Agent Communication Language for Automated Negotiation Online

Mukun Cao  
School of Management, Xiamen University, Xiamen, Fujian, China, 361005  
caomukun@yahoo.cn

Abstract—Current research in automated negotiation is lack of a unified technical standard, which is an obstacle for the system to be applied practically. The existing standard KQML (Knowledge Query Manipulation Language) has obvious flaws in its semantics definition; moreover, its function is insufficient for describing negotiation. The contribution of the work is twofold. On one hand, it expands the speech acts of KQML, and defines semantics of the new proposed speech acts. On the other hand, it designs a general interaction mechanism among negotiating agents, in which the negotiation language and ontology used by agents is defined. The work has been attempted in such a way so as to provide general support for a wide variety of commercial negotiation circumstances, and therefore to be particularly suitable for electronic commerce.

Keywords—agent, automated negotiation, agent communication language, semantics

I. INTRODUCTION

Electronic commerce (EC) is affecting business profoundly. It is changing the way businesses interact with consumers, as well as the way the consumers interact with each other. No matter whether it is a case of B to B purchase or a case of online shopping, it becomes more and more important to make the traditional negotiation price mechanism automated and intelligent [1]. Since the agent theory and technology has the characteristics of autonomy, reactivity and sociability, EC oriented automated negotiation has become an important applying field of the agent theory and technology [2] [3].

Research in automated negotiation to date has been focused on the development of negotiation protocols and strategies [4]. For example, Jennings proposed that automated negotiation research can be considered to deal with three broad topics, they are negotiation protocols, negotiation objects and agents decision making models [5]. Negotiation Protocols is the set of rules that govern the interaction. This covers the permissible types of participants, the negotiation states, the events that cause negotiation states to change and the valid actions of the participants in particular states. Negotiation Objects is the range of issues over which agreement must be reached. At one extreme, the object may contain a single issue, while on the other hand it may cover hundreds of issues. Agents’ Decision Making Models is the decision making apparatus the participants employ to act in line with the negotiation protocol in order to achieve their objectives. [5]

A. Background for Agent Communication Language

Although there are many research achievements about protocols and strategies in the field of automated negotiation nowadays, realization and real application of automated negotiation system still has a long way to go [6, 7, 8]. Up to now, most work on automated negotiation still in experimental stage. One important reason for appearing this kind of situation is lack of a unified technology standard, which makes sharing and reusing software very difficult. Take the protocol as an example; almost all the protocols proposed in former literatures are represented by researchers’ own way [9]. It is easy to understand that, in an open environment like e-commerce, it is difficult for negotiating agents using different interaction protocol to negotiate with each other automatically, just because they have not a common and general format for exchanging message.

In order to solve the above problem, some organizations proposed some solutions. At the moment, there are two main technical specifications suitable for the standardization of the agent communication format. One is Knowledge Query and Manipulation Language (KQML) [10]; another is the Foundation for Intelligent Physical Agents (FIPA) specification [11]. Our work is based on KQML, simply because of its technical maturity and current lead in market share. However, KQML has some limitations need to be improved, when it is used in the field of automated negotiation. One is that the performatives of KQML is not enough for negotiation. KQML only has a “tell” speech act to support negotiation. But the actual situation of negotiation is more complex. Negotiation need not only “tell”, but also “accept”, “reject”, “argue” and other speech acts to describe negotiation situation comprehensively and accurately. Another is that it still suffers from poorly defined semantics. As a result, each of the many KQML implementations seems unique. This makes it difficult to communicate with other agents from heterogeneous systems.

The main aim of our work is to find a way to construct an understandable interaction mechanism for agents in an automated negotiation process. Concretely, we hope that the communication model can achieve the following goals. First, negotiating agents should communicate with each other, and understand correctly...
what others say. Second, the negotiating agents should share common concepts about the objects they are negotiating over. These problems are basic and essential for realizing an applicable automated negotiation system.

In order to achieve the first goal, we propose a communication model which is mainly composed of a negotiation communication language. On one hand, negotiation is definitely a linguistic form, and the automated negotiation particularly depends on the interaction or communication between agents in open environments such as the Internet or the Semantic Web [12]. On the other hand, automated negotiation system is in essence a multi-agent system, whose running is heavily dependent on the communication between agents. So we have to solve following problems: What kinds of language do the negotiating agents use when they are in a commercial negotiation process? How to express the negotiation language correctly and formally?

In order to achieve the second goal, we propose negotiation ontology. The ontology is a way of categorizing objects such that they are semantically meaningful to a software agent. For no one can ignore the fact that agents communicating in a common language will still be unable to understand each other if they use different vocabularies for representing shared domain concepts. Ontology is required to ensure that the agents are referring to exactly the same good. Through using negotiation ontology, an agent can understand implications of all variables delivered from other agents. As a result, how to design and express the negotiation ontology is another important problem which must be solved before the negotiation interaction between agents is built.

B. Background for semantics definition

Before the grammar of the negotiation communication language is designed, one problem should be solved in advance. That is semantics. People can judge the true meaning of conversation content through the linguistic context in a conversation process. So far, computer has not such intelligence yet. As a result, if a communication language is submitted to computer to process, it is necessary to give the language a clear and accurate semantics definition.

It is significant for the negotiating agent communication language to use highly formalized semantics definition. It is mainly because that the agent communication language will be used in many different applying situations and inhomogeneous systems. The semantics definition with standard form can make the meaning of speech act to be clear, and can guarantee various application designers have a common understanding and explanation about the language, performatives and protocols. Then the applications will be developed under these stipulations. So, it is significant for realizing interoperation among inhomogeneous systems.

A typical example is KQML (Knowledge Query and Manipulation Language). When the KQML is designed at first, it is lack of a formalized semantics explanation, which incurs a widespread criticism. In view of this kind of situation, Labrou and Finin gave the semantic description for KQML, which is based on a mental state frame. The frame defines the speech act from the following 6 aspects: semantic description using natural language, formal semantic description, Pre-condition, Post-condition and Completion. For example, “tell (A,B,X): A informs B that A believe content X is true, namely BEL(A,X), where:

\[
\begin{align*}
\text{Pre}(A): & \quad \text{KNOW}(A,\text{WANT}(B,\text{KNOW}(B,S))) \\
\text{Pre}(B): & \quad \text{INT}(B,\text{KNOW}(B,(\text{BEL}(B,X) \lor \neg\text{BEL}(B,X)))) \\
\text{Post}(A): & \quad \text{KNOW}(A, \text{KNOW}(B, \text{BEL}(A,X))) \\
\text{Post}(B): & \quad \text{KNOW}(B, \text{BEL}(A,X)) \\
\text{Completion}: & \quad \text{KNOW}(B, \text{BEL}(A,X))
\end{align*}
\]

However, the above work made by Labrou and Finin didn’t use standard BDI model [14]. In their definition, they not only use BEL, WANT and INT, which come from standard BDI model, but also introduce a modal operator KNOW, which denotes that an agent knows whether a proposition is true or false. But it is not a standard BDI modal operator, which builds the semantics of KQML on a non-standard mental state frame. When an agent based on BDI model uses KQML defined like that, it needs to transform its mental state into a new expression way to fit the KQML. But this transform possibly will cause new ambiguity.

As a result, when we propose a negotiation communication language expanded from KQML, we cannot use the above semantics to explain the language, even we won’t plan to use the BDI model as the theoretical fundament, because BDI is a common theoretical model for solving general problems in the field of agent reasoning. But negotiation, as an important part of agent’s cooperation, has its particularity different behavior attributes. BDI and its subsequent theory model, however, don’t give negotiation enough regard, so the current model cannot satisfy the development of automated negotiation. We therefore proposed a novel agent theoretical model in literature [15] to fit the semantics definition of negotiation language.

In a word, the ultimate aim of researching automated negotiation is to realize a real and practicable automated negotiation system. Taking advantage of existing software agent technology is a good way for developing an automated negotiation system. Moreover, most agent development platform is based on the agent communication language. As a result, the work presented in this paper is helpful for realizing a practical automated negotiation system.

The remainder of this paper is organized as follows. Section 2 defines the semantics of the negotiation language based on the BDI-U model. Section 3 describes the frame of the communication model for automated negotiation. Section 4 discusses the negotiation language used by the agents during the negotiation process. Section 5 presents the design and expression of negotiation ontology which is an important component in the negotiation language. Finally, Section 6 draws conclusions and presents future work.
II. SPEECH ACT DEFINITION

In order to describe what the agents should say in different circumstances, we define negotiation and dialog speech act. The negotiations and dialog speech act is an expansion to the performatives defined in KQML, which contains tell, propose, argue, accept, reject, ask-if, error. Table I shows details of these speech acts and their concrete meaning. Where S indicates sender agent, R indicates receiver agent, and F indicates facilitator (communication server).

**TABLE I. AUTOMATED NEGOTIATION’S SPEECH ACTS**

<table>
<thead>
<tr>
<th>Speech Act</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tell</td>
<td>S let R known some of his own knowledge in his own belief base.</td>
</tr>
<tr>
<td>propose</td>
<td>S let R know his proposal about a certain round of negotiation</td>
</tr>
<tr>
<td>argue</td>
<td>S let R know the reason for proposing the proposal in a certain round of negotiation</td>
</tr>
<tr>
<td>accept</td>
<td>S let R know he accept R’s last proposal</td>
</tr>
<tr>
<td>reject</td>
<td>S let R know he reject R’s last proposal</td>
</tr>
<tr>
<td>ask-if</td>
<td>S want to know whether some knowledge is in the R’s belief base.</td>
</tr>
<tr>
<td>error</td>
<td>S tell R he cannot understand the last message sent by R</td>
</tr>
</tbody>
</table>

Tell has special meaning among the all negotiation speech act. Any speech act in negotiation can be expressed by tell. As a result, we give the semantic definition of tell first.

**Definition 11:** the semantics of tell is \( \text{tell}(i, j, \varphi) \)

Prerequisite:
\[ \text{Bel}(i, \varphi) \land \neg \text{Bel}(i, (\text{Bel}(j, \varphi) \lor \text{Bel}(j, \neg \varphi))) \land \text{Int}(i, \text{Bel}(j, \varphi)) \]

Reasonable result: \( \text{Bel}(j, \varphi) \)

Definition 11 shows that \( \text{tell}(i, j, \varphi) \) indicates negotiating Agent \( i \) intends to make the negotiating Agent \( j \) to believe that the proposition \( \varphi \) is true. In order to make the speech act achieve the expected effect, it is necessary to satisfy the following conditions: first, negotiating Agent \( i \) must be confident that proposition \( \varphi \) is true; moreover, negotiating Agent \( i \) considers so far negotiating Agent \( j \) doesn’t believe \( \varphi \) is true, neither believe \( \neg \varphi \) is not true; finally, it is necessary to satisfy that negotiating Agent \( i \) intends to make the negotiating Agent \( j \) to believe that the proposition \( \varphi \) is true. The expected result of the speech act after execution naturally is Agent \( j \) believes \( \varphi \) is true.

**Definition 12:** the semantics of propose is \( \text{propose}(i, j, \varphi \rightarrow \psi) \)

\[ \text{propose}(i, j, \varphi \rightarrow \psi) = \text{tell}(i, j, \text{Int}(i, \varphi) \land \psi) \]

Prerequisite:
\[ \text{Bel}(i, \alpha) \land \neg \text{Bel}(i, (\text{Bel}(j, \alpha) \lor \text{Bel}(j, \neg \alpha))) \land \text{Int}(i, \text{Bel}(j, \varphi)) \]

Reasonable result: \( \text{Bel}(j, \alpha) \) , where, \( \alpha = \text{Int}(i, \varphi) \land \psi \).

Definition 12 shows that \( \text{propose}(i, j, \varphi \rightarrow \psi) \) indicates negotiating Agent \( i \) proposes a proposal \( \varphi \) to Agent \( j \), and provide an argument \( \psi \) to explain his action; the semantics of propose can be defined via speech act \( \text{tell} \); negotiating Agent \( i \) want to tell negotiating Agent \( j \) that he intends to propose the proposal \( \varphi \), and can offer argument \( \psi \). Similarly with \( \text{tell} \) we can explain the prerequisite and reasonable result of propose.

**Definition 13:** the semantics of accept is \( \text{accept}(i, j, \varphi \rightarrow \psi) \)

\[ \text{accept}(i, j, \varphi \rightarrow \psi) = \text{tell}(i, j, \text{Int}(i, \text{Int}(j, \varphi))) \]

Prerequisite:
\[ \text{Bel}(i, \alpha) \land \neg \text{Bel}(i, (\text{Bel}(j, \alpha) \lor \text{Bel}(j, \neg \alpha))) \land \text{Int}(i, \text{Bel}(j, \varphi)) \]

Reasonable result: \( \text{Bel}(j, \alpha) \) , where \( \alpha = \text{Int}(i, \text{Int}(j, \varphi)) \)

Definition 13 shows that \( \text{accept}(i, j, \varphi \rightarrow \psi) \) indicates negotiating Agent \( i \) notifies Agent \( j \) he accepts the proposal \( \varphi \) proposed by Agent \( j \); the semantics of accept can be defined via speech act \( \text{tell} \); negotiating Agent \( i \) want to tell negotiating Agent \( j \) that he intends to make the proposal \( \varphi \) to be realized, which can satisfy the intention of Agent \( j \). Similarly we can explain the prerequisite and reasonable result of accept.

**Definition 14:** the semantics of speech act reject is \( \text{reject}(i, j, \varphi \rightarrow \psi) \)

\[ \text{reject}(i, j, \varphi \rightarrow \psi) = \text{tell}(i, j, \neg \text{Int}(i, \text{Int}(j, \varphi)) \land \psi) \]

Prerequisite:
\[ \text{Bel}(i, \alpha) \land \neg \text{Bel}(i, (\text{Bel}(j, \alpha) \land \text{Bel}(j, \neg \alpha))) \land \text{Int}(i, \text{Bel}(j, \varphi)) \]

Reasonable result \( \text{Bel}(j, \alpha) \), where \( \alpha = \neg \text{Int}(i, \text{Int}(j, \varphi)) \land \psi \)

Definition 14 shows: \( \text{reject}(i, j, \varphi \rightarrow \psi) \) indicates that negotiating Agent \( i \) notifies Agent \( j \) he rejects the proposal \( \varphi \) proposed by negotiating Agent \( j \), and at the same time, offers an argument \( \psi \). The semantics of reject can be defined via speech act \( \text{tell} \); negotiating Agent \( i \) want to tell negotiating Agent \( j \) that he doesn’t intend to make Agent \( j \)’s intention of proposing the proposal \( \varphi \) to be realized. Similarly we can explain the prerequisite and reasonable result of reject.

**Definition 15:** the semantics of speech act ask-if is \( \text{ask-if}(i, j, \varphi) \)

\[ \text{ask-if}(i, j, \varphi) = \text{Int}(i, \text{tell}(j, i, \varphi \lor \neg \varphi)) \]

Prerequisite:
\[ \neg \text{Bel}(i, \varphi) \land \neg \text{Bel}(i, \neg \varphi) \land \neg \text{Bel}(i, \text{Int}(j, \text{tell}(j, i, \varphi \lor \neg \varphi))) \]

Reasonable result: \( \text{tell}(j, i, \varphi \lor \neg \varphi) \land \text{tell}(j, i, \neg \varphi) \)

Definition 15 shows: \( \text{ask-if}(i, j, \varphi) \) indicates that negotiating Agent \( i \) asks Agent \( j \) whether the proposal \( \varphi \) is true. The semantics of ask-if can be defined via speech act \( \text{tell} \); negotiating Agent \( i \) intends to make the negotiating Agent \( j \) to tell him whether the proposal \( \varphi \) is true or not. In order to make the speech act achieve the expected effect, it is necessary to satisfy the following conditions: first, negotiating Agent \( i \) doesn’t believe that \( \varphi \) is true, neither believe \( \neg \varphi \) is false. Second,
negotiating Agent $i$ considers that so far negotiating Agent $j$ doesn’t intend to tell him the truth about whether $\varphi$ is true or not. The expected execution result of the speech act is naturally that Negotiating Agent $j$ tells Negotiating Agent $i$ the truth about whether $\varphi$ is true or false.

**Definition 16:** the semantics of speech act argue is $\text{argue}(i,j,\varphi) = \text{tell}(i,j,\varphi)$

**Prerequisite:**
$\text{Bel}(i,\varphi) \land \neg \text{Bel}(i,(\text{Bel}(j,\varphi) \lor \text{Bel}(j,\neg \varphi))) \land \text{Int}(i,\text{Bel}(j,\varphi))$

**Reasonable result:** $\text{Bel}(j,\varphi)$

Definition 16 shows: $\text{argue}(i,j,\varphi)$ indicates that negotiating Agent $i$ provides argument $\varphi$ to negotiating Agent $j$. Negotiating Agent has two ways to perform argument based negotiation through the speech act propose and argue. The difference between the two speech acts is that propose can be used to give an argument while making a proposal; argue, however, can only be used to express the argument. We can use the two speech acts according to the actual situation. Similarly with tell, we can explain the prerequisite and reasonable result of argue.

**III. NEGOTIATING AGENT COMMUNICATION MODEL**

Negotiating Agent Communication Model (NACM) is designed for solving three problems. First is entity which possibly participates in the negotiation process. They are Buyer Agents, Seller Agents and Facilitator in the model. Second is communication language possibly used by the agents. Third is the logical relation between each language. However, NACM shouldn’t be regarded as an agent interaction protocol, for there are many mature protocols which can be used nowadays. The main aim is to focus on an explicit definition of the language which the agents can use in any negotiation protocol.

In order to describe what the agents should say in different circumstances, there are two kinds of meta-languages: negotiation and dialog speech act, and network service speech act. The negotiations and dialog speech act has been defined above, here we give the network service speech act inherits from KQML, which contains register, unregister, advertise, subscribe, recruit-one, recruit-all, forward. Table 2 shows details of these speech acts and their concrete meaning. Where S indicates sender agent, R indicates receiver agent, and F indicates facilitator (communication server).

**TABLE II. NETWORK SERVICE SPEECH ACTS**

<table>
<thead>
<tr>
<th>Classification</th>
<th>Speech Act</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Service</td>
<td>register</td>
<td>S send registration request to F</td>
</tr>
<tr>
<td></td>
<td>unregister</td>
<td>S send logout request to F</td>
</tr>
<tr>
<td></td>
<td>advertise</td>
<td>S declares to R he is fit for handling some affair</td>
</tr>
<tr>
<td></td>
<td>subscribe</td>
<td>S ask R to inform him some affair in future</td>
</tr>
<tr>
<td></td>
<td>recruit-one</td>
<td>S ask F to send a message to a negotiating agent who can handle this message, and is responsible for handling this message</td>
</tr>
<tr>
<td></td>
<td>recruit-all</td>
<td>S ask F to send a message to all negotiating agents who can handle this message, and are responsible for handling this message</td>
</tr>
<tr>
<td></td>
<td>forward</td>
<td>S asks R to forward some message to a negotiating agent which is indicated by the grammar item &quot;to&quot;</td>
</tr>
</tbody>
</table>

KQML is a protocol for carrying and formatting messages based on speech acts theory. It integrates a concept of facilitator [16], which refers to a class of agents who traffic in meta-knowledge about other agents, and provide communication services such as message forwarding and broadcasting, resource discovery, matchmaking and so on. Therefore, it is the main tool for users to construct their own interaction model. Figure 2 is the communication model based on the expanded KQML.

![Communication model in multi-agent automated negotiation system](image)

A bargaining based on multi-agent automated negotiation must experience three stages.
First, Buyer Agents and Seller Agents register on the Facilitator (using performatives register or unregister), which is set up by a third party, providing some basic private information, such as name, host’s IP and port number; the Facilitator assigns an ID to them at a later stage.

As an important step, the Seller Agents will advertise some product information (using performative advertise(ask-if), as showed in fig2) to the facilitator who will help them forward the advertisement to Buyer Agents (using performative forward(ask-if) or tell).

Then, when a Buyer Agent wants something, it will query information from the Facilitator, using performatives recruit-one or recruit-all(ask-if).

If the Facilitator is ignorant of the product, the Buyer Agent will send a subscription to the Facilitator for further information, using performatives subscribe(recruit-one) or subscribe(recruit-all(ask-if)).

When the required information is available, the Facilitator will forward it to the claimer. Finally, a communication between the Buyer and Seller is constructed with the help of the Facilitator, and then direct negotiation can be implemented. Here, we extend the original KQML performative, using propose, argue, reject, accept, tell to satisfy the complicated linguistic circumstances occurred in a process of negotiation.

IV. NEGOTIATING AGENT COMMUNICATION LANGUAGE

Automated negotiation relies on the idea that agents must use a shared format of communication in order to interact smoothly. In contemporary multi-agent negotiation systems, different formats of communication have been used in different systems. However, in order to fully exploit the potential of open environments like the Internet, agents should use a united negotiating language which is most suitable to any type of negotiation in which they participate. In fact, there is currently no standard widely accepted specification for the language. Negotiating Agent Communication Language (NACL) aims to find an approach that permits agents to negotiate with most of the negotiation mechanisms. It is a syntax specification for exchanging message between agents. Its user is negotiating Agent. Using NACL, negotiating agents can make synchronous and asynchronous communication with each other through TCP/IP and HTTP.

In order to take advantage of existing resources, the syntax of NACL is compatible with KQML. For NACL is a kind of Agent Communication Language based on message, it defines the formal format of message, which contains, similar to KQML, three layers: communication layer, message layer and content layer.

Communication layer: describes a group of communication parameter related to message communication. The all technical communication parameter is specified in the communication layer. The item "sender" and "receiver" denote respectively the sender and expectant receiver of message.

Message layer: is the core of NACL. It specifies the speech-acts related to message. Every message has at least one speech act performative. In addition, message layer provides description of attributes related to content layer, for example, the language for expressing content, ontology and so on. The item "language" indicates the language used for expressing the content of message. Here we prescribe the knowledge expression language used in NACL can be KIF (Knowledge Interchange Format), Prolog, Lisp, XML and so on. Item "ontology" indicates ontology used in message.

Content layer: specifies the content of message using item "content". The content is expressed by knowledge expression language which is defined by item "language" in message layer.

In terms of interaction scene described in the communication model mentioned above, NACL can be divided into 10 meta-languages in detail. To do so, NACL is defined formally as the meta-negotiation language according to the KQML language specification, and expressed in BNF style as follows:

(1)<registration>::=register | unregister
:sender<Agent_name>
:receiver<Facilitator_name>
:language KIF | Prolog | Lisp | XML
:ontology kqml-ontology
:content<agent_information>

(2)<advertisement>::=advertise
:sender<SellerAgent_name>
:receiver<Facilitator_name>
:language KQML
:ontology kqml-ontology
:content(ask-if

(3)<query>::=recruit-one | recruit-all
:sender<BuyerAgent_name>
:receiver<Facilitator_name>
:language KQML
:ontology<kqml-ontology>
:content(ask-if

(4)<subscription>::=subscribe
:sender<BuyerAgent_name>
:receiver<Facilitator_name>
:language KQML
:ontology kqml-ontology
:content(recruit-one | recruit-all

© 2010 ACADEMY PUBLISHER
performative error can be used when one agent cannot handle messages from other agents, for example, the communicator receives exception cases. Performative error can be handled by the other performerative acts, which are defined as follows:

(1) exception ::= (error

(2) rejection ::= (reject

(3) acception ::= (accept

(4) negotiation ::= (propose | argue

(5) transmission ::= (forward

(6) notification ::= (tell

(7) query ::= (subscribe

(8) offer ::= (offer

(9) product ::= (product

(10) price ::= (price

Although the sender of transmission is facilitator, and receiver is seller agent, the factual meaning it expresses is transmitting content in “ask-if”: (from) buyer Agent:(to) seller Agent.

Negotiation ontology is based on the idea that there are some general concepts that are presented in any negotiation, and it is built on finding commonalities across different negotiation protocols. From an analysis of the classification framework illustrated in [19], the generic software framework for automated negotiation [20], and the work by Samir Aknine, Suzanne Pinson, and Melvin F. [21], we have identified the concepts and the relationships that are shared by most negotiation protocols. Figure 3 shows the negotiation ontology resulting from this merging process using an UML model to represent concepts and relationships.

The ontology is defined in terms of the following concepts; each of them highlights a different aspect of a negotiation:

(1) Classes: concepts related directly to the negotiation, such as negotiation protocol, negotiating Agent, role and negotiation rules and so on. Concretely, Negotiation protocol defines a generic protocol defining the “rules of encounter” that are followed by negotiation participants during a negotiation process. The rules describe the conditions defining the interactions between agents and the deals that can be made [19].

Agent describes a single agent or an organization of agents which participate in a negotiation. Several agents can negotiate, and they can play different roles in the negotiation;

objects describes the objects of the negotiation, that is the material or immaterial goods that are transferred once an agreement has been reached;

Negotiation rule is a set of rules that govern a specific negotiation protocol. In the ontology this means that we identify a number of negotiation rules, and the way in which they are specified defines a specific negotiation protocol. This concept is specified by the different types of rules identified in [21];

Role describes the role that an agent or an organization of agents plays in the ontology. Role represents the participants in the negotiation rule. It is specified by the roles identified in [21];

Offer: in order to express demand, negotiating agent proposes possible value for various issues of negotiation,
for example, the quantity and price of the product.

(2) Relations: describes the mutual effect between concepts in the field of automated negotiation. The basic relation in the model is from the object oriented theory. As fig3 shows, English_Auction class inherits Protocol class. Negotiation_Host and Negotiation_Participant are component of class Role. Class Protocol, Agent and Object have certain relation between them. Besides, some special relation is constructed according to concrete model. For example, there is a relation has Actor between concept Protocol and concept Agent, for describing some negotiating agent interact with each other in a negotiation protocol. The relation plays between concept Agent and Role describes that different Agent plays different role in a process of interaction. Negotiation protocol is controlled by negotiation rules through relation govern.

The relationships between the concepts are also defined to describe how the identified concepts interact to define the negotiation protocol domain. For example, a Protocol Has Agent which models the fact that at least two or more (2...*) agents interact in one negotiation protocol. That A Party Plays Role is modeling the fact that one agent can play a number of (1...*) different roles in the interaction. A protocol is also governed by a number of negotiation rules, and this aspect is also modeled by means of the relationship.

In the process of expressing ontology knowledge, the concept is expressed in detail through constraining or adding concept’s attribute. For example, concept Protocol is described through adding attribute hasActor and hasObject. The following example is based on English Auction, whose ontology is expressed by DAML+OIL language. The following code is a part of whole.

```xml
<daml:ObjectProperty rdf:ID="hasActor">
  <rdfs:domain rdf:resource="#Protocol"/>
  <rdfs:range rdf:resource="#Agent"/>
  <daml:minCardinality>2</daml:minCardinality>
</daml:ObjectProperty>
<daml:Class rdf:about="#English-Auction">
  <rdfs:subClassOf rdf:resource="#Protocol"/>
  <rdfs:subClassOf>
    <daml:Restriction daml:minCardinality="3">
      <daml:onProperty rdf:resource="#hasActor"/>
    </daml:Restriction>
    <daml:Restriction>
      <daml:onProperty rdf:resource="#hasObject">
        <toClass rdf:resource="#Protocol"/>
      </daml:Restriction>
    </daml:Restriction>
  </rdfs:subClassOf>
</daml:Class>
```

© 2010 ACADEMY PUBLISHER
VI. CONCLUSIONS AND FUTURE WORK

For negotiation is a combination of logic and utility, the theoretical model supporting for the negotiation behavior of agent need to combine the utility concept with the agent’s mental states. BDI-U semantics model combined with utility provides theoretical foundation for defining the semantics of agent’s negotiation communication language, and it is helpful for expanding the original semantics of KQML to better suit the situation of multi-agent automated negotiation. The research result of this paper will be meaningful for the development of a practical automated negotiation system, and will be helpful for the research of interaction and communication mechanism for multi-agent negotiation.

In this paper, a novel negotiating agent communication model is established. NACM defined above are explicit and formal specifications for the agents negotiating in an E-business environment. It defines the negotiation language template shared among all agents formally and explicitly. The novelty of the communication model is twofold. In fact it is synthesis work in both agent communication technology and automated negotiation theory, which are important areas of e-commerce research. More importantly, NACM build the foundation for developing an automated negotiation system.

Some approach we have presented in this paper is still at an early stage, and there are a number of issues that need to be further investigated. One is how to perfect the negotiation ontology. Creating and expressing any size of ontology is difficult and time consuming work. We still need to investigate whether this negotiation ontology is sufficient to permit the necessary interaction or whether a different type of knowledge should be included in the ontology. Another aspect which we have disregarded in the paper but which we are planning to investigate is how to design an analogous negotiating model complying with FIPA specifications.

ACKNOWLEDGEMENTS

This work is supported by grant 2009J05162 of the China Fujian Province Natural Science Foundation for Young Researchers, and by grant70902236 of the China National Natural Science Foundation.

REFERENCES


Mukun Cao was boring in Yulin, Shaanxi, China, 1978. He received his B.E. (System Engineering, 2001) and his PhD (Management Science and Engineering, 2007) from Harbin Institute of Technology, China. He is currently an assistant professor in Management Science at Xiamen University, China. His research interests are: management information systems, electronic commerce, agent technology and multi-agent systems.