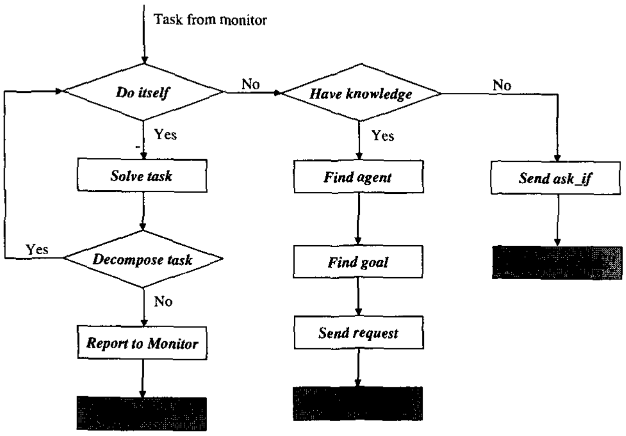

Figure 3. The planner structure

4. A Negotiation Model

We describe negotiation processes in MAS with resource-bounded BDI agents that do not have a complete state of knowledge about others and propose a negotiation algorithm to resolve goal conflicts for BDI agents that can undertake negotiation by changing their beliefs. We then address rational communicative behaviors in autonomous, self-interested, resource-bounded artificial BDI agents that have to what to communicate, to whom, and how.

4.1 An ACL for BDI Agents

We use a variant of the KQML, FIPA ACL specification, and the negotiation meta-language as our negotiation language for BDI agents to resolve goal conflicts. Though our communication language is similar to the KQML and FIPA ACL, we are not forced to make the assumption that the agents know and understand these languages. The actual exchange of messages is driven by the participating agents' own needs, goals, or mental attitudes. We represent the set of beliefs as *B*, the set of desires as *D*, and the set of intentions as *I*. Each agent has a unique identifier and we denote the set of identifiers of the agents involved in negotiation as *Agents*. Assumed BDI agents are negotiating about the allocation of deficient resources, agents require the allocation of deficient resources to achieve their goals. We denote a set of goals as *Goals* and a set of the resources as *Resources*. In this case, we can define a communication language *CL* for BDI agents as follows:

Definition 1. Given $a1, a2 \in Agents$, $g \in Goals$, $r \in Resources$, and $m \in B, D$, or I, we define a *CL*:
request(a1,a2,g,r) ∈ CL   *ask_if(a1,a2,m)* ∈ *CL*
inform(a1,a2,m) ∈ CL   *give(a1, a2, r)* ∈ *CL*
reject(a1,a2,g,r) ∈ CL   *achieved_goal(a1,a2)* ∈ CL
alternative(a1,a2,g,subgoals) ∈ *CL*

In the definition 1, *request(a1, a2, g, r)* means that agent *a1* requests deficient resources *r* from agent *a2* to achieve its goal *g* where *g* is the reason why *a1* needs *r*. *ask_if(a1, a2, m)* indicates that *a1* asks *a2* if *m* is true while *inform(a1, a2, m)* that *a1* informs *a2* that *m* is true.

4.2 A Negotiation Protocol for BDI Agents

Negotiation protocols could cover the permissible types of participants e.g., the negotiators and any relevant third parties, the negotiation states e.g., accepting requests or negotiation closed, the events which cause negotiation states to change e.g., no more requests or request accepted, and the valid actions of the participants in particular states e.g., which messages can be sent by whom, to whom, at what stage.

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 4 as a finite state diagram. In Figure 2, S0~S7 represent different negotiation states during a negotiation process. S0 is the initial state and S7 is the terminal state in which an agreement or disagreement is reached. The process of negotiation starts when an agent generates a *request* message. Other agents then either accept it, reject it, or make a counter-request.

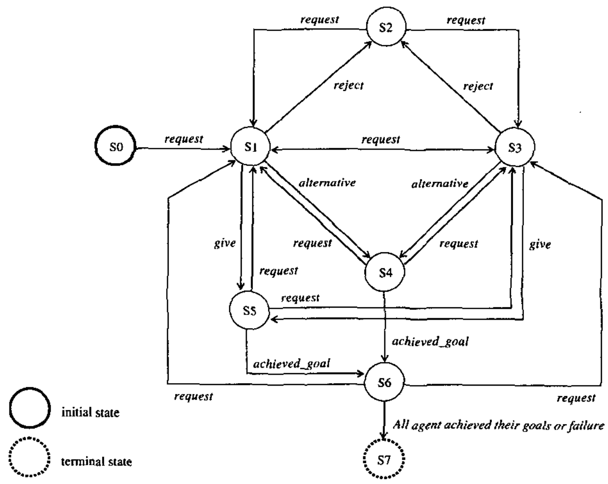

Figure 4. A negotiation protocol for BDI agents

4.3 Interpretation and Generation of Messages

The interpretation of messages implements a negotiation-state transition and the generation of messages determines a message to be taken in a particular state. However, the actual effect of a message depends upon the agent's interpretation. This interpretation process is highly domain-specific and is also dependent upon the internal structures presented in the agent architecture. For this reason, we illustrate how our framework can be used to define a comparatively simple open-minded agent. Naturally this does not prescribe how all agents should behave, but rather exemplifies the concepts of our model which can be used to define many other types of agent

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*, (4-1)

G(*request*(*a1*, *a2*, *g*, *r*)) = *alternative*(*a2*, *a1*, *g*, *subgoals*)

If *g* satisfies the unacceptable conditions and there does not exist an alternative, (4-2)

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*(*a*1, *a*2, *g*, *r*)) = *give*(*a*2, *a*1, *r*) (4-3)

If *a2* has received an alternative, *G*(*alternative*(*a*1, *a2*, *g*, *subgoals*)) = *make a replanning* (4-4)

If *a2* has received resources *r*, *G*(*give*(*a*1, *a2*, *r*)) = *continue planning* (4-5)

If a request was rejected by *a*1, *G*(*reject*(*a*1, *a*2, *g*, *r*)) = *search an alternative* (4-6)

If *a*1 has notified its goal achievement, *G*(*achieved_goal*(*a*1, *a*2)) = *make a replanning* (4-7)

If *a2* was asked whether *p* is true or not, (4-8)

G(*ask_if*(*a*1, *a2*, *p*)) = *inform*(*a2*, *a*1, *p*) or *inform* (*a2*, *a*1, not *p*)

If *a*1 informs *a2* of the truth of *p*, *G*(*inform*(*a*1, *a2*, *p*)) = *continue planning* (4-9)

Agents also have a simple theory of act that integrates a model of the available resources with their planning mechanism and forms another part of the theory contained in their beliefs. In the cases of (4-3), (4-4), (4-5), (4-7), and (4-9) of the definition 3, *a*1 and *a2* should modify their knowledge bases. For example, in the case of (4-5), if *a2* sends *give*(*a2*, *a*1, *nail*) to *a*1, then *a*1 and *a2* must modify their knowledge bases based on the following two ideas: *ownership* and *unicity*.

5. Experiments and Comparisons

5.1 Experiments

We consider three home improvement BDI agents with different objectives and resources. Agent *a1* has the intention of hanging a picture, *i*(*do*(*a*1, *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 *a*1 and agent *a3* has a hanger. Agent *a2* has the intention of hanging a mirror, *i*(*do*(*a2*, *hang_mirror*)), and believes that it has a mirror and a nail. It also believes that its name is *a2* and agent *a*1 has a screw, a hammer, and a screwdriver. Finally, agent *a3* has the intention of hanging a clock, *i*(*do*(*a3*, *hang_clock*)), and believes that it has a clock and a hanger. It also believes that its name is *a3* and agent *a*1 has a hanger nail.

We construct BDI agents and allow them to negotiate with each other from the top-level window in Figure 5. 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 5, Agent1Window class displays *Simple BDI agent*1 window with two panes: the top shows all outputs, *stdout* and *stderr*, from Prolog and the bottom is an editable text field which is sent to Prolog's input, *stdin*, after hitting the Enter key. That is, this window interacts with the back-end Prolog *stdio* streams. Agent1Window class then consults a routine of initializing *Agent*1 using sendAndFlush method

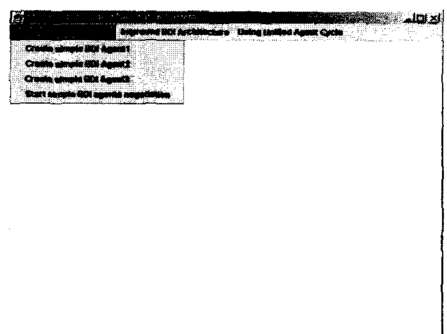

Figure 5. The top-level window for negotiation

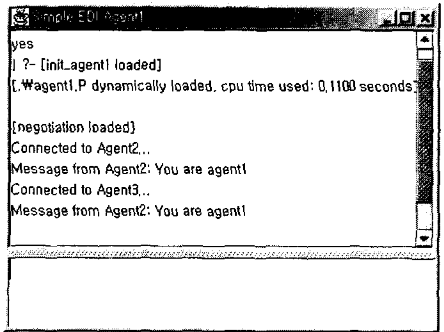

Figure 6. Construction of BDI *Agent*1

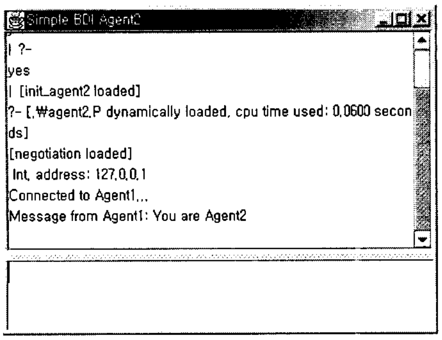

Figure 7. Construction of BDI *Agent2*

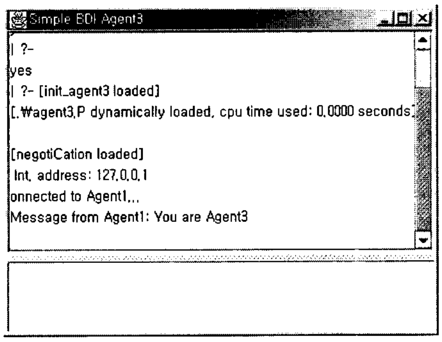

Figure 8. Construction of BDI *Agent3*

in PrologEngine class. Agent2Window and Agent3Window classes perform similar tasks to Agent1Window class. Figure 6, 7, and 8 illustrate *Agent1* window, *Agent2* window, and *Agent3* window, respectively.

PrologEngine class represents and gives access to a running Prolog process in background. Multiple instances correspond to multiple Prolog processes, outside the Java virtual machine. sendToFlush method in this class sends a string to Prolog's input. Let's look at the initialization routine of *Agent1* in the form of XSB Prolog. This routine consults negotiation mechanism routines for BDI agents and the knowledge base of *Agent1*. It also prepares *Agent1* for communication with *Agent2* and *Agent3*. There are many mechanisms of communication such as stream-oriented and message-oriented, but we use buffered, message-based communication mechanism using sockets: communication processes exchange messages that have well-defined boundaries, and use *socket_send/3* and *socket/3* to talk to each other where *p/n* represents predicate *p* has an arity of *n*.

On the other hand, the initialization routine of *Agent2* consults negotiation mechanism routines for BDI agents and the knowledge base of *Agent2*. After it prepares *Agent2* for communication with *Agent1*, *Agent2* sends a message to *Agent1*, receives a confirmation message, and waits for a message from *Agent1*. This means that we assume that *Agent1* gives an initial request, but if *Agent2* does, we may have the same result.

When we click *Start simple BDI agents negotiation* item in Figure 5, negotiation among *Agent1*, *Agent2*, and *Agent3* starts. Agents continue to negotiate with one another by sending and receiving messages and finally reach a mutual agreement state through negotiation and achieve their conflicting goals. Figure 9, 10, and 11 show these negotiation processes.

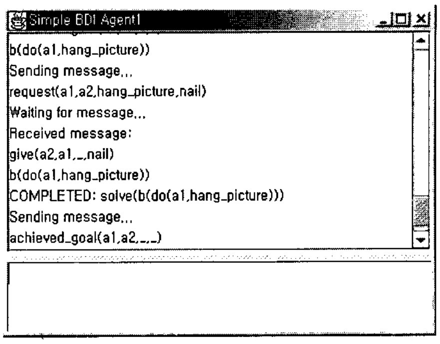

Figure 9. Negotiation process of *Agent1*

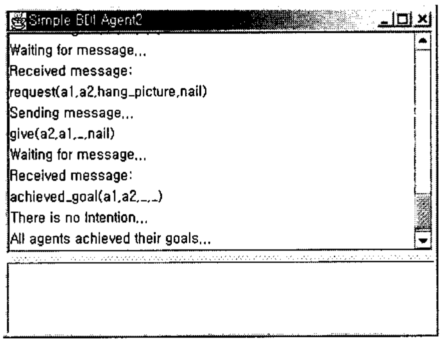

Figure 10. Negotiation process of *Agent2*

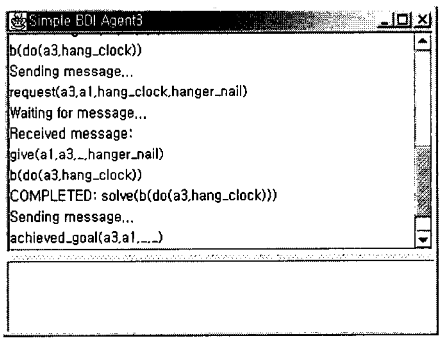

Figure 11. Negotiation process of *Agent3*

5.2 Comparisions

It is known that agents cooperate or compete with one another to solve specific problems in MAS. In the paper, we suppose BDI agents that intend to accomplish conflicting goals and we investigate the possibility of negotiation among them. Even though this study is carried out under the suggestion that agents are to negotiate essentially and they are in an amicable atmosphere, we show that negotiation among agents occurs smoothly on the situation that agents use the unacceptable conditions mentioned here and do not have complete knowledge about others.

The model by Rao and Georgeff [14] is one of models that represent agents' beliefs, desires, and intentions using modal logic. Specifically, since this model represents sets of possible worlds that agents could be included as their beliefs, it is powerful but not realistic. On the other hand, Parsons and Giorgini [13] establish beliefs, desires, and intentions as different units, respectively in order to model BDI agents and they represent interrelationships among beliefs, desires, and intentions using the bridge rules. In their modeling, if the finish of a work is added to communication unit *C*, i.e., *C*: *done*(e), this fact is also added to belief unit *B*, i.e., *B*: *B*(*done*(*e*)). If an agent has an intention, which is added to intention unit *I*, i.e., *I*: *I*(*does*(*e*)), it is delivered to the communication unit, *C*: *does*(*e*).

In this paper, we build a multi-agent system with an indirect interpretation in which *B*, *D*, and *I* are taken as meta-predicates and consider that the axiomatization for beliefs is the standard *KD*45, *D* and *K* axioms for desires and intentions, and some axioms for resource-bounded BDI. In our system, when each agent interacts one another and notices inconsistency of its beliefs, we try to confine beliefs as much as possible, making each agent modify its beliefs. In Rao and Georgeff axiomatization, if an agent has an intention to do a particular action, the agent does the action. We also try to confine intentions to commit when the agent believes that it can do the action. To make work execution more delicate under logic programming environments, we separate rules from knowledge base to solve problems and we organize BDI agents as four factors: knowledge base, planner, monitor, and communicator.

Comparisons with some of negotiation techniques for resolving conflicts are as follows. CNP does not allow agents to cause goal conflicts and a manager agent is responsible for all negotiations. Unlike 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.

The argumentation system [12, 18] is an advanced negotiation system, but it 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. Therefore, the argumentation system consumes much time for the evaluation of the argument. The loading of communication is high because the argumentation system transports messages added to arguments that the agent makes for itself. In this paper, if agent *a2* receives a message, *request*(*a*1, *a*2, *g*, *r*), *a2* 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. We show the capability of negotiation among agents without using arguments so that the loading of communication is also reduced comparing with the argumentation system.

We suppose the environments in which agents negotiate about accomplishment of conflicting goals and allocation of the deficient resources. Kasbah [3], an electronic commerce system implemented by LISP, creates a buying agent and a selling agent substituting for users. These agents negotiate with one another on Kasbah market. Kasbah agents 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.

6. Conclusions and Future Research

We have represented agents' beliefs, desires, and intentions for MAS in logic programming environments and introduced a negotiation mechanism for BDI agents and considered a MAS in which beliefs, desires, and intentions are taken as meta-predicates and defined some axioms for resource-bounded BDI agents. These axioms help agent to act rationally, to reason about the environments, and to plan its goal achievement. We have then established a BDI agent architecture with four components: knowledge base, planner, monitor, and communicator and shown how goal conflicts through negotiation in MAS are resolved using our ACL and negotiation algorithm in practice.

Negotiation strategies of agents vary according to the environments that they belong to. This paper has raised the necessity of many types of specialized negotiation strategies, for example, negotiation strategies for agents requiring allocation of the deficient resources, for agents in retail commerce transaction, and for auction agents should have complete differences, respectively. We could carry out a type of hypothetical inference about abilities and actions of other agents, adding a function of abductive inference to the structure of BDI agents. In other words, inferring what intention, goal, or knowledge an agent has from an action of the agent hypothetically, we will use the inferred results as important evidences of negotiation in the state of cooperation, competition, or goal disagreement among agents. Finally, we have assumed that different agents share the same communication language. However, this assumption is not essential. Indeed, translator agents could be defined, acting as mediators between agents with different communication languages. This may be possible by virtue of the metalogic features of the language.

References

[1] P. Anthony, W. Hall, V. Dang, and N. R. Jennings. "Autonomous Agents for Participating in Multiple On-line Auctions". In *Proceedings of the International Joint Conferences on Artificial Intelligence Workshop on E-Business and the Intelligent Web*, pages 54-64, 2001.

[2] M. K. Chang and C. C. Woo. "A Speech-act-based Negotiation Protocol: Design, Implementation, and Test Use". *The Association for Computing Machinery Transaction in Information Systems*, 12(4):360-382, 1994.

[3] 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.

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

[5] E. H. Durfee, V. R. Lesser, and D. D. Corkill. "Trends in Cooperative Distributed Problem Solving". *The Institute of Electrical and Electronics Engineers Transaction Knowledge Data Engineering*, 1(1):63-83, 1989.

[6] FIPA Agent Communication Language Message Structure Specification. http://www.fipa.org/specs/fipa00061/. *The Foundation for Intelligent Physical Agents* (FIPA), 2000.

[7] FIPA Interaction Protocol Library Specification. http://www.fipa.org/ specs/fipa00025/. *The Foundation for Intelligent Physical Agents* (FIPA), 2000.

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

[9] InterProlog: A Java front-end and enhancement for Prolog. http://www.declarativa.com/interprolog/. *Declarativa*, Portugal, 2002.

[10] A. Lomuscio, M. Wooldridge, and N. R. Jennings. "A Classification Scheme for Negotiation in Electronic Commerce". *Agent-Mediated Electronic Commerce: A European AgentLink Perspective*, pages 19-33, 2001.

[11] H. J. Muller. Negotiation Principles. *Foundations of Distributed Artificial Intelligence*, pages 211-229, 1996.

[12] 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.

[13] S. Parsons and P. Giorgini. On Using Degrees of Belief in BDI Agents. *Technical Report*, Department of Electronic Engineering, Queen Mary and Westfield, 1998.

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

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

[16] T. Sandholm. "An Implementation of the Contract Net Protocol Based on Marginal Cost Calculations". In *Proceedings of the twelfth International Workshop on Distributed Artificial Intelligence*, pages 295-308, 1993.

[17] C. Sierra, P. Faratin, and N. R. Jennings. "A Service-oriented Negotiation Model Between Autonomous Agents". In *Proceedings of the Eighth European Workshop on Modeling Autonomous Agents in a Multi-Agent World*, pages 17-35, 1997.

[18] C. Sierra, N. R. Jennings, P. Noriega, and S. Parsons. "A Framework for Argumentation-Based Negotiation". In *Proceedings of the Fourth International Workshop on Agent Theories, Architectures and Languages*, pages 177-192, 1997.

[19] M. Wooldridge and S. Parsons. "Languages for Negotiation". In *Proceedings of the Fourteenth European Conference on Artificial Intelligence*. 2000.

[20] D. Zeng and K. Sycara. "Benefits of Learning in Negotiation". In *Proceedings of the American Association on Artificial Intelligence*, pages 36-41, 1997.

저 자 소 개

이명진(Myung-Jin Lee)

1994년: 계명대학교 대학원 컴퓨터공학과 (공학석사)

2002년: 계명대학교 대학원 컴퓨터공학과 (공학박사)

2003년~현재: 아시아대학교 인터넷비즈니스학과 전임강사

관심분야: 에이전트 시스템, 전자상거래

E-mail: mjlee@asiau.ac.kr

김진상(Jin-Sang Kim)

제10권 3호 참조