

## A Negotiation Protocol to Support Agent Argumentation and Ontology Interoperability in MAS-based Virtual Enterprises

G. Wang, T. N. Wong, X. H. Wang

Department of Industrial and Manufacturing Systems Engineering  
The University of Hong Kong  
Hong Kong

wanggong@hkusua.hku.hk, tnwong@hku.hk, wangxh@hkusua.hku.hk

**Abstract**—The virtual enterprise (VE), which is formed according to some business opportunities through the collaboration of supply chain partners, is an effective way of business operation in the dynamic global market. This paper proposes an automated negotiation protocol for multi-agent system (MAS) based virtual enterprises. Firstly, to facilitate the functional integration of VEs, a MAS framework is developed to represent the whole life cycle of VEs. Secondly, a negotiation protocol supporting agent argumentation and ontology interoperability between agents is designed to handle the one-to-many negotiation scenarios in the VE context. The negotiation ontology is defined to combine the negotiation domain knowledge. Meanwhile, the ontology interoperability and agent argumentation mechanism is illustrated to ensure agents' mutual understandings. Finally, the proposed MAS framework is implemented upon the JADE platform. Simulation experiments are carried out to verify the agent interaction sequences in the negotiation protocol.

**Keywords**—multi-agent system; virtual enterprises; negotiation protocol; ontology; argumentation

### I. INTRODUCTION

The virtual enterprise (VE) is an effective and flexible form of supply chain strategic alliances in the dynamic global market condition. It can be defined as a way of organizing supply chain and manufacturing activities, where different and independent partners exploit business opportunities by establishing an enterprise cooperative [1]. Throughout the whole VE life span, geographically distributed VE participants have to negotiate and cooperate to achieve mutual gains as well as ultimate customer satisfaction.

To deal with the collaboration and interaction problems in this distributed, autonomous and cooperative environment, the multi-agent system (MAS) technology has been explored to portray these features. The MAS-based VE architecture can both facilitate the operational integration of VEs and enhance the business automation.

In the MAS, intelligent agents engage in automated negotiations to solve conflicting issues. In general, the one-to-one bilateral negotiation model has been widely adopted to represent automated negotiations [2]. Agent interaction protocols in the bilateral negotiation models are commonly based on the contract net protocol (CNP) [3] which is

defined for the one-to-one bidding scenario. For instance, in E-commerce applications, the bilateral negotiation protocol regulates one buyer agent and one seller agent to bargain upon conflicting issues iteratively until a common agreement is achieved. In the VE, multiple supply chain partners may have to participate in one-to-many buyer-seller or seller-buyer multilateral negotiations. The simple bilateral negotiation protocol is not suitable for the more complex environment.

In most of the situations, the VE participants are familiar with each other because of the short-term or long-term cooperation. The mutual trust or reliability enables them to exchange additional information during negotiations so as to enhance the negotiation effectiveness. To reflect this reality, the negotiation model in the VE context should better be shifted from competitive games to cooperative argumentations. Therefore, the negotiation protocol for VE application should be specially designed for supporting the agent argumentation.

The pre-requisite of agent-based automated negotiation is that agents executing in different hosts must have a common vocabulary of negotiation items and commit to some common rules of encounter [4], otherwise they cannot understand each other. To organize the negotiation terminologies, the concept of ontology [5] has been adopted in the automated negotiation domain. This approach will enhance agents' awareness of negotiation knowledge, and ideally elevate agents' flexibility towards the changing environment. For research on ontology-based negotiations, there is still a lack of suitable negotiation protocols to facilitate ontology interoperability between agents.

The main objective of this paper is to propose an agent negotiation protocol for the MAS-based VE architecture which can support agent argumentation and ontology interoperability in the VE context. The rest of this paper is structured as follows. In section 2, a MAS-based VE architecture is set up to represent major functions in the whole VE life cycle. Section 3 proposes the agent negotiation protocol and negotiation ontology, and also describes the ontology interoperability and agent argumentation mechanism. Section 4 shows the system implementation and experimental results. Finally, in section 5, conclusions are drawn, with suggestions for future research.

## II. THE MAS ARCHITECTURE FOR VE

The four-phase VE life cycle concept [6, 7] has been adopted in the proposed MAS architecture. Accordingly, the four phases are: identification, formation, operation and dissolution.

In the identification phase, the business opportunities are identified through market analysis of the VE initiator. In the formation phase, qualified VE partners will be selected from potential VE partners. In the operation phase, VE partners will perform the assigned contractual responsibilities coordinated by the VE initiator. In the dissolution phase, the VE is to be dissolved after all the transaction processes of products and services have been accomplished.

The MAS architecture for VEs is shown in Figure 1. The aforementioned VE functioning activities are integrated into six types of agents: task decomposer agent (TDA), coordinator agent (CA), knowledge manager agent (KMA), buyer agent (BA), seller agent (SA), and performance evaluator agent (PEA). The VE initiator comprises all the six agents, while VE partners are simplified to be a buyer or seller agent.

Agent functions are generally described as follows.

The task decomposer agent (TDA) is responsible for decomposing market opportunities and scheduling plans into subsidiary single tasks and sending the tasks to the coordinator agent for negotiation.

The coordinator agent (CA) takes the role of process organizer throughout the whole VE life cycle. A separate coordinator agent instance (CA<sub>i</sub>) is in charge of a single task. In the VE formation phase, it delivers knowledge queries to the knowledge manager agent and initializes the buyer or seller agent to carry out negotiations according to the related negotiation knowledge. In the operation phase, the CA will reschedule the production process and renegotiate with the

VE partners if inconsistencies are detected from the partners' performances. In the dissolution phase, it will release all the agents when receiving the ending signal.

The knowledge manager agent (KMA) is designed for handling the knowledge related activities, including knowledge recording, classifying and retrieving. Knowledge governed by the KMA is to be organized in the form of ontology. When the knowledge queries are received from the CA, the KMA will retrieve the knowledge within the knowledge base. In the VE dissolution phase, it will record all the performance information and send the ending signal.

The buyer agent (BA) and seller agent (SA) are negotiation counterparts which have the same functionality. They perform negotiation procedures for the benefit of trading opponents just as the buyer and seller bargaining in the real world. If the VE formation requirement involves procurement or contract manufacturing type of tasks, the CA will initialize a BA. Conversely, the SA will be initialized when the requirement involves distribution tasks (e.g. supplying products to distributors). Negotiations are carried out between each pair of BA instance (BA<sub>i</sub>) and SA or SA instance (SA<sub>i</sub>) and BA, until the mutual agreement is achieved. In the VE formation phase, final negotiation outcomes will be sent back to the CA for selecting the bidding winners from the potential VE partners. The bidding winners will be contracted as VE partners. In the operation phase, the BA and SA can still negotiate if there are inconsistencies in the manufacturing and delivery processes between VE initiator and partners.

The performance evaluator agent (PEA) is responsible for evaluating VE partners' performances when the performance data are collected in the VE operation phase. If inconsistencies are detected from the partners' performances, the PEA will inform the CA.

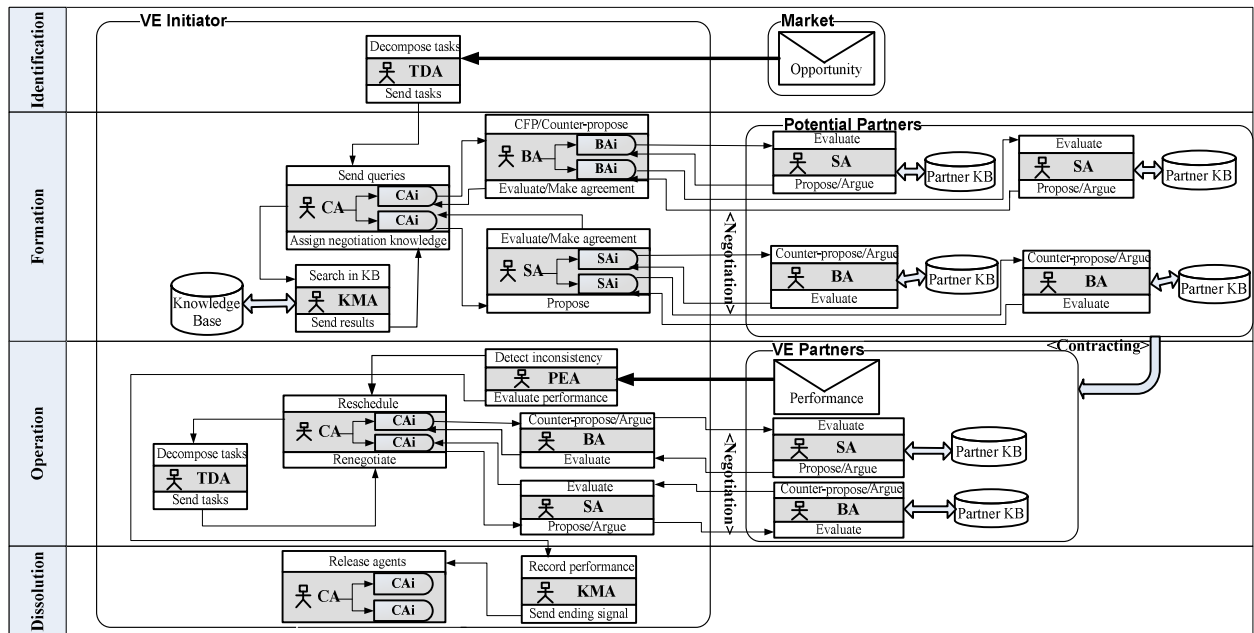


Figure 1. The general MAS architecture for VEs.

### III. THE NEGOTIATION PROTOCOL AND ONTOLOGY

Generally, there are three broad areas needed to be considered when building a sophisticated agent negotiation model [8]: negotiation issues, negotiation protocol and negotiation strategies. The negotiation protocol defines the “rule of encounter” between agents. In the proposed negotiation protocol, new encounter rules are defined to regulate the agent argumentation and ontology interoperability. The negotiation ontology is introduced to incorporate all the negotiation items.

#### A. The Negotiation Protocol

The proposed negotiation protocol is a one-to-many protocol supporting agent argumentation and ontology interoperability. Figure 2 shows the interaction diagram of the proposed negotiation protocol in the form of AUML (Agent Unified Modeling Language). Extended from the CNP, three major points of adjustment are made. Firstly, to handle the one-to-many negotiation scenario, a coordinator is introduced into the protocol. It initializes and governs the buyer instances to negotiate with multiple sellers. The one-to-many negotiation scenario can be converted to many one (buyer instance)-to-one (seller) negotiations. Secondly, the “Argue” interaction sequence is added. It encloses agents’ argumentations for their current situations or against the opponent’s requests. Thirdly, to support the ontology interoperability, a knowledge manager is also introduced into the protocol. It involves all the ontology definitions of different negotiation participants. Messages between the buyer and seller are needed to be translated through the knowledge manager to ensure agents’ mutual understanding.

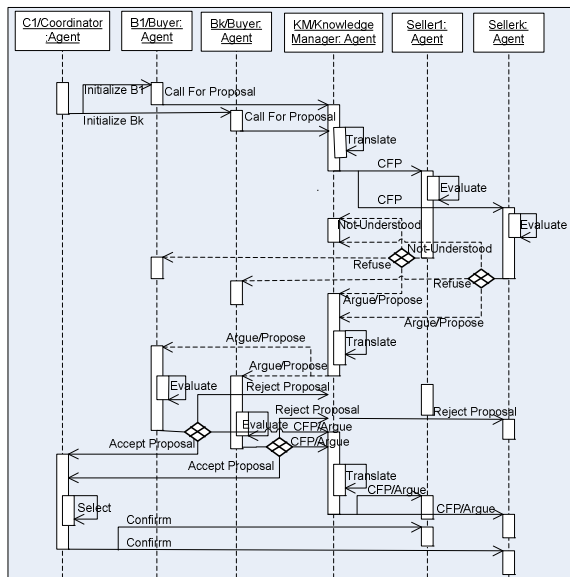


Figure 2. The AUML diagram of negotiation protocol.

As depicted in Figure 2, the CFP/Counter-propose and Propose messages are transferred between each pair of buyer agent instance and seller agent. If an agent identifies that it is not appropriate to propose an offer, it can argue about its

current situation using the Argue message. These messages are handled through the Translate action of the knowledge manager. The Translate action involves mapping ontologies and translating message contents between different agents. When a new message arrives, the buyer or seller agent evaluates the content of the message via the Evaluate action and then delivers reaction. The coordinator’s Select action is for the winner selection in the VE formation phase.

#### B. Negotiation Ontology Definition

The negotiation ontology defines the basic terminology that permits agents to negotiate. In order to support FIPA (Foundation for Intelligent Physical Agents) terminologies, and the complex communication between agents, the relevant elements in the negotiation domain will be classified into concepts and agent actions, according to their generic semantic characteristics. Concepts are entities with complex structures described in terms of attributes. The attributes can be either basic items with specific data types (e.g. integer, string, and etc.) or subclasses inherited from other concepts. Agent actions are special concepts indicating actions that agents can perform.

In the proposed negotiation ontology, concepts are defined as Order, Quantity, Price, Duedate, Warranty, Payment, Issue, IssueStatus, AID (Agent Identifier), Strategy and Negotiation, while agent actions are defined as InitializeNegotiation, InitializeStrategy, Argue and Negotiate. Figure 3 displays four agent actions and five related concepts using the Ontoviz Protégé (an ontology editing framework) plug-in [9].

Regarding the InitializeNegotiation agent action, an agent initializes a round of negotiation under the current negotiation status. Agents use this action to clarify the negotiation variables such as negotiation participants, negotiation issues, deadline and so on, and then assign these attribute values to the related Negotiation concept.

The InitializeStrategy action means that an agent prepares to assign a negotiation strategy to a specific issue against a negotiation counterpart. The negotiation strategy may be elicited from the historical strategy database in consideration of the current negotiation status. After all the attributes in the Strategy concept are decided, this strategy becomes the current active strategy in use.

The Argue action indicates that an agent needs to argue about some status for some reason against its negotiation opponent.

The Negotiate action shows that an agent performing as a negotiation initiator is ready to carry out a round of negotiation with a negotiation participant on some negotiation issues.

All along these action processes, the AID concept describes the identity of an agent in the agent system.

The Issue concept clarifies the name and value of the current negotiation object. It is the basic concept when configuring the values of negotiation issues.

The IssueStatus concept indicates the current status of a negotiation issue and also the suggested adjustment of that issue. Details of the other concepts are not listed in this paper for simplicity.

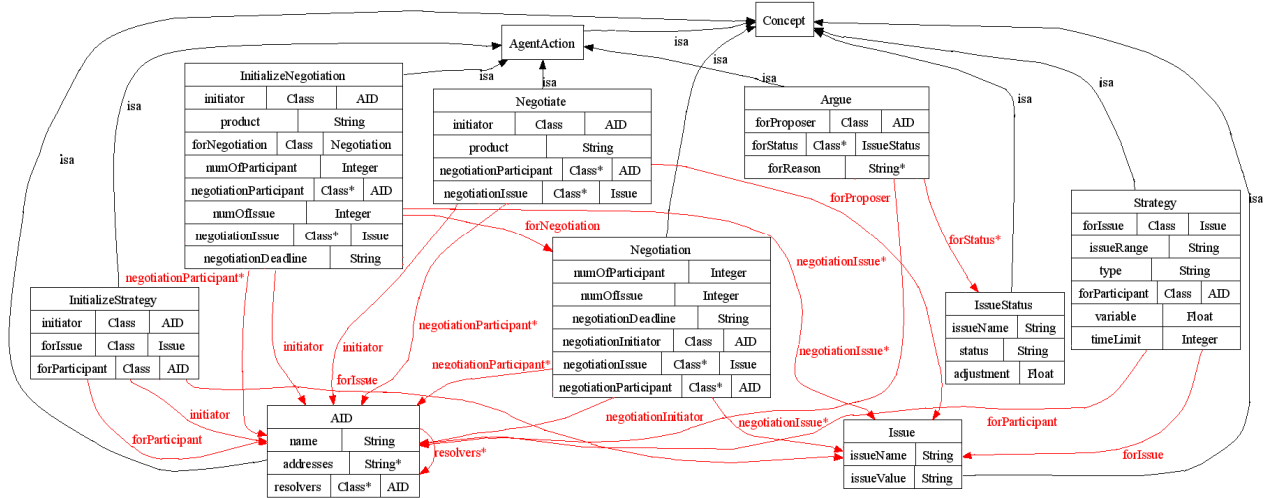


Figure 3. Agent actions and related concepts in the negotiation ontology.

### C. Ontology Interoperability and Agent Argumentation Mechanism

In the VE formation phase, all the ontology structures of potential VE partners will be registered within the knowledge manager agent of the VE initiator. The knowledge manager will do the mapping between the potential VE partners' ontologies and the VE initiator's own ontology structure. Ontology correspondences will be generated accordingly. Details of the ontology matching algorithm are illustrated in the authors' related work [10]. Based on the FIPA ACL message and content language regulation [11], the ontology operability mechanism can be summarized as message content extraction, attribute mapping and message content reassembly.

In a FIPA-compliant agent platform, the contents of messages can be constructed using agent actions defined in the ontology structure. Figure 4 expresses an ACL message involving certain ontology terms.

```
(INFORM
:sender ( agent-identifier :name S1@port3
:addresses (sequence http:// address3))
:receiver (set ( agent-identifier :name KM@port1
:addresses (sequence http://address1 ))
:content "((action (agent-identifier :name S1@port3
:addresses (sequence http://address3))
(Argue
:forProposer (agent-identifier :name B1@port2
:addresses (sequence http://address2))
:forStatus (sequence (IssueStatus
:issueName Price
:status \"out of range\"
:adjustment 1.2))
:forReason \"Material costs increase\")))\"
:language fipa-sl
:ontology Negotiation-ontology
:conversation-id cfp_argue )
```

Figure 4. An example of ACL message.

When a message is received from another agent, the agent action included in the message can be extracted from the message content along with its attribute values. After that, both the attribute names and values are visible within the knowledge manager. Then, all the related attributes will be replaced by their correspondences according to the ontology mapping correspondences.

For the attribute mapping, assuming there are two sets of ontology structures belonging to a buyer agent and a seller agent, namely, buyer-ontology and seller-ontology. The buyer-action in the buyer-ontology and seller-action in the seller-ontology are correspondence agent actions. The buyer-action involves an attribute named buyer-attribute, while the seller-action involves an attribute named seller-attribute, and they are mapping correspondences. Replacing the buyer-attribute by the seller-attribute, the buyer-ontology can be mapped into the seller-ontology.

After all the matching attributes are replaced in the correspondence action, the content of a new message will be filled using this action. At this end, an out-going message is reassembled and will be sent to the negotiation opponent. That is to say, translation has been operated between two sets of ontology structures. The following codes in Figure 5 simply illustrate the ontology interoperability mechanism.

```
receive (oldMessage) ;
oldAction = extractContent (oldMessage) ;
if (oldAction.equals (buyer-
ontology.buyer-action)) {
newAction = seller-ontology.
seller-action;
if (oldAction.getAttribute.equals (buyer-
attribute)) {
newAction.setAttribute (seller-
attribute); } }
fillContent (newMessage, newAction);
send (newMessage) ;
```

Figure 5. Ontology interoperability mechanism.

In the protocol, a buyer or seller agent has the opportunity to argue against an incoming request. When receiving an opponent's request (e.g. CFP), the agent checks its mental state to decide if the request goes beyond its capacity or tolerance, if so, it will give a reason for the argument and suggest a possible adjustment of the request. The negotiation opponent may readjust its request in accordance with the argument.

The "Argue" interaction sequence in the protocol can be expressed using the ACL message "INFORM" and the "Argue" action in the negotiation ontology. As an example, Figure 4 shows the message structure of an argumentation (arguing about the status of price).

#### IV. SYSTEM IMPLEMENTATION

The proposed multi-agent negotiation protocol for VE formation phase has been implemented upon the JADE platform. It is a prevalent agent-oriented middleware supporting FIPA regulation and ontologies.

##### A. A Simulation Case

A simulation case is designed for the partner selection negotiation in the VE formation phase. In the simple case, one VE initiator (VI) is going to procure product A to fulfil a market opportunity. Through negotiations, the VI needs to select one partner from two potential VE partners (VP1 and VP2) to form a simple VE. VP1 and VP2 are confronting a situation of material costs increase, so their agents are prepared to argue about this situation if VI's query price is too low. Moreover, VI, VP1 and VP2 have different ontology structures in their agent systems which may lead to misunderstanding.

It is obvious that negotiation in this case involves one buyer and two sellers. Four types of agents are developed, which are coordinator agent (CA), knowledge manager agent (KMA), buyer agent (BA) and seller agent (SA). The CA, KMA and BA belong to VI, while SAs represent VP1 and VP2.

Three sets of ontologies are involved in this simple VE. For the VI, its ontology is defined with the same negotiation ontology structure introduced in Section 3. For VP1 and VP2, two respective seller ontologies are defined. A negotiation strategy database table is set up using MySQL to store BA's historical strategies against familiar partners.

##### B. System Execution Results

Experiments have been carried out to test the functioning of the system. All the agents run in the same container. CA1, KMA1, SA1 and SA2 are the local names of CA, KMA and SA. BA1 and BA2 are two instances of BA. Two experiments, Exp1 and Exp2, are set up to test the effectiveness of the argumentation mechanism in the protocol. Exp1 uses the simple bidding protocol, while Exp2 adopts the argumentation-based protocol.

Firstly, order information is input into the system. In both Exp1 and Exp2, price and due date are the two negotiable issues in the order. Buyer and sellers' initial value settings of the issues are listed in Table 1.

TABLE I. INITIAL SETTINGS OF NEGOTIATION ISSUES

	BA(BA1 and BA2)	SA1	SA2
Price	168	280	270
Due date	46	63	60

CA1 is initialized by VI's order data, it then searches out two seller agents with local names SA1 (belonging to VP1) and SA2 (belonging to VP2) within the DF (Directory Facilitator). Subsequently, CA1 initializes two buyer instances BA1 and BA2 and informs them about the initial negotiation values. Then, BA1 and BA2 will start to negotiate with SA1 and SA2 respectively.

Figures 6 and 7 show the initial tracing results of the agent interaction sequences in Exp1 and Exp2 respectively. The difference between them is that SAs in Exp2 can send INFORM messages to argue that the price values in the CFP messages are too low. The BA can then readjust the price values referring to SAs' arguments in their next CFP message. As shown in the diagrams, the CFP, PROPOSE and INFORM messages have been translated through the KMA before sent to the opponent.

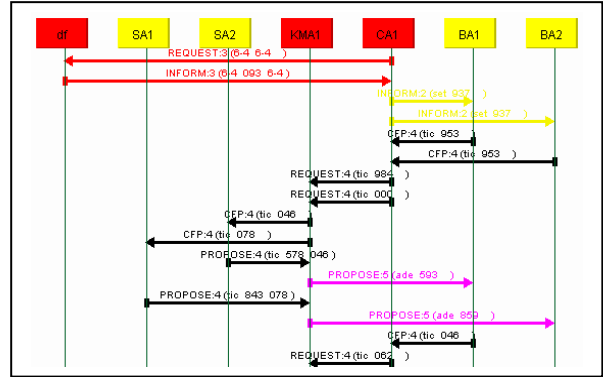


Figure 6. Tracing diagram of Exp1.

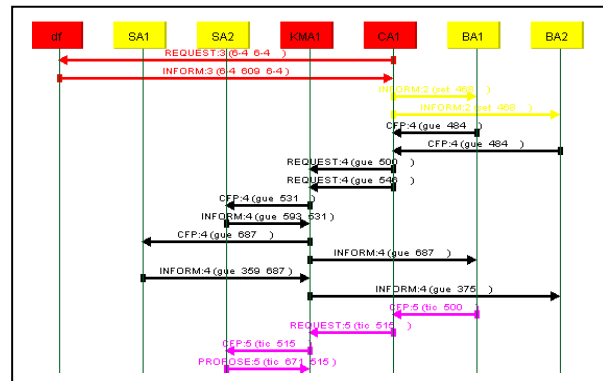


Figure 7. Tracing diagram of Exp2.

Table 2 shows the negotiation results in one simulation run for Exp1 and Exp2. In Exp2, the SAs' arguments for price are similar to the expression in Figure 4. The BA instances readjust the price values accordingly. For their next CFPs, the price values are increased.

TABLE II. NEGOTIATION RESULTS FOR EXP1 AND EXP2.

	Exp1		Exp2	
<b>Time limit</b>	TL <sub>SA1</sub> =296360ms TL <sub>BA1</sub> =283063ms	TL <sub>SA2</sub> =297204ms TL <sub>BA2</sub> =283063ms	TL <sub>SA1</sub> =297172ms TL <sub>BA1</sub> =275235ms	TL <sub>SA2</sub> =297110ms TL <sub>BA2</sub> =275235ms
<b>BA's first CFP issue values</b>	Price <sub>BA1</sub> =168 Duedate <sub>BA1</sub> =46	Price <sub>BA2</sub> =168 Duedate <sub>BA2</sub> =46	Price <sub>BA1</sub> =168 Duedate <sub>BA1</sub> =46	Price <sub>BA2</sub> =168 Duedate <sub>BA2</sub> =46
<b>BA's adjusted issue values upon SAs' argument</b>	\		Price <sub>BA1</sub> =201.86 Duedate <sub>BA1</sub> =46	Price <sub>BA2</sub> =203.08 Duedate <sub>BA2</sub> =46
<b>Negotiation strategies</b>	Fixed concession rate 0.0013 or time dependent concession		Fixed concession rate 0.0013 or time dependent concession	
<b>Final agreement</b>	No agreement within the time limit		Price <sub>SA1</sub> =243.28 Duedate <sub>SA1</sub> =53	Price <sub>SA2</sub> =248.54 Duedate <sub>SA2</sub> =56
<b>Negotiation rounds</b>	R <sub>BA1-SA1</sub> =491	R <sub>BA2-SA2</sub> =511	R <sub>BA1-SA1</sub> =294	R <sub>BA2-SA2</sub> =238
<b>Order winner</b>	None		SA1	

It can be seen that the argumentation facilitates agents to reach the final agreement. In Exp1, no seller wins the order. In Exp2, SA1 finally wins the order with lower price and shorter due date, and is selected as the VE partner.

The system execution results verify that agents can interact smoothly under the proposed negotiation protocol. The proposed negotiation protocol can support ontology interoperability and agent argumentation in the context of VEs.

## V. CONCLUSIONS AND FUTURE WORK

This paper presents a multi-agent negotiation protocol for the MAS-based VEs. The negotiation protocol has been developed to handle the one-to-many negotiation scenario. It is able to support agent argumentation and ontology interoperability. In the MAS architecture, six types of intelligent agents have been designed to perform different functional activities in the VE life cycle. Negotiation ontology has also been defined to combine the negotiation domain knowledge and clarify the relationships between concepts and agent actions. The proposed negotiation protocol has been implemented and tested in the JADE platform. The system execution results can verify the interaction sequences in the negotiation protocol.

For the future work, firstly, the conflict solving negotiation in the VE operation phase will be implemented upon the JADE platform. Secondly, more simulation experiments in various negotiation situations will be designed to further evaluate the effectiveness and efficiency of the system architecture and negotiation protocol.

## REFERENCES

- [1] L. M. Camarinha-Matos, H. Afsarmanesh, C. Garita, and C. Lima, "Towards an architecture for virtual enterprises" *Journal of Intelligent Manufacturing*, vol. 9, pp. 189-199, 1998.
- [2] S. Talluri, "A buyer-seller game model for selection and negotiation of purchasing bids," *European Journal of Operational Research*, vol. 143, pp. 171-180, 2002.
- [3] R. Smith, "The contract net protocol: high-level communication and control in a distributed problem solver," *IEEE Transactions on Computers*, vol. 29, pp. 1104-1113, 1980.
- [4] V. Tamma, S. Phelps, I. Dickinson, and M. Wooldridge, "Ontologies for supporting negotiation in e-commerce," *Engineering Applications of Artificial Intelligence*, vol. 18, pp. 223-236, 2005.
- [5] R. Studer, V. R. Benjamins, and D. Fensel, "Knowledge engineering, principles and methods," *Data and Knowledge Engineering*, vol. 25, pp. 161-197, 1998.
- [6] L. Mikhailov, "Fuzzy analytical approach to partnership selection in formation of virtual enterprises," *The International Journal of Management Science*, vol. 30, pp. 393-401, 2002.
- [7] T. Y. Kim, S. Lee, K. Kim, and C. H. Kim, "A modeling framework for agile and interoperable virtual enterprises," *Computers in Industry*, vol. 57, pp. 204-217, 2006.
- [8] P. Faratin, C. Sierra, and N. R. Jennings, "Negotiation decision functions for autonomous agents," *International Journal of Robotics and Autonomous Systems*, vol. 24, pp. 159-182, 1998.
- [9] Ontoviz 1.0, <http://protege.stanford.edu/plugins/ontoviz>
- [10] X. H. Wang, T. N. Wong, and G. Wang, "Agent ontology interoperability approach for MAS negotiation in virtual enterprises," Proc. Second International Conference on Agents and Artificial Intelligence, Jan. 2010.
- [11] FIPA, Foundation for Intelligent Physical Agents, <http://www.fipa.org>