Decision Making Model for Intelligent Agent in Automated Negotiation

Mu-kun Cao
Management School, Xiamen University, Xiamen, Fujian 361005, China
caomukun@yahoo.cn

Abstract—**Traditional research in automated negotiation is focused on negotiation protocol and strategy, but they cannot satisfy all the requirements for realizing a practical automated negotiation system. This paper focuses on agent's independent decision-making process; formally defines automated negotiation's abstract concept model, which is made up of three sub-concept models, they are negotiation environment, negotiation process and negotiating agent; designs negotiating agent's architecture based on the concept model, which can support both goal-directed reasoning and reactive response; proposes an algorithm for running the decision making model. Finally, for illustration, the model is applied to an exemplified negotiation process for aircraft purchasing.**

Index Terms—**Automated negotiation, agent, multi-agent system, belief-desire-intention model**

I. Introduction

Today, e-commerce is characterized by dynamic trade. Its role has not only been confined to provide trading places and related information for the buyers and sellers, but has changed from a simple trade matcher to a price coordinator [1]. The tremendous successes of online auctions show that the dynamic trade based on e-negotiation will gradually become the core of e-commerce [2]

Research in automated negotiation to date has been focused on the development of negotiation protocols and strategies [3]. For example, Jennings considered that automated negotiation research can be considered to deal with three broad topics, they are negotiation protocols, negotiation objects and agents decision making models [4]. Recent work is mainly focused on how to construct a negotiation model, which has ability to control the whole process of negotiation, to balance conflict of interest, not on providing support to unilateral negotiator's decision-making. 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 [5, 6]. The reason cannot be just found from research of negotiation protocol and strategies, but from the lack of research on negotiating agents.

Obviously, negotiation protocol and strategy algorithm is heavily dependent on the construction of an agent. Especially, negotiation strategy can be regarded as a function module in agent's architecture. Then, how do the agents comply with certain protocols and execute certain strategies? Agent is a rational entity with *Belief*, *Desire* and *Intention*. In other words, how do the protocol and strategy take part in the BDI reasoning? We consider that the research of negotiation protocols and strategies cannot answer the above problems that must be solved properly when a practical automated negotiation system will be realized.

Negotiating agent is negotiation's executor, whose responsibility is to implement certain negotiation strategies complying with certain negotiation protocols. As a result, it plays an important role in realization of automated negotiation system. Negotiating agent has has some special attributes; for example, it mainly uses Speech Act to interact with other agents and environment, simply because negotiation is a kind of linguistic behavior. Therefore, negotiating agent's concept shouldn't be replaced or covered up by traditional agent's concepts, but existing theory and technology of agent can be applied for research of negotiating agent.

The main aim of the work is to find a way to construct the decision making model for the negotiating agent in automated negotiation. The remainder of this paper is organized as follows. In order to explain the negotiating agent's concept clearly from mathematical point of view, Section 2 proposes a conceptual model for automated negotiation. This section divides into four sub-sections. Section 2.1 defines concept of negotiation environment; section 2.2 defines concept of negotiation process; section 2.3 defines concept of negotiating agent; and section 2.4 run the whole system based on the above three concepts. On the base of the negotiating agent's concept model, Section 3 proposes negotiating agent's architecture. Section 4 propose an algorithm for running the architecture and the decision making model. Section 5 illustrates an exemplified negotiation process for aircraft purchasing to apply the model. Finally, Section 6 draws conclusions and presents future work.

II. Automated Negotiation Concept Model

Recent theoretical work about agent has clarified the role of goals, intentions, and commitment in constraining the reasoning that an agent performs [7]. There has been some work about constructing abstract model for agent. From an analysis of literatures [8, 9], we found that these former works are mainly focused on the description of agent's decision process, seldom paid attention to its surrounding environment. More over, these models are quite general. Automated negotiation, however, is a special application area, and the negotiating agent's

doi:10.4304/jsw.6.8.1537-1544

behavior is heavily depending on the negotiation's environmental factors. Therefore, it needs particular model to fit these requirements. So, we hope the automate negotiation model presented below has the following abilities:

It can roundly describe elements of environment with which the negotiation deals.

It can effectively describe the negotiation's dynamic characteristics.

It can formally define the abstract architecture of negotiating agent.

In order to satisfy the above requirements, the abstract model of multi-agent automated negotiation system is divided into three sub-concept models; they are Negotiation Environment, Negotiation Process and Negotiating Agent.

A. Negotiation Environment

The concept model of Negotiation Environment (NE) is constructed on the basis of Negotiation Protocol. The formal definition is as follows.

Definition 1(Negotiation Environment): a Negotiation Environment is a form

$$NE =< A, P, TP, O, V, SA, Message, T, Thread > \tag{1}$$

where:

$A = \{1, 2, \cdots, n\}$ is the set of code name representing negotiating agent.

P is the negotiation protocol expressed by a certain Agent Communication Language (ACL), such as KQML or FIPA ACL.

$TP = \{tp_1, tp_2, \cdots tp_n\}$ is the set of negotiation topics, in multi-attribute negotiation, for example, the topics can be price, quantity or quality.

$O = \{o_1, o_2, \cdots o_n\}$ is the set of ontology, which is the category of concept relating to certain topics. It consists of concepts and structure, which is made up of concepts.

V is the set of effective value of topics determined by *O*.

$SA = \{\alpha_1, \alpha_2, \cdots, \alpha_n\}$ is the finite set of executable speech-acts. Negotiating agents are assumed to have a repertoire of possible Speech-Acts available to them, which transform the state of the negotiation. Generally, the element in this set is fixed and predefined, the speech-acts can be *bid*, *accept*, *reject* and so on;.

$Message = TP \times O \times V \times SA$ is the set of messages sent by negotiating agent between themselves. Using *Message* in negotiation thread, negotiating agent can express its argument in conflict, and sent revised messages to other agents. An integrated set of messages is a Cartesian product of topics set (*TP*), ontology set (*O*), variables set (*V*) and speech-acts set (*SA*). A message is a four-element form (tp, o, v, α), which means that negotiating agent performs speech-act α to act on the negotiation topic *tp*, the result is giving ontology *o* a value *v*. This is theoretical form of message, and its final expression is a character string expressed by a certain agent communication language.

$T = \{t_1, t_2, \cdots, t_i, \cdots\}$ is the set of time, t_i represents a time point, and prescribe that if $i < j$ then $t_i \prec t_j$. The reason for describing time as a set of discrete time point but not a continuous function is that every negotiation state takes place at a certain time point.

$$Thread_{ijk} = (o_k(tp_k, (message(i, j, t_1), \\ message(j, i, t_2), message(i, j, t_3)\cdots)))$$ is the set of all negotiating threads; where $i, j \in A$, $o_k \in O$, $tp_k \in TP$, $t_k \in T$. $Thread_{ijk}$ is the negotiating thread between $Agent_i$ and $Agent_j$ on $Topic_k$; the end of a thread means the completion of negotiation between the two agents. $message(i, j, t_k)$ is another expression for message, which means that Agent$_i$ sends the message to Agent$_j$ at time point t_k.

A timer begins to work and the system performs corresponding negotiation thread when the first successful negotiation interaction appears. If a negotiation finishes successfully, the agreement is recorded, while the unfinished negotiations continue. If there aren't any negotiating processes before the time interval ends, it means all the negotiation thread have finished; if there is some unfinished threads when the time interval ends, then terminate the threads compulsorily. If all work is done, the system finishes its running, or enters the next running.

B. Negotiation Process

The concept of Negotiation Environment defines the automated negotiation from macroscopical viewpoint. The following concept, Negotiation Process and Negotiating Agent, will describe microcosmic aspects of the automated negotiation.

For convenience, assume the process of negotiation will finally terminate. Then, it is in any of a finite set of discrete, instantaneous negotiation states. The rationality can be guaranteed because any continuous negotiation process can be modeled as a sequence made up of discrete state with any attainable precision. Here, a state denotes a bargaining. The formal definition is as follows.

Definition 2 (Negotiation Process): a negotiation process is a form

$$NP =< GS, S_i, LS, trans, gen, gs_0, ls_0 > \tag{2}$$

Where

$GS = \{gs_0, gs_1, \cdots, gs_n\}$ is the set of global negotiation states, which is a holistic description for negotiation process. In auction, for example, the highest bid price in every round is a kind of global state.

$S_i = \{s_{i0}, s_{i1}, \cdots, s_{ij}, \cdots, s_{in}\}$ is the set of negotiating agent's instant negotiation state. Here, *i* is a negotiating agent's code name; s$_{ij}$ represents the state, at which Agent *i* is, when it is in round *j*. In auction, for example, an agent's bid price in a round is its instant negotiation state.

$$LS = \{\{s_{10}, s_{20}, \cdots, s_{n0}\}, \{s_{11}, s_{21}, \cdots, s_{n1}\}, \cdots, \\ \{s_{1j}, s_{2j}, \cdots, s_{nj}\}, \cdots, \{s_{1m}, s_{2m}, \cdots, s_{nm}\}\}$$ is the set

of local negotiation states, which is composed of sets of every negotiating agents' instant negotiation states in every round. In auction again, for example, when agents finish bidding in every round, the bid price of every agent make up a set that can be regarded as an element in the set of local state.

$trans : \wp(LS) \times (\{1,2,\cdots,n\} \times Message)^n \to LS$ is local state transformer function, which represents behavior of the negotiation. It describes the process that the local negotiation states transform from one to another when every negotiating agent makes decision and sends message to other agents. The definition of the function indicates two things. One is that negotiation state is *history* dependent, that is, the next negotiation state is depending not only on the current local negotiation state, but also partly on the system's previous local states (here, $\wp(LS)$ is power set of *LS*, the same to the followings). Another is *non-deterministic;* the definition permits the uncertain negotiation states. If $trans(\cdots) = \varnothing$, then there are no possible successor local states. In this case, we say that the system has *ended* its run, and prescribe all negotiations will terminate for convenience of discussion. The function indicates an agent makes a decision about what speech-act to perform based on the history of the system that it has witnessed to date. Note that the agent is supposed to be certain, though, negotiation state is potentially uncertain. The following equation is a further explanation for the function, it can help us understand the function more clearly

$$(\{1,2,\cdots,n\} \times Message)^n =$$

$$\begin{pmatrix} (1,message_1), & (1,message_2), & \cdots, & (1,message_m), \\ (2,message_1), & (2,message_2), & \cdots, & (2,message_m), \\ \cdots, & \cdots, & \cdots, & \cdots, \\ (n,message_1), & (n,message_2), & \cdots, & (n,message_m), \end{pmatrix}^n$$

$gen : GS \times LS \to GS$ is global negotiating state generation function, which describes the process that the system generates current global negotiation state according to last global state and current local state. Generally, the global state in automated negotiation system refers to holistic situations of negotiation process, which should be broadcast to every negotiating agent. In auction, for example, the highest bid price of current stage announced by the system can be regarded as current global state, and its generation process can be described visually by this function.

$gs_0 \in GS$ is system's initial global negotiation state.

$ls_0 \in LS$ is system's initial local negotiation state, and obviously, we can know $ls_0 = \{s_{10}, s_{20}, \cdots, s_{n0}\}$

C. Negotiating Agent

Construction of the Negotiating Agent's concept is based on the *belief-desire-intention* (BDI) model of rational agency. The BDI model gets its name from the fact that it recognizes the primacy of beliefs, desires, and intentions in rational action. Here, we use *goal* as a substitute for *desire*. When the negotiating agent's concept is constructed formally, we get its abstract architecture at the same time. The formal definition is as follows.

Definition 3 (Negotiating Agent): a negotiating agent is a form

$$Agent = < B, G, I, NS, listen, choose, filter, plan, ans, react > \quad (3)$$

Where

$B = \{b_0, b_1, \cdots, b_k\}$ is the set of negotiating agent's believes, which correspond to information the agent has about the negotiation, including domain knowledge, environment parameters, the opponents' believes and so on.

$G = \{g_0, g_1, \cdots, g_m\}$ is the set of negotiation goals, which represents states of negotiation that the negotiating agent would, in an ideal world, wish to be brought about.

$I = \{i_0, i_1, \cdots, i_n\}$ is the set of negotiation intentions, which represents goals that negotiating agent has committed to achieving.

$NS = \{ns_1, \cdots, ns_i, \cdots, ns_n\}$ is the set of negotiation strategies performed by negotiating agent. Where, $ns_i = < pre_i, body_i, post_i >$ is a triple form, which represents that a strategy is composed of pre-condition (*pre*), strategy's body (*body*) and rational result (*post*). In implemented negotiating agents, pre- and post-conditions are often represented as atoms of first-order logic, and beliefs and intentions as ground atoms of first-order logic. Finding a strategy to achieve an intention then reduces to finding a strategy whose pre-condition unifies with the agent's beliefs, and whose post-condition unifies with the intention.

$listen : \wp(B) \times WS \times PS \to \wp(B)$ is the listening function, which represents the negotiating agent determines a new set of beliefs on the basis of the current beliefs, global state and local state. This function is updating belief function, too.

$choose : \wp(B) \times \wp(I) \to \wp(G)$ is the choosing negotiation goals function, which takes sets of beliefs and intentions and returns a set of goals, which means that the goal is generated from beliefs, and implemented negotiating agent requires the goals be consistent with former intentions.

$filter : \wp(B) \times \wp(G) \times \wp(I) \to \wp(I)$ is the filtering negotiation intention function, which takes sets of beliefs, goals and intentions and returns a set of intentions selected by the agent to achieve, on the basis of its former beliefs, goals and intentions. This function describes the negotiating agent's deliberation process.

$plan : \wp(B) \times \wp(G) \times \wp(I) \to NS$ is the planning negotiation strategy function, which on the basis of an agent's current beliefs, goals and intentions, determines a negotiation strategy to achieve the intention.

$ans : \wp(I) \times NS \to Message$ is the answer function, which represents negotiating agent performs speech-act planning to generate message, according to current negotiation intentions and all the available negotiation strategies.

$react: WS \times PS \rightarrow Message$ is the reaction function mapping the Cartesian product of the set *WS* and *PS* to the set of *Message*. In order to promote the reactive ability of negotiating agent, in some certain circumstance, negotiating agent can response directly to changes from outer environment through this function without implementing the above deliberation process.

D. Run of Automated Negotiation System

Now let's consider the behavior of the whole system. Let the global negotiation state starts from gs_0, and local negotiation state starts from $ps_0 = \{s_{10}, s_{20}, \cdots, s_{n0}\}$ .Let: B_0, G_0 and I_0 be the initial set of beliefs, goals and intentions, respectively. The set of available messages is

$$Message = \{message_1, message_2, \cdots, message_m\} \quad (4)$$

First, negotiating agent synchronizes with the negotiation process by listening to the global and local negotiation state to update its belief set through the *listen* function, and get a new belief set B_1, so we have

$$\|\wp(B_1)\| = listen(\|\wp(B_0)\|, ws_0, ps_0) \quad (5)$$

where, $\|\wp(B_1)\|$ represents an element in the power set of B_1, the same to followings.

Then, the current set of goals is updated to G_1 through the *choose* function.

$$\|\wp(G_1)\| = choose(\|\wp(B_1)\|, \|\wp(I_0)\|) \quad (6)$$

Then, we have the set of current negotiation intention I_1 through the *filter* function.

$$\|\wp(I_1)\| = filter(\|\wp(B_1)\|, \|\wp(G_1)\|, \|\wp(I_0)\|) \quad (7)$$

According to the current beliefs, goals and intentions, we can get the negotiation strategy which should be planed to use in the circumstance.

$$ns_1 = choose(\|\wp(B_1)\|, \|\wp(G_1)\|, \|\wp(I_1)\|) \quad (8)$$

Finally, through speech-act planning, negotiating agent transforms the results, which have been got from the above process of deliberation, to a *message*, and sends the *message* to other agents to finish an interaction. The process can be expressed as

$$message_i = ans(\|\wp(I_1)\|, ns_1) \quad (9)$$

As mentioned above, negotiating agent can also response directly to changes from outer environment by realizing the *react* function as follows

$$message = react(gs, ls) \quad (10)$$

When all the negotiating agents in the system have finished the above processes, every negotiating agent selects the Speech-Acts and forms a message to act on the negotiation state. The result of the action is some local states that the negotiation can reach. However, just one local state, which is not known by the agent in advance, can be realized. Then, the agent continues to implement another action, and the negotiation gets to another state in the set of possible states. Here, the system's local negotiation state transforms to the next state ls_1.

$$\begin{aligned} ls_1 = trans(\|ls_0\|, ((1, message_k), \cdots, \\ (i, message_j), \cdots, (n, message_l))) \end{aligned} \quad (11)$$

where, $i, j, k, l \in N$, $1 \le i \le n$ and $1 \le j, k, l \le m$.

At this time, the global negotiation state changes into gs_1

$$gs_1 = gen(gs_0, ls_1) \quad (12)$$

Then, the whole negotiation process goes on to the next loop. Now, we can get any global state gs_n and local state ls_n. Formally,

For $n > 0$

$$\begin{aligned} ls_n = trans(\|ls_{n-1}\|, ((1, message_k), \cdots, \\ (i, message_j), \cdots, (n, message_l))) \end{aligned} \quad (13)$$

$$gs_n = gen(gs_{n-1}, ls_n) \quad (14)$$

The process mentioned above continues until the negotiation is over.

An automated negotiation system *ANS* is a triple form containing negotiation environment, negotiation process and negotiating agent, which can be expressed formally as follows.

Definition 4 (Automated Negotiation System): automated negotiation system is a triple form

$$ANS = < NE, NP, Agent > \quad (15)$$

Where

NE is the negotiation environment;

NP is the negotiation process;

Agent is the set of all negotiating agent participating in the negotiation.

Definition 5 (Run): a run of system $ANS = < NE, NP, Agent >$ is a sequence of interleaved negotiation's global states and local states, which can be defined formally as

$$run: ls_0, gs_0 \xrightarrow{ls_1} gs_1 \xrightarrow{ls_2} \cdots \xrightarrow{ls_n} gs_n \quad (16)$$

here, the definition implies that whether an agreement is reached or not, the negotiation will terminate finally.

Any system will have a set of possible runs associated with it; we denote the set of runs of a system *ANS* by $R(ANS)$. We assume $R(ANS)$ contains only *terminated* runs.

III. Negotiating Agent Architecture

Negotiating Agent Architecture (NAA) is designed for describing internal structure of negotiating agent. The theoretical foundation of the architecture comes from the negotiating agent's concept model, which have been defined above as an abstract architecture.

Recent work has classified agent architecture into reactive system, real-time reasoning system and hybrid system. The hybrid agent has features in common with both reactive agent and real-time reasoning agent. NAA is a kind of hybrid agent architecture (Figure 1). That means NAA has features coming from both BDI deliberative agent, which is based on goal directed reasoning, and reactive agent, which is controlled by reactive behavior.

There has been some work in the design of agent architecture that attempts to integrate goal directed reasoning and reactive behavior. For example, the PRS interacts with the environment through four mechanisms: *sensors* and *monitor* (which are in charge of perception from the environment), *effectors* and *command generator* (which are used for acting on the environment) . However, it is a common architecture for solving general problems in the field of reasoning systems and seems to be too complex for the applications of negotiation on-line. Since the negotiation is a kind of linguistic mechanism and a language generator for interaction with other agents. They are form, the NAA, different to traditional agent architecture, just needs a communication NAA's *communicator* and *speech-act planner*. In addition, NAA has a *reactive filter* for the purpose of increasing the system's capacity for reactivity.

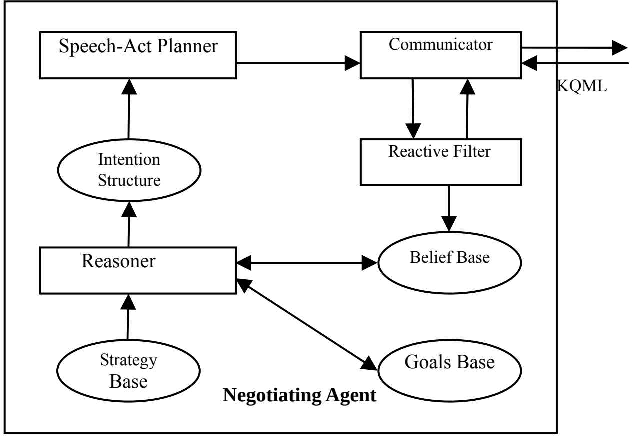

Figure I. Negotiating Agent Architecture (NAA)

Belief Base is a container for the current beliefs of the agent, which realizes the set of beliefs in the negotiating agent's concept model. Typically, beliefs include facts about static properties of the negotiation application domain, and facts acquired when the agent executes its reasoning. The knowledge contained in the belief base is represented in first-order predicate calculus.

Goals Base realizes the set of goals in the concept model. Goals in the base are expressed as conditions over some interval of time, and are described by applying various temporal operators to state descriptions. This allows representation of a wide variety of goals, including goals for achieving maximum price, goals for shorter bargaining time and so on. A given speech-action or sequence of speech-actions, is said to be appropriate for achieving a given goal, if its theoretical execution results satisfy the goal description.

Strategy Base is a kind of knowledge about how to accomplish given goals or react to certain bids from other agents, and is presented by declarative procedure specifications. This base realizes the set of negotiation strategy in the concept model. Each strategy consists of a *body*, which describes the algorism of the strategy, and a *condition* that specifies under what situations the strategy is applicable. Together, the condition and body express a declarative fact about the results and utility of performing certain negotiation strategies under certain conditions.

Intention structure realizes the set of intentions in the concept model. It is a data structure organizing all those goals that the agent has chosen for execution, either immediately or at some later time. These adopted goals are called intentions. The set of intentions comprising the intention structure form a partial ordering. An intention earlier in the ordering must be either realized or dropped (and thus disappear from the intention structure) before intentions appearing later in the ordering can be executed.

Reasoner runs the entire system, which realizes the *listen* function, *choose* function, *filter* function and *plan* function in the concept model. From a conceptual standpoint, it operates in a relatively simple way. At any particular time, when certain goals are active in the system and certain beliefs are held in the belief base, then selected goals will be placed on the intention structure, and then a subset of strategies in the system will be invoked. Finally, one or more of these applicable strategies will then be chosen for execution.

Speech-Act Planner is a language generator, which realizes the *ans* function in the concept model. The agent must select what it should say based on the relevance of the speech-act's expected outcome or rational effect of its goals. *Speech-Act Planner* can select appropriate *performatives* and form KQML or FIPA ACL messages according to the prospective rational effect of a certain intention. In other words, with the aid of the *speech-act planner*, the communicator has something to say.

Communicator is in charge of the agent's interaction with the environment, including other agents. It has the ability to process Agent Communication Language (ACL). It receives KQML messages about negotiation from the environment, and then parses them to get useful information for the agent to process. Finally, it sends KQML messages back to the environment.

Reactive Filter realizes the *react* function in the concept model. It is a reactive mechanism, whose purpose is to provide agent with fast, reactive capabilities for coping with events that are unnecessary or difficult for the reasoning mechanism to process. A typical event, for example, would be the wrong KQML message received by the *communicator*. The *reactive filter* provides the agent with a series of *situation-reaction rules* for processing wrong messages, and for preventing other unpredictable situations. When a given rule is activated, an appropriate action is sent to the agent's communicator, which will send a responsive KQML message very quickly and directly to the environment. So, this mechanism guarantees a certain degree of reactivity.

IV. NEGOTIATION REASONING ALGORITHM

We have designed a negotiating agent architecture, in which has a reasoning machine to control and coordinate the reasoning between negotiation belief, desire and intention. The following algorithm for controlling the reasoning is non-formally expressed in 8 steps:

Step 1: Initialize belief base *B*, negotiation desire base *D*, and negotiation intention structure *I*

Step 2: Negotiating agent monitors changes in the status of negotiation (such as receiving bargaining information sent by other agents), and adds the informed events in the interactive triggered belief base *B_T*.

Step 3: Update belief base *B* using current interactive triggered belief *B_T* and run-time belief *B_D*, *B_I*

Step 4: Apply varieties of negotiation strategy to generate negotiation options, all options will be submitted to the desire base *D* to generate a new negotiation desire. That is, every option is a negotiation desire.

Step 5: Combining with a utility model to calculate utility of all the current options in the desire base; according to the rule NR2, select the option which has the largest utility to be current intention of this round of negotiation. Then it is submitted to intention structure *I*. Here *I* was a queue.

Step 6: Determine whether the conditions for implementing current intention exist. If they do not (such as network interruption), then withdrawn from circulation, clear the intention queue; if they do, then implement the first intention in the queue.

Step 7: Carry out Speech Act planning; select the appropriate speech act verb to express the content of the negotiation intention, complete the interaction with other agent.

Step 8: According to the reasoning rules NR3 and NR4, the data of the negotiation desire and intention generated in the current round of negotiation are dumped to the agent's belief base. This step updates the *B_D* (runtime belief about desire) and *B_I* (runtime belief about intention). Then cycle returns to Step 1.

From the above description we can see that the desire base plays a unique role in the agent's negotiation reasoning process. In fact, it is a temporary mechanism for data storage, just like a run-time memory, because the system doesn't retain the desire data's persistent state. Its main task is responsible for choosing negotiation intentions. Therefore, it is a logic link between the negotiating agent's rational reasoning and decision making.

```
BDI Reasoner Run
Begin
  B := $B_0$;     //Initialize Agent's belief
  D := $D_0$;   //Initialize Agent's negotiation desire
  I := $I_0$;     //Initialize Agent's negotiation intention
   while true do
     B_T := Listen();    //Monitor the negotiation's status,
                         //updates interactive triggered belief
     B := updateB(B_T, B_D, B_I);    // Updates belief base
     D := generateD(B, I);   //Generates negotiation desire
     I := generateI(B, D, I); //Generates negotiation intention
     α := plan(B, I, SA);    //Execute speech act planning,
                             //choose speech act from SA
     if(not(Impossible(I, B))) then    //If the conditions for
                                       //executing intention exit
        execute(α);    //Execute negotiation speech act
        B_D := D; //Update runtime belief about negotiation desire
        B_I := I; //Update runtime belief about negotiation intention
     else
        quite
     end if
   end while
End
```

Figure II. Negotiation reasoning algorithm

In addition, negotiating agent uses negotiation strategy in a different way. In general, the usual negotiation systems pre-define negotiation strategy which can be implemented by agents, such as the Kasbah. Agent in the negotiation process cannot decide on their own which negotiation strategy should be used. That is unreasonable. In order to obtain maximum benefit, we should give agents autonomous ability to choose the most appropriate negotiation strategy to implement, according to the situation of negotiation at that time. The negotiating agent designed in this paper has such characteristics. It can take full advantage of high speed computing ability of computer, and respond to the current negotiation with every strategy it has, and then decide to adopt which strategy should be selected, by comparing the utility of the results calculated from the strategies. To do so, negotiating agent is given maximum autonomy for negotiation decision making.

Figure 3 is the formal description for negotiation reasoning algorithm.

V. CASE STUDY

The simplest negotiation model is bilateral negotiation with single attribute. But in most cases, the negotiators have to process several attributes of the product at the same time. The following case is a bilateral negotiation with multi attributes. The case is from INSPIRE system (InterNeg Support Program for Intercultural REsearch) developed by InterNeg research group Carleton University [10]. The core content of the case is described as follows:

A simple negotiation has been set up with the objective of trying to secure a contract between two companies, Rosa Inc. and Casa Ltd. Rosa wants to sell an aircraft which Casa is considering purchasing. Two agents, Misty and Smiley, negotiate for Rosa and Casa. "Misty" negotiates on behalf of Rosa Inc. and "Smiley" represents Casa Ltd. Both Misty and Smiley have carefully read the information about their respective organizations to understand the problem and its issues. There are only two issues in this simple negotiation: the *price* of the aircraft and the terms of the *warranty*. It has been established that the normal price of this aircraft is in the range of $300 000 to $320 000. The sensible increase is of $10 000. Thus, the price *options* are $300 000, $310 000, and $320 000. In this industry there are four types of warranty typically available. The options are: no warranty, a 6 month, one year, and a 2 year warranty. Both negotiators analyze the two issues and their associated options in terms of their relevance to their respective organizations and move to the pre-negotiation phase.

A. Preparation

In real business negotiations, the issues are always determined in advance, including quantity, price, delivery time, and so on. The negotiating parties have different preferences to the issues. The *weight* indicates the importance of the issues to the negotiator. The results of Misty's and Smiley's preference are shown in Table 1.

In order to measure the merits of the negotiation proposal, it is needed to calculate the value of the current proposal's utility. Utility function is given below

$$u_i = \sum_{j=1}^{n} w_j y_i(x_j)$$

$y_i(x_j)$ is pretreatment to the dada of negotiation attribute.

If attribute *j* of proposal *i* is about benefit, then

$$y_i(x_j) = \frac{x_j - x_j^{\min}}{x_j^{\max} - x_j^{\min}}$$

If attribute *j* of proposal *i* is about cost, then

$$y_i(x_j) = \frac{x_j^{\max} - x_j}{x_j^{\max} - x_j^{\min}}$$

Where, x_j is the value of current attribute *j*, $x_j^{\max}$ and $x_j^{\min}$ are the maximum and minimum value of the attribute.

The value of u_i indicates whether the offer or counteroffer is good or bad. In this study, it is assumed that there is a rational agent, who seeks to maximize self-interest. Then, in most circumstances, it would choose the offer or counteroffer with the largest utility.

TABLE I
NEGOTIATING AGENT'S PREFERENCE

Misty's		Smiley's	
Negotiation Attributes	Weight	Negotiation Attributes	Weight
Price	0.7	Price	0.5
Warranty	0.3	Warranty	0.5

B. Negotiation Process

Since each negotiator can access the system and make a proposal independently, we will follow Misty's side of the negotiation.

TABLE II
NEGOTIATING AGENT MISTY'S INITIAL BELIEF SET

Static Belief	Initial Belief Set by Users (Object, Attribute, Value)	Interactive Triggered Belief	Run Time Belief
Decision Making Model M Negotiation Strategy S1 Negotiation Strategy S2	T1: aircraft:=price&wrranty E1: (aircraft, max_price, 320,000$) E2: (aircraft, min_price, 300,000$) E3: (aircraft, warranty, 0 months) E4: (aircraft, warranty, 6 months) E5: (aircraft, warranty, 12 months) E6: (aircraft, warranty, 18 months) E7: (aircraft, warranty, 24 months) E8: (price, weight, 0.7) E9: (warranty, weight, 0.3)	Null	Null

Before we simulate the whole negotiation process, let's first look at Misty's initial state. It is not difficult to understand that Misty's desire base is empty at the beginning, and the intention structure is an empty queue. The initial belief is shown in Table 2.

Let us assume that Smiley first bid $ 300,000 and 24 months of warranty, that is a value pair (30, 24). Misty receives this information, and adds it to the interactive triggered belief base, then calls two negotiation strategy algorithms to calculate a counteroffer to Smiley. The results are shown in Table 3.

TABLE III
MISTY'S 1ST DECISION RESULT

Options	Negotiation Strategy	Negotiation Options (x_1, x_2)	Data Preprocessing Results $(y_1(x_1), y_2(x_2))$	Utility u_i
1	S1	(32, 6)	(1.00, 0.75)	0.925
2	S2	(31.5, 12)	(0.75, 0.50)	0.675

Negotiating agent Misty adds results calculated by two negotiation strategies to the desire base. At this time, Misty's desire base contains two sets of data of negotiation desire: (32, 6) and (31.5, 12). Because the utility of option 1 is greater than option 2, according to the reasoning rule NR2, Option 1 will be selected as an new negotiation intention to be implemented, and will be added to the queue of intention awaiting processing. At this point, there is intention (32, 6) waiting for processing in the queue of intention. Subsequently, according to the reasoning rule R3 and R4, the above data of negotiation desire and intention will be dumped to run-time belief base as historical data, which can be used as system log for later explanation. The course runs circularly until the two agents agree with each other. The whole simulation process is shown in Table 4

TABLE IV
ALL RESULTS OF THE NEGOTIATION PROCESS

Smiley's proposal	Plan to be selected	Negotiation Strategy	(x_1, x_2)	$(y_1(x_1), y_2(x_2))$	u_i	Misty's proposal
(30, 24)	*11	*S1	(32, 6)	(1.00, 0.75)	0.925	(32, 6)
	12	S2	(31.5, 12)	(0.75, 0.50)	0.675	
(30.5, 18)	*21	*S1	(31.7, 12)	(0.85, 0.50)	0.745	(31.7, 12)
	22	S2	(31.5, 12)	(0.75, 0.50)	0.675	
(30.7, 18)	31	S1	(31.3, 12)	(0.65, 0.50)	0.595	(31.5, 12)
	*32	*S2	(31.5, 12)	(0.75, 0.50)	0.675	
(31, 18)	41	S1	(31, 18)	(0.50, 0.25)	0.425	(31.2, 12)
	*42	*S2	(31.2, 12)	(0.60, 0.50)	0.570	
(31, 12)	51	S1	(30.5, 12)	(0.25, 0.50)	0.325	ACCEPT
	52	S2	(31.2, 18)	(0.60, 0.25)	0.495	

At the end of the negotiation process, as expressed in Table 4, the utility of Smiley's proposals is 0.5, which is better than the two proposal made by Misty. Therefore, as a rational agent, Misty logically ACCEPT Smiley's proposal to terminate the negotiation.

In the whole process of the negotiation, we can see that negotiating agent Misty does not simply use the strategy S1 or S2, but select one that can generate better result to use, according to the actual situation. That makes the negotiating agent's ability of autonomous decision-making increased significantly. Moreover, to do so, other agents cannot easily grasp his bidding strategy, thereby enhancing the concealment of decision making.

The internal states of negotiating agent change constantly during the process of negotiation, the following Table 5, Table 6 and Figure 4, respectively lists the final state situation of Misty's belief base, desire base and intention structure.

TABLE V
FINAL STATE OF MISTY'S BELIEF SET

Static Belief	Initial Belief Set by Users	Interactive Triggered Belief	Runtime Belief (not including self-know data of negotiation desires)
M S1 S2	T1, E1 E2, E3 E4, E5 E6, E7 E8, E9	((price, 30), (warranty,24)) ((price, 30.5), (warranty,18)) ((price, 30.7), (warranty, 18)) ((price, 31), (warranty, 18) ((price, 31), (warranty, 12))	((price, 32), (warranty,6)) ((price, 31.7), (warranty,12)) ((price, 31.5),(warranty, 12)) ((price, 31.2), (warranty, 12) ((price, 31), (warranty, 12))

TABLE VI
FINAL STATE OF MISTY'S DESIRE SET

Round	Strategy S_1	Strategy S_2
1	((price, 32), (warranty, 6))	((price, 31.5), (warranty, 12))
2	((price, 31.7), (warranty, 12))	((price, 31.5), (warranty, 12))
3	((price, 31.3), (warranty, 12))	((price, 31.5), (warranty, 12))
4	((price, 31), (warranty, 18))	((price, 31.2), (warranty, 12))
5	((price, 30.5), (warranty, 12))	((price, 31.2), (warranty, 18))

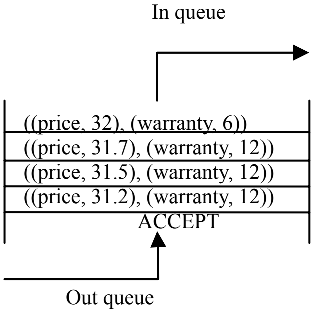
In queue
((price, 32), (warranty, 6))
((price, 31.7), (warranty, 12))
((price, 31.5), (warranty, 12))
((price, 31.2), (warranty, 12))
ACCEPT
Out queue

Figure III. Running state of Misty's negotiating intention queue

VI. CONCLUSIONS

In this paper, the decision making model defined is explicit and formal specifications for building the negotiating agents in an E-commerce environment. The novelty of the model is twofold. In fact it is synthesis work in both agent architecture theory and automated negotiation theory, which are important areas of e-commerce research. More importantly, the concept model and NAA build the foundation for developing an automated negotiation system. However, there are a number of issues needed to be further investigated. For example, we have disregarded in the paper but which we are planning to investigate is the theoretical foundation for negotiation reasoning of the negotiating agent. So far, integrating the BDI theory and utility theory seems to be a good choice.

ACKNOWLEDGEMENTS

This work is supported by grant 2009J05162 of the China Fujian Province Natural Science Foundation for Young Researchers, and by grant 70902042, 70971112 of the National Natural Science Foundation of China.

REFERENCES

[1] C. Bartolini, C. Preist and N. R. Jennings, A software framework for automated negotiation. Software Engineering for Multi-Agent Systems III: Research Issues and Practical Applications, Springer Verlag 2005, LNCS 3390: 213-235

[2] R. Lomuscio, M. Wooldridge and N. R. Jennings, "A classification scheme for negotiation in electronic commerce", Int Journal of Group Decision and Negotiation 2003, 12 (1): 31-56.

[3] V. Tamma, M. Wooldridge, and I. Dickinson. An Ontology-based Approach to Automated Negotiation. In Proceedings of the Fourth International Workshop on Agent-Mediated Elctronic Commerce (AMEC-2002), Bologna, Italy, July 2002.

[4] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra and M. Wooldridge. Automated negotiation: prospects, methods and challenges. Int. J. of Group Decision and Negotiation, 2001, 10(2): 199-215.

[5] Resinas M, Fernandez P, Corchuelo R.. A conceptual framework for automated negotiation systems. Proceedings of Intelligent Data Engineering And Automated Learning - Ideal 2006, 4224:1250-1258

[6] Kwei-Jay Lin. E-Commerce Technology Back to a Prominent Future. IEEE Internet Computing, 2008, 12:60-65

[7] Azzurra Ragone, Tommaso Di Noia 2008. Logic-based automated multi-issue bilateral negotiation in peer-to-peer e-marketplaces. Autonomous Agent and Multi-Agent System,2008,16:249–270

[8] Kiam Tian Seow, Kwang Mong Sim 2008. Collaborative assignment using belief-desire-intention agent modeling and negotiation with speedup strategies. Information Sciences, 2008,178:1110–1132

[9] Jiang WJ, Xu YS 2006. A novel multi-agent automated negotiation model based on associated intent. Agent Computing And Multi-Agent Systems, Book Series: Lecture Notes In Artificial Intelligence, 2006, 4088: 608-613

[10] Using Inspire: An Example. http://interneg.carleton.ca/inspire/doc/demo.html. June, 1st , 2009

Cao Mukun was born 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 system, electronic commerce, agent technology and multi-agent systems.