FACILITATING NEGOTIATIONS IN AGENT MEDIATED ELECTRONIC COMMERCE

Chunyan Miao[a], Angela Goh[a], and Zhonghua Yang[b]

[a] School of Computer Engineering, Nanyang Technological University
Nanyang Avenue, Singapore 639798
E-mail: cymiao@cs.sfu.ca, asesgoh@ntu.edu.sg

[b] School of Electrical and Electronic Engineering, Nanyang Technological University
Block S2, Nanyang Avenue, Singapore 639798
E-mail: ezhyang@ntu.edu.sg

Abstract

There is no doubt that agents play an increasingly predominant role in e-commerce, whether these are business-to-consumer or business-to-business applications. However most of the current e-commerce agents only support a single bid for a product at a fixed price. Although price is an important factor, it is not the only concern of both business and consumer. There is doubt as to whether such agents satisfy both parties. Negotiation on a variety of issues is needed in order to reach an agreement. In this paper, a computational agent negotiation (CAN) model is proposed to facilitate multiple-issue negotiation via an agent. The main contribution of the CAN model is it enables agents to participate actively in the negotiation with various feedback instead of simply an agreement or rejection.

Keywords: *Agent; Agent negotiation; Electronic Commerce; Fuzzy Cognitive Map (FCM).*

1 Introduction

In the last few years, we have witnessed a surge of e-commerce systems operating on the Internet. By 2003, the Internet will become a predominant mechanism for conducting business, whether it be business-to-consumer or business-to-business [1]. However, a large number of current e-commerce systems operating on the Internet are nothing more than electronic catalogues with fixed prices that allow users to browse, choose and make purchases by means of a electronic transaction [2]. In recent years, a small number of advanced e-commerce systems called agent mediated e-commerce systems have emerged.

Agent technology has become a new paradigm for developing software applications in the 1990's [3]. There has been much discussion about the definition of an agent. This confusion about what an agent is had led to the publication of the well-known paper *"Is it an agent or just a program"* [4].

An agent can be viewed as an encapsulated software program that acts *autonomously* to achieve its goal. Although there is still no universal definition of an agent, it is well accepted that agents possess characteristics that clearly distinguish them from other software programs [5]. Unlike traditional software programs, agents are pro-active, goal oriented and have clear objectives to achieve. Agents have control over both of their internal state and their behavior, which make them act autonomously in response to environmental changes. Agents have been applied to many application domains, including electronic commerce, to solve real-world problems [5]. Moreover, agents have been advocated as the next generation model for engineering complex software systems in the new millennium [6].

Agents can be further empowered by forming multi-agent systems (MAS). Due to their autonomous characteristics, agents are independent and decentralized. Therefore agents in a MAS need to interact with each other. Examples of these interactions include messaging, information exchange, request for a particular action and so on. One of the most advanced and important agent interactions is agent negotiation [7]. Agent negotiation enables the agents to negotiate or even argue with each other in order to reach an agreement. For instance, a buyer agent and a seller agent need to negotiate with each other in order to reach an agreement of a business transaction in electronic commerce.

The term "agent mediated electronic commerce" was first defined and proposed by the MIT agent group [8]. They identified six stages in electronic commerce that agents can be applied to, including product information identification, product brokering, merchant brokering, negotiation, purchase, and product service. It is observed that agents such as shopping assistants and merchant brokers are now available in some advanced e-commerce systems. There is no doubt that agents play an increasingly predominant role in e-commerce, whether these are business-to-consumer or business-to-business applications. However most of the current e-commerce agents only support a single bid for a product at a fixed price. Although price is an important factor, it is not the only concern of both business and consumer. There is doubt as to whether such agents satisfy both parties. Negotiation on a variety of issues is needed in order to reach an agreement. In this paper, a computational agent negotiation (CAN) model is proposed to facilitate multiple-issue negotiation via an agent. The main contribution of the CAN model is it enables agents to participate actively in the negotiation with various feedback instead of simply issuing an agreement or rejection.

Following this introduction, the next section describes related work. Section 3 presents the proposed Computational Agent Negotiation (CAN) model. Finally, the conclusion is reached in Section 4.

2 Related Work

To overcome the limitations of current agent-mediated electronic commerce systems, the MIT Agent group proposed a dynamic attribute-based negotiation approach through which offers could be selected based on multiple criteria and not on price alone[9]. They used an air ticket bidding system called SARDINE [10] (System for Airline Reservations Demonstrating the Integration of Negotiation and Evaluation) to demonstrate their approach.

The basic idea and process of the dynamic attribute-based negotiation is described as follows:

- to collect the buyer preferences: the buyer agent asks the buyer to indicate his/her preferred parameters of the target product and how flexible he/she is on each parameter. The flexibility rating of "very flexible", "somewhat flexible" and "not flexible" is used by the buyer agent to determine an acceptable range of each parameter. Additionally, this flexibility metric is used as an indicator of how important each parameter is to the buyer. For instance when a buyer is very flexible on a certain flight attribute such as 'time-of-the-day', this indicates that he/she is willing to consider any time of the day for the flight. Hence, 'time-of-the-day' is not considered an important attribute to this person.
- Based on the preference and flexibility rating, the buyer agents locate the offers and calculate the distance from ideal value to available value of all parameters.
- The buyer agent locates the best one for the buyer by minimizing the above-mentioned distance and submits the bids to the appropriate Seller agent.
- The seller agent will analyze the flight bids and determine if they should be accepted or rejected.
- Once all bids have been returned, the buyer makes the final decision as to which is the best choice.

The advantage of the dynamic attribute-based negotiation is it allows the buyer to express to a buyer agent preferences for multiple factors relating to a target product instead of the single price issue. However, with current approaches, after receiving a bid from a buyer agent, the seller agent can only response with either an "accept" or a "reject". The seller agent does not have the ability to make an counter-offer, which means the seller agent has to "reject" the bid even if it can partially satisfy the buyer's requirement.

Chhaya Mudgal al etc. [11] propose an influence diagram model for multiple issue agent negotiation. Based on the model, the agent can be in an Offer or Counter-offer state repeatedly during the negotiation, which means the response from the seller agent could be "accept", "reject", or "counter-offer". The buyer agent takes into account multiple preferences of the buyer, which includes the preferred price, urgency of demand, the importance attached to price, and the user's risk behavior. These preferences are represented by nodes that may affect the decision in the influence diagram. In addition to the above preferences, a conditional node which represents a Seller or Competitor's action will also influence the agent's decision making. The main contribution of the influence diagram model is that it contains a chance node for modeling the Competitor's action instead of only considering the buyer's preferences. Nevertheless, during the negotiation, both parties may be faced with the need to consider a variety of issues. Hence, many other parameters also play important roles in the negotiation. Unfortunately, the preferences modeled by the inference diagram are fixed and very limited.

3 Computational Agent Negotiation (CAN) Model

Some researchers have suggested that a new generation of electronic commerce systems based on automatic agent negotiation[12][13] will emerge in the next few years. Automatic negotiation can significantly reduce negotiation time, and can also remove some of the reticence of humans to engage in negotiation. This reticence may arise from embarrassment, personality, etc. Therefore, the research on mechanisms for agent negotiation has received a great deal of attention in the multi-agent systems community[12].

Motivated by this background, we propose a computational agent negotiation model to enable the automatic negotiation of multiple issues and to facilitate both parties to play an active role during the negotiation.

3.1 Modeling the Life Cycle of Agent Negotiation

[Definition 1] *Agent negotiation* is a process in which agents initiate a proposal and negotiate with each other (e.g. make offers or counter-offers), in order to reach an agreement

The life cycle of agent negotiation includes the following stages:

1) Initiate a negotiation
2) Propose the negotiation
3) Get offers or counter-offers from counterpart agents that may include Sellers, Competitors and so on.
4) Negotiate
5) Close negotiation (reach agreement or time out)

[Definition 2] The *Computational Agent Negotiation (CAN) model* is represented by a directed graph. It consists of a set of *negotiation items* (represented by nodes) and a set of *negotiation weights* (represented by directed edges). The negotiation items describe the factors an agent takes into account during the negotiation, and the negotiation weights specify the relationships between negotiation items and indicate how important the relationships are.

The CAN model is based on the Fuzzy Cognitive Map (FCM) theory [14]. FCMs are signed and directed graphs with feedback that model the world as a collection of concepts and casual relationships between concepts.

This section illustrates how the CAN model can be built and used in the life cycle of agent negotiation. An example scenario consists of a buyer agent who helps a buyer to purchase an air ticket from airline agents. To initiate a negotiation, the buyer agent will collect the factors that the user will take into account when purchasing an air ticket. These factors are defined as negotiation items, and represented by nodes in the CAN model. A variable number of such factors may be stipulated. After all the negotiation items have been defined, the agent will specify the relationships between the items, and assign the weights according to the buyer's preferences.

In this scenario, the buyer agent takes into account the following concerns from the buyer: *ticket price*, *number of connection*, *airline*, and *level of satisfaction*. It is assumed that the airlines are ranked by their track record which includes safety, service and so on. There is a direct correlation between the rank and the price of the ticket. In other words, the higher the rank is, the more expensive the air ticket will be.

Based on the factors shown in Figure 1, the buyer agent defines the set of negotiation items as: {ticket price, number of connection, airline rank, level of satisfaction}. The item *level of satisfaction* is the item upon which the final decision is made. In this paper, it is termed "decisional item".

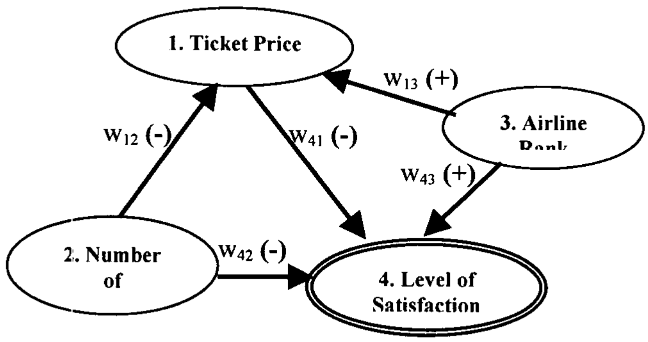

Figure 1. Computational Agent Negotiation Model

The CAN model of the buyer agent can be built as shown in Figure 1. $Item_i \rightarrow Item_j$ with *weight* w_{ji} indicates that $Item_i$ influences $Item_j$ by *weight* w_{ji} . $Item_i$ is called as *causal item*, and $Item_j$ is called as *effect item*. The *weight* w_{ji} indicates how strong the influence is. The plus (+), and minus (-) sign followed by *weight* w_{ij} in the graph indicates that the influence is positive or negative. For instance, w_{41} indicates that the ticket price is inversely proportional to level of satisfaction. This is shown by the w_{41} (-) symbol. In other words, the higher the *Ticket Price* is, the less the buyer is satisfied.

The double-circled node (e.g. the *Level of Satisfaction* in the example) is named the *decisional node*. The agent will make decisions based on the value of this node using a decision making function.

When the negotiation is initiated, both the items and the weights will be assigned a value that is termed item state value and weight value respectively.

[Definition 3] The *Value Set of the Computational Agent Negotiation (CAN) model*, V_{CAN} is defined as a 3-element tuple, $V_{CAN} = \{I, E, W\}$:

- $I = \{r_i \mid r_i ;\ i=1, 2, \ldots n\}$ represents state value set of **negotiation items**;
- $E = \{x_i \mid x_i \in [-1, 1];\ i=1, 2, \ldots n\ \&\ x \text{ is a real number}\}$ represents evaluation value set of **negotiation items**;
- $W = \{w_{ij} \mid w_{ij} \in [-1, 1];\ i=1, 2, \ldots n; j=1,2, \ldots n\ \&\ w_{ij} \text{ is a real number}\}$ represents value set of **weights**;

The item state value is initialized to the expected ideal value. The item evaluation value is intitialized to zero. It represents a value that compares the expected ideal value of the item and the offered item value received. The item evaluation value will be mapped onto a member value of the fuzzy set in the range [-1, 1], i.e., a fuzzy distance value, upon receiving offers from counterparts. The value of the weight is initialized to a fuzzy value that reflects how strongly one item influence another.

To specify how much one item influences another, people often use fuzzy expressions such as {*highest, very high, high, a bit high, average, a bit low, low, very low, lowest*}. All these fuzzy expressions can be mapped onto a member value of the fuzzy set in the range [-1, 1], through fuzzification methods. Such mapping is called fuzzification [15]. For instance, the fuzzy expression set {*highest, very high, high, a bit high, average, a bit low, low, very low, lowest*} can be mapped to {*0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1*} after fuzzification. Therefore the weight value in the CAN model is in the range [-1, 1].

The initiation of an agent negotiation is considered completed after the CAN model is built and the item values and weight values are assigned. Based on the CAN model, the buyer agent will propose the negotiation with the negotiation items and their values. In the CAN model, the decisional node is considered a *private node*, which will not be accessible by counterparts agents. The rest are considered as *public nodes*. Therefore, when preparing the negotiation proposal, only public items and its value set will be included.

[Definition 4] A *proposal of negotiation* consists of a set of negotiation items and a set of expected values corresponding to the negotiation items.

Once the negotiation is proposed, the agent will obtain responses from the seller agents. In our example, the seller agent (or counterpart) of the buyer agents are airline agents. Similarly, the airline agents also have set up their CAN model based on their own considerations. They will evaluate the proposals and respond with an "accept", "reject", or propose a counter-offer.

[Definition 5] A *counter-offer* consists of a set of negotiation items and a set of new values corresponding to the expected value set of negotiation items given by the negotiation proposal.

For instance, although the airline agent could not provide *airline A* requested in the negotiation proposal, the airline agent had no difficulty in satisfying all the other requirements. Therefore, the airline agent would like to make a counter-offer instead of simply rejecting the proposal. For example, it may offer *airline B* which has a relatively competitive record with *airline A*, to replace *airline A*.

On the other hand, the buyer agents will review all the offers made by the airline agents. If there is an ideal offer that matches the buyer's preference, the buyer agent will accept this ideal offer, and reject the counter-offers. Otherwise, the buyer agent will evaluate all the counter-offers based on the CAN model, and decide which is the best choice on behalf of the buyer. In the worst case, the negotiation may be timed out without any agreement being reached.

3.2 Evaluation and Decision Making

This section describe how an agent evaluates the offers and determines the best offer based on the CAN model.

Assume that a counter-offer has been proposed in response to an initial proposal:

$I_{Counter\text{-}offer}$ represents the new value set of negotiation items given by the counter-offer and $I_{Proposal}$ represents the expected value set of negotiation items given by the negotiation proposal. The evaluation value set of negotiation items $E_{Offer\text{-}Evaluation}$ can be obtained by comparing $I_{Counter\text{-}offer}$ and $I_{Proposal}$, and mapping the distance between them into a member value of the fuzzy set in the range [-1, 1], through the fuzzification methods.

If there are *n* items nodes in the CAN model, a *n***n* Weight Matrix (W) can be built:

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \cdots w_{1n} \\ w_{21} & w_{22} & w_{23} \cdots w_{2n} \\ w_{31} & w_{32} & w_{33} \cdots w_{3n} \\ \cdots & \cdots & \cdots \\ w_{n1} & w_{n2} & w_{n3} \cdots w_{nn} \end{bmatrix}$$

A 1*n Item Item Evaluation Matrix (C) can be also built:

$$[x_1, x_2, x_3, \ldots x_n]$$

By multiplying Matrix (C) and Matrix (W), a new 1*n Matrix (C*W) is obtained:

$$C*W = [x_1, x_2, x_3, \ldots x_n] \cdot \begin{bmatrix} w_{11} & w_{12} & w_{13} \cdots w_{1n} \\ w_{21} & w_{22} & w_{23} \cdots w_{2n} \\ w_{31} & w_{32} & w_{33} \cdots w_{3n} \\ \cdots & \cdots & \cdots \\ w_{n1} & w_{n2} & w_{n3} \cdots w_{nn} \end{bmatrix} =$$

$$[y_1, y_2, y_3, \ldots y_n]$$

Note that

$$y_i = \sum_j w_{ij} x_j \tag{1}$$

where y_i is the sum of the products of the item evaluation value of all the *causal items* that influence $Item_i$ and the weight values between the two item nodes.

Assuming the *Item i* represents the decisional item, the agent will then make decisions based on this value. In the above example, y_i represents the value of *Level of*

Satisfaction. Taking y_i as the input of the decision making function of the *Item i*, the state value of the *Item i* can be further computed as:

$$x_i = f_i(y_i) = f_i(\sum_j w_{ij} x_j) \quad (2)$$

where x_i is the new state value of the *item i*, w_{ij} is taken from the weight matrix W and f_i is the decision making function of *Item i*.

The *causal item* value may affect the weight of the influence on the *effect item* in the real world problem. In the CAN model, the weight value is not limited to a fixed value in a fuzzy set. There may be a relationship function with the evaluation value of its *causal items*. This is termed *dynamic weight* and is illustrated in Figure 2. Dynamic weight allows its value to be calculated automatically by the state value of its *causal item*.

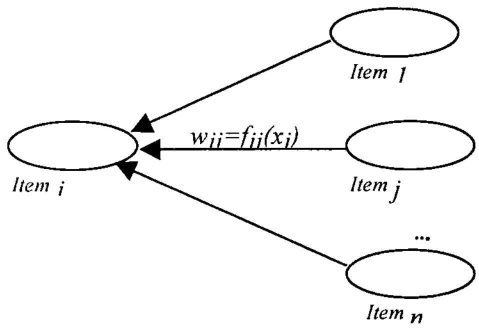

Figure 2. Illustration of Dynamic Weight

Assume that the weight w_{ij} represents the causal relationship from $Item_j$ to $Item_i$. The state value of $Item_i$ is determined by the effects from other items. This can be shown in the following equations:

$$x_i = f_i(w_{i1}, w_{i2}, \dots, w_{in}),$$

$$w_{ij} = f_{ij}(x_j), \quad i,j = 1,2, \dots, n$$

where x_i is the state value of the *item i*, w_{ij} is the weight value from *Item j* to *Item i*. f_{ij} is the relationship function between the weight w_{ij} and its causal item *Item j*.

By using the above formulae, the value of the decisional node (i.e. the value of item *Level of Satisfaction* in the example) can be obtained. The best choice will be determined by evaluating all the received counter-offers.

In the example shown in Figure 1, the higher the *Ticket Price* is, the lower the *Level of Satisfaction* will be; the better the *Airline Rank* is, the higher the *Level of Satisfaction* will be; and an increased *No. of Connections* will lead to a lower *Level of Satisfaction*. For example, the weight values may be defined as follows:

$$w_{41} = -0.5; \; w_{42} = -0.2; \; w_{43} = 0.3$$

where w_{41} represents the influence from *ticket price* to *level of satisfaction*, w_{42} represents the influence from *no. of connections* to *level of satisfaction*, and w_{43} represents the influence from *airline rank* to *level satisfaction*.

As described, the range of the item evaluation value is [-1, 1]. An item may be computed and mapped to [-1, 1] according to the difference of ideal value and the offer value received. When the buyer agent receives an offer which is exactly the same as what the user expects, then the item evaluation value equals zero:

$$y_4 = (-0.5*0) + (-0.2*0) + (0.3*0) = 0$$

where y_4 is the sum of the products of the item evaluation value and the weights of all the causal items that influence *level of satisfaction*. So the *level of satisfaction* remains unchanged. On the other hand, a buyer agent may receive an offer where the offered *ticket price* is higher than expected, while the rest of the item values are the same as expected. Assuming that due to the difference, the item evaluation value of *ticket price* is mapped to 0.2, then:

$$y_4 = (-0.5*0.2) + (-0.2*0) + (0.3*0) = -0.1$$

This indicates that the *level of satisfaction* is decreased. The decision function will determine whether this *level of satisfaction* is acceptable to the buyer.

This section illustrates that the CAN model provides a practical and simple way to equip agents with the ability to negotiate and make decisions on behalf of human beings.

4 Implementation

A proof-of-concept prototype is now being implemented using Java to enable the negotaition via agents based on the CAN model. Figure 3 shows the class diagrams of the prototype system.

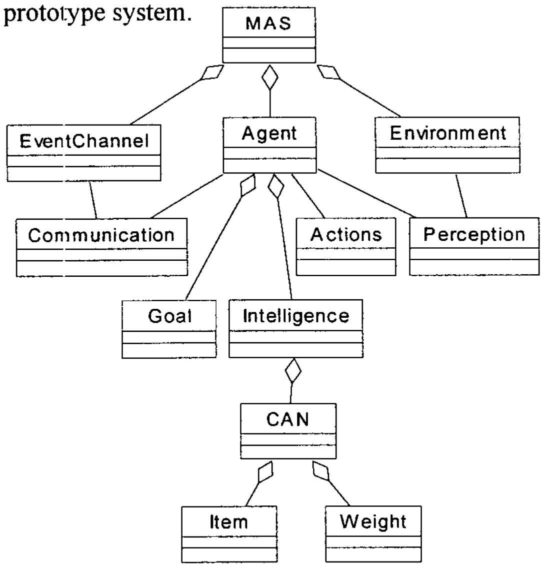

Figure 3. The Agent Class Diagrams

The multi-agent prototype system consists of agents, its environment, and an event channel. An agent possesses a goal as well as intelligence to act autonomously towards its goal. Therefore it can infer and determine the action to be taken based on its goal and knowledge. Agents have the ability to perceive within its environment and to negotiate (communicate) based on the knowledge represented by the CAN model.

The agent negotiation is carried out via an event channel. The event channel provides event services that enable the agents to negotiate with each other. Event channel receives negotiation proposals from supplier agents and delivers the negotiation proposals to consumer agents. In the prototype system, the event channel is implemented by CORBA event services using Visibroker for Java 3.2. Figure 4 shows the relationships between consumer agents, supplier agents and the event channel.

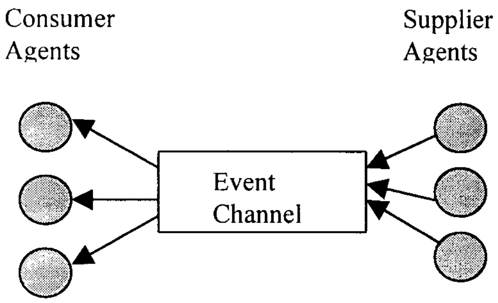

Figure 4. The Agent Negotiation Event Channel

The supplier agent is an event generator, who delivers an event (e.g. a negotiation proposal) to the event channel. The consumer agent is an event receiver who receives the event (e.g. a negotiation proposal) from the event channel. In our example, both the buyer agent and the seller agent can act as the consumer agent or the supplier agent. When the buyer agent proposes a negotiation proposal, it is regarded as a supplier agent. When the buyer agent received a counter offer made by the seller agent from the event channel, it becomes an consumer agent. On the other hand, when the seller agent receives a proposal made by the buyer agent from the event channel, it is regarded as a consumer agent. Similarly, when the seller agent makes a counter offer, it becomes a supplier agent.

After the supplier agents and consumer agents subscribe to the event channel, they can look for negotiation partners and negotiate with each other based on the CAN model via different push-pull modes.

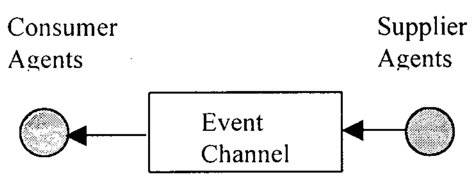

Figure 5. The Push –Push Communication

As shown in Figure 5, the push supplier agent pushes the events to the event channel whenever it initiates an negotiation proposal. In turn, event channel pushes the events to all consumer agents when and only when the events are pushed by supplier agents, which forms a push-push communication of agents.
Instead of receiving the negotiation proposals from the event channel passively, a consumer agents also can pull the events from the event channel actively. Meanwhile, a supplier agent may also allow the event channel to pull events continuously from it to get the available negotiation proposals, which forms a pull-pull communication of agents. Besides push-push, and pull-pull communications, agents can use push-pull, or pull-push communication depending on different situations. As it has been shown, both the buyer agent and the seller agent are able to look for negotiation partners and negotiate with their partner agents via the event channel through different combinations of pull/push communications. Moreover, the negotiation partner is able to propose counter offers and negotiate with the proposal initiator through the communication mode it prefers.

5 Conclusion

In this paper, we have proposed a new approach for automatic agent negotiation using the Computational Agent Negotiation (CAN) model. Compared with existing approaches of agent negotiation, the CAN model has the following advantages: 1) It supports automatic agent negotiations with multiple issues; 2) It models the whole life cycle of agent negotiation; 3) It enables both parties to play an active role during the negotiation; 4) It facilitates various feedback by counter-offers instead of a simple acceptance or rejection; 5) It provides flexibility for agents to adjust negotiation weights dynamically, and to represent fuzzy thinking during the negotiation. A further case study will be done in the future work to prove the practicability of the CAN model in life-like agent mediated electronic commerce applications.

References

[1] Wilderman J. 2000. The Es Have It: E-Business, E-Commerce, E-tailing and the Web, GartnerGroup Research Report.

[2] Lomuscio A.R., Wooldridge M. and Jennings N. R. eds. 2001. A Classification Scheme for Negotiation in Electronic Commerce, *Agent-Mediated Electronic Commerce: A European AgentLink Perspective* , Springer Verlag.

[3] Genesesretch M. R., Ketchpe S. P. 1994. Software Agents, *Communications of ACM*, Vol. 37:48-53.

[4] Franklin S., Graesser A. 1996. Is it an Agent, or just a

Program?: A Taxonomy for Autonomous Agents. In Proceedings of the Third International Workshop on Agent Theories, Architectures, and Languages.

[5] P Singh., Huhns M. N. 1998. *Readings in Agents*, Morgan Kaufmann.

[6] Jennings N.R., Sycana K., Wooldridge M. 1998. A Roadmap of Agent Research and Development, *Autonomous Agents and Multi-Agent Systems,* Vol. 1, Issue 1:7-38.

[7] Kraus S. 1997. Negotiation and Cooperation in Multi-Agent Environment, *Artificial Intelligence*, Vol. 94:79-97.

[8] Guttman R., Moukas A., and Maes P. 1998. Agent-mediated Electronic Commerce: A Survey, *Knowledge Engineering Review,* Vol. 13.

[9] Morris J., Maes P. 2000. Negotiating Beyond the Bid Price. In Proceedings of the CHI 2000 Workshop: Designing Interactive Systems for 1-to-1 E-commerce, The Hague (The Netherlands).

[10] Morris J., Maes P. 2000. Sardine: An Agent-facilitated Airline Ticket Bidding System. In Proceedings of the Fourth International Conference on Autonomous Agents (Agents 2000), Barcelona, Catalonia, Spain.

[11] Mudgal C., Vassileva J. 2000. An influence diagram model for multi-agent negotiation. In Proceedings of the International Conference on Multi-Agent Systems, Boston.

[12] Jennings N. R., Faratin P., Lomuscio A. R., Parsons S., Sierra C. and Wooldridge M. 2001 Automated negotiation: prospects, methods and challenges, *Int. J. of Group Decision and Negotiation* , Forthcoming.

[13] Guttman R., Maes P. 1998. Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce. In Proceedings of the Second International Workshop on Cooperative Information Agents, Paris, France.

[14] Kosko B. 1986. Fuzzy Cognitive Maps, *International Journal Man-machine Studies,* No. 24:65-75.

[15] Zadeh L. A. 1965. Fuzzy Sets, *Information and control.*