

# KNOWLEDGE-EMPOWERED AGENT NEGOTIATION FOR ELECTRONIC COMMERCE WITH KNOWLEDGE BEADS

Zhuang Yan, Simon Fong  
Faculty of Science and Technology  
Universidade de Macau  
Macao  
{syz, ccfong}@umac.mo

Shi Meilin  
Department of Computer Science  
Tsinghua University  
Beijing, China  
shiml@tsinghua.edu.cn

## Abstract

*To automate the agent negotiation process, it has been widely accepted that two important tasks must be done: (1) formalize the negotiation process; and (2) incorporate necessary negotiation knowledge and intelligence. However, due to the fuzzy and complex nature, the lack of knowledge interoperability and knowledge-reuse in most agent-based services deployed has posed certain drawback to automated negotiation. An object-oriented ontology-based building block for knowledge representation, namely, Knowledge Bead (KB), has been proposed in [1]. It was designed as a foundation to enable automated agent negotiation in e-trading environment in a systematic way. This paper features the research work of KB on constructing its taxonomy and methodology, and the use of the knowledge carried by KB's in automated negotiation process. We anticipate the more efficient electronic commerce achieved with the support of the knowledge-empowered agent negotiation.*

## 1. INTRODUCTION

To tackle with the negotiation problems for developing efficient agent trading techniques for e-commerce, it is believed to be necessary in associating it with information discovery and ontology issues [3]. By imposing certain structures, rules, or conventions on the interaction between the agents, negotiation can be made easier. This suggests a sophisticated negotiation system is needed, which can do business assessment, coalition formation, criteria evaluation and knowledge management. It would embrace the full automation of online trading agents that are capable to cope with the rapid changes experienced by various business-to-business (B2B) commercial activities. Specifically, for e-commerce scenario, the enormous volume of information describing the products for different traders on the Internet imposes difficulties on finding and matching the appropriate information for the final deal. Moreover, a common ontology for both buyer and seller agents should

be built up during or even before the negotiation. Thus an integrated solution is required for solving all these problems. We proposed our integrated solution based on the definition of Knowledge Bead (KB) [1], which provides an object-oriented way to specify the knowledge in agent negotiation for B2B e-commerce. In addition to the inheritance and hierarchy features of object-oriented modeling, each KB when being used as a leaf in the tree that represents a product specification carries a set of attributes, together with weight, criteria, and rules for describing about every attribute. This makes it possible to represent the various forms of knowledge, including the specification of products, user preferences, negotiation strategies, constraints and the desired final deals in a flexible and efficient way. As virtually everything in a negotiation process is specified using KB's, a common ontology is formed by KB's to facilitate the same points of reference for the negotiation parties. For the same reason, information discovery and knowledge reuse in the online trading environment now becomes standard operations on the defined KB's.

There has been quite a lot of research work on knowledge management in agent system over these years. In particular, the main effort has been put on the knowledge representation and ontology implementation [4]. Different agent's knowledge models have been built up for e-commerce users and automated negotiations [5] [6]. However, besides user-specified preference, the various forms of business intelligence are also important for decision support for actual-world procurement scenarios [11]. Among most the existing negotiation systems, the details on how to extract, integrate, and utilize the knowledge have remained vague.

To manage business data in a variety of forms and turn it into effective knowledge, there are three steps to follow: 1) identify different forms of knowledge; 2) construct methodology for manipulating the knowledge; and 3) integrate knowledge into agent negotiation process. KB is the fundamental building block of the knowledge concerned in e-trading system. The first step is to formulate two main forms of knowledge, namely, general knowledge and meta-knowledge. The second step defines the necessary methods to manipulate a KB and the knowledge about a KB, namely, the meta-KB. The ultimate resolution is the fusion of knowledge carried by KB's into the negotiation process.

To start an electronic procurement process, a Request of Quote (RFQ) is prepared by the buyer, and sent to a group of suppliers. After receiving the quotes proposed by the suppliers, a preliminary assessment is performed by the buyer to choose a set of qualified sellers for further negotiation. A widely accepted model is to split the negotiation process into several phases, such that the negotiation progresses through pre-negotiation preparation, conduct of the negotiation, and post-negotiation settlement.

This paper is organized into two parts. In the first part, Section 2 introduces the basic concept of KB and its use in representing knowledge in e-trading environment. Two different forms of knowledge contained in KB's

are detailed in section 3. After that, the methodology of KB's is illustrated via an example. The second part of the paper in Section 4 talks about the fusion of KB's in agent negotiation. Section 4.1 presents the different agent roles in knowledge management for automated negotiation. Then the discussion of three negotiation phases is followed.

## **2. KNOWLEDGE REPRESENTATION USING KB**

Throughout the trading process, there is a large amount of data which must be collected and processed. Each trader, i.e. buyer or seller, maintains its own database and knowledge base, and is represented by an agent that has access to the data and knowledge. Every negotiation process includes the requests for quotes (RFQs), quotes received from suppliers, buyer's concern and additional constraints, participating supplier's information, applicable business rules, and the final contract if dealt successfully. The whole process requires knowledge in a variety of forms a good deal.

Knowledge is the information extracted from raw data. In principle, there are two different forms of knowledge: general knowledge, and meta-knowledge. General knowledge provides the specification of different categories of objects in the e-commerce domain, which forms the basis of the knowledge space. As far as e-procurement is concerned, the general knowledge in general includes product templates, information about traders, RFQs, quotes and contracts. Business intelligence used in negotiation process is represented as meta-knowledge, which is the knowledge about knowledge. It includes the appropriate links to traders' preference, constraints and criteria, conventional rules for business negotiation, statistics of similar deals, and past trading records. It specifies different forms of dependency existed among attributes defined in KB's. Moreover, meta-knowledge can also present any affinity related with the current KB to a certain dataset. To specify dependency, meta-knowledge can be associated to attributes in a KB. To refer to other knowledge scope, meta-knowledge can be presented in a meta-KB of the current KB with different tags.

Knowledge bead has been defined in [1] as an encapsulation of definition, behavior, and data. It can be a composite object, or a simple, atomic part object in most cases; each has their own methods and data. Definition means a static unique description; this can be a UPC (Universal Product Code) or a unique index implemented at the ontology databases for referencing this KB. Behavior is described by a set of possible methods and rules manipulating KB's and their attributes. Some typical ones include KB formation, duplication, attributes alteration, pruning and linking to other KB. They are analogous to class functions in object-oriented programming, and can be inherited from base classes. Data consist of the defined attributes contained in the knowledge beads. With each attribute, a weight is a relative priority indicating how important this attribute is in the BOM (Bill of Material), which is

delivered as RFQ to potential sellers. During the continuous research on KB's in the context of agent negotiation, we find it necessary to broaden the representing scope of KB. For example, we can let the buyer define for each attribute a preference rank for evaluating the quote offered by a supplier. We will give more details about ranking in the discussion of negotiation process.

Once we have the ontology-based foundation, namely KB, the next step is to provide the same points of references for communicating agents, i.e. we need to establish the taxonomy of knowledge contained in KB's. The goal of a corporate taxonomy is to provide a list of authorized terms in knowledge management and information seeking [7], as well as the mapping between concepts to connect traders with the right knowledge at the right time. We discuss the details of knowledge taxonomy and methodology of KB's in the following section. In this paper, we consider the data and knowledge in the context of automated negotiation.

### 3. KNOWLEDGE TAXONOMY & KB METHODOLOGY

#### 3.1. General Knowledge

General knowledge deploys data-oriented taxonomy which are implemented as various types of KB templates. Templates are organized based on the product categories or themes, such as on eBay (<http://www.ebay.com>). The product space is represented as a labeled, directed graph with two types of nodes: a leaf KB node and a category KB node, as depicted in Figure 1. A leaf KB inherits attributes and behaviors from its parent category KB, along with new features and operations added. Each KB carries an identity number and a category name. The category name provides the basic domain information about the current KB. It is represented as a sequence of labels corresponding to the edges in the path, e.g.

*/ProductCategories/Electroics&Computers/Cameras&Photo/DigitalCameras*

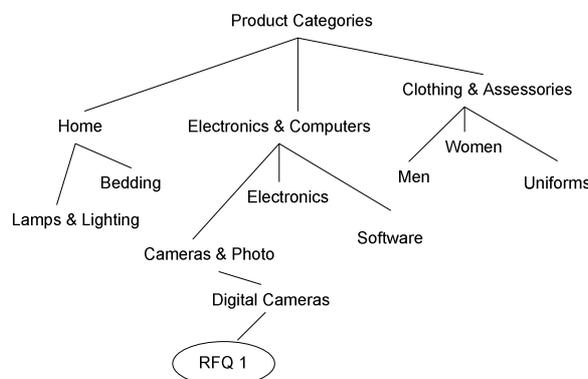


Figure 1. Part of the product space of KB's

To illustrate, we use a RFQ represented in KB as an example. A RFQ is prepared by a buyer to start the e-procurement process. It contains the initial information of the requested products. The example shown in Table 1 depicts a KB filled for purchasing a digital camera. Although there are much more details that should be considered in the real situation when buying a camera, here we just briefly specify some of them for illustration only. We will talk more about how to make use of the KB in automated negotiation later in the paper. In many circumstances a buyer may have its own concern on the requested products which is not written in the RFQ. For example, the buyer may not want the sellers know he/she has the upper limit on the price (so to have a better deal). Instead, this information will be used when the collected quotes are assessed and during further negotiation. Buyer's concern is thus hidden for own use and separated from the RFQ which is made public to all participating sellers. As an extended part of the corresponding RFQ, it is represented using a sub-KB which is associated to the KB defined for the RFQ.

Table 1. A RFQ KB filled for purchasing digital camera (DC)

<b>KB#RFQ_DC_1</b>			
<b>Attribute</b>	<b>Value</b>	<b>Weights</b>	<b>Negotiable</b>
<i>Resolution</i>	>= 340,000	10	NOT
<i>Removable Memory</i>	Yes	10	
<i>Features</i>		7	NOT
<i>Display Size</i>	>=1.5 Inches	3	
<i>Tripod Mount</i>	Not Required	0	
<i>Batteries</i>	Lithium Ion	5	NOT
<i>Size</i>	Pocket Size	8	
<i>Delivery_Date</i>	Before Aug.1	10	
<b>Hidden Part</b>			
<i>Price</i>	<=\$400		<a href="#">Rule#Price_1</a>
<i>Quantity</i>	[1, 5]		
<i>Delivery_Date</i>			<a href="#">Rule#Delivery_1</a>

KB's describing traders and deals also belong to data-oriented taxonomy. They have a different domain space other than the product space, but with the same category hierarchical structure.

### 3.2. Meta-Knowledge

After data-oriented taxonomy has been created for general knowledge, meta-knowledge is to be arranged to assist automated negotiation. Traders represent and obtain the general knowledge through the attributes value

specified in a KB template. For meta-knowledge, the goal is to identify and then make use of the existing dependency and affinity among attributes and among KB objects.

### 3.2.1. Dependency

When the basis of the agent negotiation context is constructed using KB's, meta-knowledge of dependency is about the business trading rules and other constraints specified by the trader on the current KB. We first talk about constraints. There are two types of *constraints*. The fundamental one is either the valid range specified for an individual attribute or an inter-attribute constraint specified for multiple attributes within the same KB. We can refer to this kind of constraint in the specification of a single KB. Another type is a constraint between the current KB object and other related ones. For example, a BOM consists of several RFQs for multiple different product items. A constraint says that the delivery date of two specific items, among all the RFQs, should be the same. The buyer can then define '*delivery\_date*' as an attribute with the constraint in both of the two RFQs. The attribute thus becomes a *pivot* attribute to link between the two RFQs.

Besides constraints, trading rules are also important forms of dependency in meta-knowledge. In e-commerce, business intelligence includes a variety of business conventions and negotiation strategies. It is a natural way to represent these knowledge using *if-then* rules. In principle, an *if-then* rule is represented as an action triggered by a certain condition. Conditions are pre-defined on attributes in a KB template, which are referred as pivot attributes. Once a condition is satisfied on a pivot attribute, a change to an attribute value or even the transition from the current KB to another KB will be triggered, depending on the action specified in the rule. They are present in a KB to reveal the dependency between the current KB and another. One example is: *if the Delivery\_Date in a quote offered by a seller is late but less than 5 days, then the buyer can further ask for a reduced price with 20% off for the requested product*. Following this rule, if the condition is met, it is possible for the corresponding buyer's agent to automatically produce an updated RFQ with the new expected price. For the same RFQ example in Table 1, this rule is included in the hidden part, which is only available for the owner, as shown in the shaded part of the table. The hidden attribute '*Delivery\_Date*' associated with the current RFQ is a pivot attribute. The rule *Rule#Delivery\_1* is specified in the *negotiable* column, it will trigger a new RFQ when the defined condition is satisfied.

### 3.2.2. Affinity

While dependency is about the relationship among attributes, affinity is about the inherent ontology among objects. Meta-knowledge of affinity indi-

cates that a group of objects from different category domains share the same characteristic. It can be present in different forms. For example, a supplier in the category of '*electronics and computers*', also provides lamps which belong to the '*home*' category. To represent the same supplier under different categories, each KB category is assigned a canonical name, which is the primary category name. A KB with a different category name from the canonical name is associated with a pivot attribute storing the canonical category name (e.g. '*electronics and computers*'). The important purpose of using canonical name is to provide the same reference point for different ontology domains to which traders refer. Via the canonical name, KB's originated from different ontology domains can be linked in an effective way.

For traders who need to do e-trading regularly, it will be helpful to set up the trader's profile. Among the rest, the trader's preference is specified in the *preference KB*. One type of the preference is *static*. Once specified, the static preference will be associated to the each RFQs (or, quotes) in a whole BOM if applicable. For example, a buyer may specify in the preference profile his/her regular payment method. A second example is about the quality level the buyer usually requires. There is also other information that can be considered as static preference, e.g. the seasonal markup of the price. Another type of preference is *dynamic*. In a challenging e-trading environment, the traders often find it necessary to change their preference and evaluation criteria in a dynamic way. Especially during negotiation, there is almost always a need for concession before the final deal is reached. The dynamic preference is specified as a pivot attribute in the current KB (RFQ or quote). When the trader decides to adjust the preference during negotiation, the pivot attribute helps to change from the current KB to a new KB with the preference changed and other attributes changed in consequence. The transition is represented using an *if-then* rule defined on the pivot attribute. Rules in detail will be present in the following section about dependency among KB's. We referred the two KB's as KB's based on different concepts. The concept-based alternatives are embedded in RFQ and quote to speed up the negotiation process. It allows greater flexibility in describing the desired product under different concepts. The concept-based automated negotiation model was discussed in detail in [2].

Another form of affinity is to group similar KB's into a dataset for knowledge reuse. It serves to represent various forms of business intelligence. Groups of KB's can be discovered during negotiation and afterwards. Quotes and deals based on the same rule or constraint are grouped together. For example, if the seller is negotiating with a buyer who has a past deal in the seller's trading record. Or, the buyer has a preference profile similar to a previous buyer of the seller. Then the preference and thus the behavior of the buyer will be considered predictable by the seller's trading agent relying on the past records. This is based on the premise that people's preferences are correlated; groups have similar preferences so that the person who needs to

make a choice can instead utilize the choices made by others in the group. Another example is to group all the deals with a price discount. The relevant suppliers will form a favorite supplier list for the buyer. The affinity list can be used as an additional constraint during negotiation process. The buyer agent can give favor to the suppliers in the favorite list for ranking the quotes. It can also ask for more discounts when constructing counter-proposal to the affinitive supplier, depending on the rule specified.

### 3.2.3. Meta-KB

From the foregoing discussion of meta-knowledge about different forms of affinity, it is not difficult to realize that when it is necessary to refer to the affinity, certain external datasets shall be involved. To our knowledge, most current automated negotiation systems lack the ability of specifying the explicit use of knowledge base in a systematic way, so to assist an efficient automatic negotiation process. For this purpose, we define *meta-KB* as a special meta-object for describing the affinity of a KB in concern, and the datasets to which it can refer correspondingly. In this paper, a meta-KB specifies the ontological properties which are necessary in the agent negotiation context. It has the form in the following Table 2. We define different tags for different ontological property. Note that the concept of meta-KB can be applied to a broader scope in the e-commerce domain (e.g. the logistics system). More tags can be defined appropriately to assist the use of meta-knowledge in the corresponding application context.

In this example, the first tag *KB\_ID* indicates the relevant KB's identifier. *CNAME* refers to the canonical name of the KB's category. *PREFERENCE* refers to the profile containing attributes applicable to the current KB. The applicable attribute is specified in the pivot column. *CONCEPT* applies the rule which is used to switch from the current KB to another KB referred with a different concept. *AFFINITY* refers to a dataset containing certain form of business intelligence. The corresponding rule specifies how to make use of the dataset. The detailed discussion is followed in the next sub-section.

Table 2. Meta-KB of the RFQ KB in Table 1

Meta-KB		
Tag	Value	Pivot/Rule
<i>KB_ID</i>	KB#RFQ_DC_1	
<i>CNAME</i>	../Electronics&Computers	
<i>PREFERENCE</i>	Buyer-Profile	PaymentMethod
<i>CONCEPT</i>	KB#RFQ_DC_2	Rule#Concept_1
<i>AFFINITY</i>	Supplier-Group#2	Rule#Supplier_1

### 3.3. Methodology of KB's

We establish knowledge taxonomy because the different forms of knowledge are manipulated in different ways. When a new KB is created, according to the category to which it belongs, it is inserted into the appropriate domain space, such as product, trader space, and etc., respectively. The KB identifier is represented as the category name together with the name of the leaf node in the graph. An identifier can be looked up first following the sequence of labels contained in category name, and then the name of the leaf node. As an example, the KB in Figure 1 has the following identifier:

*/ProductCategories/Electronics&Computers/Cameras&Photo/DigitalCameras/RFQ1*

Meta-knowledge is manipulated to deal with either the imposed dependency or the inherent affinity. To show the methodology of KB's in agent negotiation, we use an example to illustrate. Consider a buyer who wants to buy a digital camera. The information agent provides a RFQ template then with the appropriate attributes filled by the buyer as in Table 1. The negotiating agent gets a list of quotes and performs the preliminary screening process based on the attributes. It then makes use of the meta-KB of the current RFQ as depicted in Table 2. Here we assume that knowledge agent is instructed to make use of the following meta-knowledge: a buyer profile containing the common preference specified by the buyer, a concept-based alternative KB, and an affinity group of KB's which is a list of preferred electronics suppliers.

In Table 2, the pivot attribute '*PaymentMethod*' is indicated as the *PREFERENCE*. We assume that the buyer use cash-on-delivery as the regular payment method. Thus the current RFQ includes 'cash-on-delivery' as its payment method automatically. The *CONCEPT* tag associates to a concept-based alternative KB with a rule '*Rule#Concept\_1*', which says that "if the *Size* is not pocket size, then abandon the current RFQ and switch to *KB#RFQ\_DC\_2*", which is depicted in Table 3. Note that the new RFQ has the attributes value and weight changed in bold. The 'cash-on-delivery' payment method is also added. To handle the *AFFINITY* tag, the negotiating agent searches in the affinity group '*Supplier-Group#2*'. We assume it contains the records of suppliers of final deals with a certain discount recorded. The rule '*Rule#Supplier\_1*' says that 'if it is a preferred supplier in the affinity group, then search the past records for the maximum discount, and ask for this maximum discount on the price proposed by the supplier'. Besides the rule about the '*Delivery\_date*' as described above, the buyer has also included another rule '*Rule#Price\_1*', in the in the sub-KB of RFQ saying that "if the quoted *Price* is higher but no more than 5% of the expected upper limit, then ask for free accessory". The negotiating agent then compares each quoted value with the *Price* specified in the sub-KB of the buyer's concern, which is kept separately by the local information agent. If the condition on price is met, then the negotiating agent will inform the information agent to

prepare a new RFQ for the next bargain. In this way, proposal and counter-proposal are exchanged until the final contract is signed, or otherwise the negotiation is aborted.

Table 3. An alternative RFQ KB

<b>KB#RFQ_DC_2</b>			
<b>Attribute</b>	<b>Value</b>	<b>Weights</b>	<b>Negotiable</b>
<i>Resolution</i>	$\geq 340,000$	10	NOT
<i>Removable Memory</i>	<b>Not Required</b>	2	
<i>Features</i>		7	NOT
<i>Display Size</i>	$\geq 1.5$ Inches	3	
<i>Tripod Mount</i>	Not Required	0	
<i>Batteries</i>	Lithium Ion	5	NOT
<i>Size</i>	<b>Regular</b>	5	
<i>Payment Method</i>	<b>CashOnDelivery</b>	N/A	
<i>Delivery_Date</i>	Before Aug.1	5	
<b>Hidden Part</b>			
<i>Price</i>	<b><math>\leq \\$350</math></b>		<a href="#">Rule#Price_1</a>
<i>Quantity</i>	[1, 5]		
<i>Delivery_Date</i>			<a href="#">Rule#Delivery_1</a>

Notice that it is sometimes necessary not to fully depend on the information agent to automatically produce the updated RFQ with the new attribute added, especially when it is to bargain a better price for a product (after all, money is usually the bottom-line factor in many negotiation processes). For the previous example, the buyer is given the flexibility to specify that he wants a free spare battery, instead of being given any free accessory that is not fit of the buyer. Thus the online buyer is suggested to manually provide information to the information agent during negotiation. The same consideration also applies to rules, i.e. traders can by themselves adjust the rules used by the knowledge agents.

#### 4. AGENT NEGOTIATION

This section details the automation of agent negotiation via KB's. We first talk about the agent architecture required for the fusion of KB's into negotiation process. In the pre-negotiation preparation phase, we discuss the construction of KB's containing knowledge and meta-knowledge which is necessary to describe the negotiation context. We then describe the mathematical algorithms for the assessment of alternative offers provided by the suppliers. Next, during negotiation, the negotiation protocol and strategies are discussed.

Finally, for post-negotiation settlement, we present the knowledge management techniques used to gather knowledge for future reuse.

#### 4.1. Agents in Negotiation Process

In a negotiation scenario using KB's, there are mainly three kinds of agents invoked: information agents, knowledge agents, and negotiating agents. These agents function as suggested by their names accordingly. Information agents are responsible for formulating the elementary knowledge in category-based KB's. They work locally on the trader's machine, process on the elementary information provided by the trader, and provide the general knowledge to both knowledge agents and negotiating agents whenever necessary. Knowledge agents are equipped with advanced knowledge management capabilities. It is their job to form the meta-knowledge as described above. The workspace of a knowledge agent consists of a knowledge base that keeps all the meta-KB's and affinity groups of KB's, and a rule base that maintains the rules on KB's. A knowledge agent serves negotiating agents as a back-end assistant in providing the business intelligence which is needed in a negotiation scenario. It can work on a trader's machine, or, a remote one which is dedicated for the electronic procurement. Given a complete negotiation context with all the information and knowledge specified and intelligence formulated, a negotiating agent is responsible for the following steps: 1) collect a preliminary set of quotes from suppliers' information agents; 2) perform assessment on collected quotes and screen for a negotiable set of quotes; 3) for each quote candidate, negotiate with the supplier's negotiating agent using the negotiable attributes, buyer's concern, additional constraints, together with the business intelligence in various forms; 4) done if a contract can be reached, and record the deal accordingly; abort if failed. Note that in step 3 and 4, the information agent and knowledge agent are also invoked, providing as well as updating the general knowledge and meta-knowledge needed by the negotiating agent. The model is illustrated in Figure. 2.

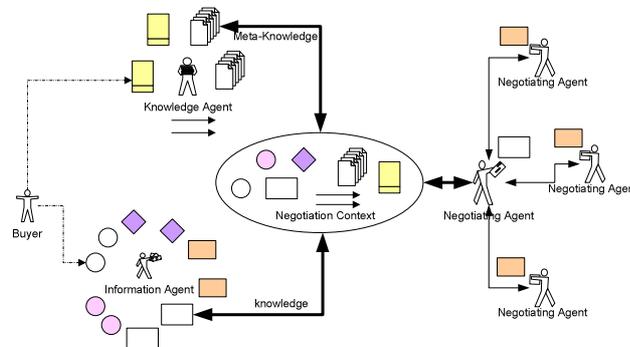


Figure 2. Agents in Knowledge Management for Automated Negotiation

## 4.2. Pre-negotiation Preparation

According to the procurement requirement, the information agent, on behalf of the trader, first establishes the necessary KB templates for a variety of product categories. When it is the time to prepare a RFQ from the buyer side, the buyer can choose to use a predefined KB template from a certain category for the specification on existing or new attributes, or, to build a new KB template on its own. Every filled RFQ will then be saved for quick reference when similar procurement is required in the future time. An example of a KB template filled for purchasing digital camera is given in Table 1.

A successful e-trading paradigm must allow flexible negotiation with strict constraints. In particular, trader's constraints and preferences captured during pre-negotiation preparation play an important role in negotiation automation. Usually this comes along with the specification of product attributes in KB templates, as illustrated in Table 1, and the associated meta-KB in Table 2. Weights in the range of 0 to 10 are assigned to each attribute, where 0 indicates a least important attribute and 10 indicates a most important attribute. Assigning weights for a multi-item and multi-attribute RFQ is a tedious and time-consuming job for the buyer. The KB template allows the buyer to only assign the attribute weights when they find it important to do so. This can be a preferred attribute or a trivial attribute for the buyer. Otherwise, the default weight 5 is assigned. These weights will be used in evaluating the quotes received, which we will discuss in the later subsection. Moreover, there are some dynamic issues to consider when filling in the KB's templates. Two types of attributes can be observed: *explicit attribute* and *implicit attribute*. Explicit attributes are those that buyers can give explicit value in the specification. Implicit attributes are those that buyers give no explicit description. For the digital camera example in Table 1, optical zoom and display size are explicitly specified, while tripod mount is specified as 'Not Required', which means that the buyer has no specific requirement in this attribute. Notice also that tripod mount has the user preference value 0, indicating it's a trivial attribute. Implicit attributes provide different extent of flexibility for the trading agents conducting negotiation. If two offers both satisfy the buyer's preferences, then the one with extra attributes fulfilled maybe considered superior. Attributes that must be satisfied are of the type *NOT Negotiable*. Others are considered negotiable. The purpose of distinguishing non-negotiable attributes from negotiable type is to speedup the procurement process by negotiating with only the negotiable attributes. There are also hidden attributes which are not to be seen by the other side. These are usually considered as different constraints on the KB's attributes set by the buyer. They can be used in either the automatic quote evaluation as a constraint, or agent negotiation via argumentation as a justification, or both, depending on the trader's business policy.

### 4.3. Conduct of the Negotiation using KB's

Negotiation is a strategy-based process governed by some explicit and implicit rules. For different negotiation scenarios, the negotiating agents are able to select appropriate negotiation strategies based on the current negotiation context and the history knowledge. Negotiation context is the set of knowledge and meta-knowledge gathered to describe the current negotiation scenario. In this paper our focus is from the buyer's point of view. The context thus comprises the current RFQ, buyer's constraints, quotes received from suppliers, supplier profiles, and business rules. History knowledge comprises the past records of relevant negotiation transactions, which might be helpful to be reused for the current scenario. As discussed in the previous sections, the meta-KB describes the ontological property of the current KB in concern. It contains the information about both the negotiation context and the applicable history knowledge.

Before the negotiation process really takes place, there is a preliminary search for qualified offers. The buyer's negotiating agent communicates with a list of available suppliers' negotiating agents. In this preliminary phase, the non-negotiable attributes and inter-attribute constraints specified in the RFQ template are checked. Quotes with non-negotiable attributes unmatched or constraints unsatisfied are discarded. If the number of qualified quotes filtered is small, the negotiating agent can directly start to negotiate with the qualified suppliers respectively. Later close the deal with the optimal bargain. However, if the number of qualified quotes filtered is big, it is impractical to negotiate with the great number of qualified suppliers. Instead, the negotiating agent first ranks the offers. Then choose from the ranked list, a number of top offers to start negotiation. Note that the exact number of top ones to be chosen is up to the buyer. For ranking and finally determining the optimal deal(s), we need to consider the multiple attributes and constraints in the negotiation context. We defer the discussion about evaluation of quotes to the subsection 4.3.2.

After the buyer has got a list of negotiable suppliers, the negotiation process takes place between the buyer and each supplier, respectively. We adopt the argumentation-based negotiation approach [10]. In this approach, agents negotiate as usual by sending each other proposals and counter-proposals, these proposals are accomplished by supporting arguments (explicit justifications). The mechanism of KB's allows a flexible change made to the current RFQ (or quote). To automate the negotiation process, the negotiating agent performs the following steps:

- 1) Examine the quote. The negotiating agent checks if every attribute in the quote is of the optimal value, according to the ranking criteria, which will be presented later when we discuss the evaluation of quotes. It considers the quote as an optimal offer if all attributes have got the optimal values, aban-

dons further negotiation, and goes to step 4. Otherwise, it starts the negotiation in the following steps.

2) Start with the negotiable attribute carrying the highest weight. Check the corresponding attribute provided by the supplier. If it is set to not negotiable by the supplier, then skip to the next negotiable attribute carrying the second highest weight. Otherwise, construct the counter-proposal for the supplier. The meta-knowledge specified in the associated meta-KB is deployed in this step. The way to construct the counter-proposal is detailed in the next subsection. The hidden concern of the buyer is operated in the same way, except that the limits of the hidden attributes are not exposed to the suppliers.

3) Wait for the response sent from the supplier. If the new arrived quote shows no difference compared with the previous one, it indicates that the supplier gives no more concession. The negotiation is abandoned. The negotiation agent goes to the next step. Otherwise, go back step 2 to construct the next counter-proposal.

4) Gather the quote together with others negotiated from other suppliers. Evaluate and rank the quotes. Pick the optimal one as the final deal.

#### **4.3.1. Construct Counter-Proposal**

Counter-proposal contains information about a bargain or a concession. Based on the predefined rank of the attribute values, if the received value is better than required, then just accept it. Otherwise, the mismatch of the attribute values from both sides triggers the bargain. The proposed bargain value depends on the difference between both sides. Unless there is any explicit rule specified beforehand, usually it is a practical way to set the bargain value as the mid value of the difference, provided that the received value is within the predefined valid range. If the received value is out of the predefined valid range, the previously set value is sent again to indicate no concession on the current attribute. Only when there is no difference between two consecutive RFQs/quotes received, the receiving negotiating agent can be determined to abandon further negotiation. The use of rules provides another flexible way to construct the counter-proposal. When there is a need to bargain, the negotiating agent checks the ‘*if*’ condition to trigger the ‘*then*’ action. The rule ‘*Rule#Delivery\_1*’ associated with the attribute ‘*Delivery\_Date*’ in Table 1 is an example illustrated in the previous section 3.2.2.

Meanwhile, the meta-knowledge contained in the meta-KB plays an important role in constructing the counter-proposal during the negotiation process. Existing negotiation systems have put a lot of effort on dealing with constraints and rules in the RFQ, i.e. to associate them to certain attributes. However, it is not easy for them to make efficient reuse of the past records. With meta-KB, we identify the affinity existed between the current KB and the referred KB, namely, with the tags ‘*PREFERENCE*’, ‘*CONCEPT*’, and ‘*AFFINITY*’. With the ‘*PREFERENCE*’ tag, the negotiating agent instructs

the knowledge agent go fetch the attributes specified in the buyer profile, and then attach to the proposal automatically. With the ‘*CONCEPT*’ tag, the knowledge agent is instructed to load the new RFQ desired to construct the counter-proposal. With the tag ‘*AFFINITY*’, the knowledge agent follows the associated rule, search in the affinitive past records, and perform further action to construct the counter-proposal. The contribution here is the efficient way for knowledge reuse from the applicable past records.

#### 4.3.2. Evaluation and Ranking of Quotes

The basic purpose of evaluating a quote is to determine the satisfaction level of the quote, i.e. how well that the quote matches the RFQ. When a group of qualified quotes is ranked, the satisfaction level is used as an index to give the rank in a descending order, so that the top ones are picked to proceed with further negotiation. When the negotiating agent abandons further negotiation, the last quote received is again evaluated to obtain the satisfaction level, which is compared with quotes derived from other peer negotiations. Finally, quote with the highest satisfaction level is picked as the final deal. To evaluate a quote, the first step is to determine if it is a valid quote by examining all the attributes according to the predefined valid range. An invalid quote is discarded. The second step is to produce a satisfaction level using the attribute weights and other constraints defined for the relevant RFQ.

There are various kinds of choice model for evaluating and selecting optimal alternatives. Basically, traditional Operations Research (OR) algorithms and new problem specific algorithms have been followed. Most important ones are the game-theoretic approach, Multi-Attribute Utility Theory (MAUT), and Multiple Criteria Decision Making (MCDM). It has been pointed out in [8] that game-theoretic approach tends to assume that traders’ preferences do not change when they negotiate, and their negotiation stances can not be justified according to different attribute values. MAUT [9] is a tool for making decisions involving multiple interdependent objectives based on uncertainty and preference (utility) analysis. The choice is determined by the maximization of a utility function defined over a set of decision alternative. The weight associated with each attribute in the KB is for the purpose of building the scaling utility functions, which are used for subjective measurement of user preference. On the other hand, MCDM models do not require the specification of a value or utility functions explicitly thus allowing for deviations from rationality. For the time being, our formulation of the problem mainly applies MAUT, and uses the following equation (1) to calculate the satisfaction level about a quote offered :

$$S_T = \sum w_i^2 \cdot m_i \quad \text{for } i = 0 \text{ to } n \quad (1)$$

where,  $S_T$  is the value of the satisfaction level of the quote offered;  $w_i$  is the attribute weight value between 0 to 10;  $m_i$  is a matching score between 0 to 1, depending on how much the attribute value of the offer matches that of the request; and  $n$  is the number of aspects considered. The greater the value of  $S_T$  the better the quote is perceived by the buyer. By squaring the weights, we put an emphasis on the values of greater weights preferred by the buyer. By default, the weight is 5. The matching score  $m_i$  is obtained according to the ranking criteria about the attribute value, which treat different types of attribute in a different way.

There are two types of attributes. One is of discrete enumerated type (e.g. Boolean type). The other is specified by a value range (e.g. an integer range). When a RFQ/quote template is filled, every attribute is also associated a tag to indicate the rank of the possible values. In the case of the enumerated type, the ranking is specified to every possible value. We assume that a smaller number is a higher rank and 1 is the highest rank. Given the rank number  $r$ , and the maximum rank number  $max$  (lowest rank), the matching score is calculated as:

$$m_i = \frac{max - (i - 1)}{max} \quad (2)$$

where  $i$  is the index of rank number  $r$  in the range  $[1, max]$ .

For the example in Table 4, the color of the requested product is one of the attributes in concern. If the quote offers a pink color, the matching score will be 0.75.

Table 4. Ranking example of attributes of enumerated type

Color	Value of Preference	Rank	Matching Score [0,1]
Silver	Favorite	1	1
Pink	Preferred	2	0.75
Black	Acceptable	3	0.5
Golden	Acceptable	3	0.5
Others	Least Liked	4	0.25

Now consider a numeric type with a given range. For values valid within the range, the ranking criteria can be specified as one of the three: 1) the bigger the better, e.g. memory of a computer; 2) the smaller the better, e.g. weight of a laptop; 3) the same if within a certain range, namely, flat, e.g. if the buyer does not care the deliver day within a week. If there is no monolithic ranking in the range, the range can be further divided into a number of sub-ranges, each with a monolithic ranking and a matching score. For example in Table 5, assume that the valid range of an integer attribute is  $[10, 100]$ ,

given the ranking criteria for each sub-range, the matching score of the sub-ranges is calculated using the same way as for the enumerated type. If the quote offers a value 85, the matching score will be 0.762.

Table 5. Ranking example of attributes of numeric type

Valid Range [10, 100]	Value of Preference	Rank	Matching Score [0,1]
[10, 40]	Bigger the Better	[1, 31]	[0.032, 1]
[40, 80]	Flat	1	1
[80, 100]	Smaller the Better	[1, 21]	[0.048, 1]

The foregoing discussion only considers the satisfaction about attributes. In some occasion, for example, the buyer may want to favor a certain supplier from the affinity group, or, want to give more value to an extra free offer provided by some supplier. There are different ways to handle this kind of favor. We adopt the one to add supplier as one more attribute to calculate the satisfaction level. The favorite supplier is given a higher weight and a matching score 1. The additional favor is controlled by a rule for evaluating the quote.

#### 4.4. Post-negotiation Knowledge Management

The main task of post-negotiation settlement is to close the deal and log the negotiation process for future reference and knowledge reuse. This makes the whole trading process a complete loop. Actually, throughout the negotiation process, the KB's database and knowledge base are successively appended and updated. For a successful e-trading cycle, efficient and effective knowledge management must be performed in this phase. There is a pretty large amount of information to be logged. For data-oriented categorization, the fundamental log is the final deal (or, a fail) and seller information, which are logged into the domain spaces of deals and suppliers, respectively. However, that simply log the deals is not enough to provide the convenient reference for future negotiation scenarios. The meta-knowledge about the deal (or, a fail) must be dug out. To our knowledge, most existing negotiation systems lack the capability of knowledge management after the negotiation process. Thus, the limited knowledge gathered from negotiation process hinders the knowledge reuse which is important in automated procurement scenarios.

Each negotiation transaction starts with a RFQ and ends with either a successful deal or a fail. During the process, quotes received and counter-proposals reconstructed are exchanged. The automation of the negotiation

process is guided by the constraints and rules. Past records are referenced to assist in a better bargain. To manage the knowledge in a negotiation transaction, one task is to record the affinity groups that have been referenced in a successful deal. As specified in the meta-KB, rules are associated to a certain affinity group of KB's. Following the rules, the negotiating agent constructs the counter-proposal. If the counter-proposed value is accepted by the other side, it implies that the reference to the affinity group is helpful. For example, some products are subject to the season. During the hot season, demand goes up and the price gets higher. An affinity group can be established to include all the deals completed in the hot season. Useful information is collected such as the supplier information, price dealt and delivery status. Consider the current negotiation scenario was a similar one. By following the rule which makes reference to the affinitive past records, the counter-proposal was finally accepted by the other side. Now the successful deal can be added to the same affinity group, with the most update transaction time recorded. In the future scenario, when the affinity group is referenced again, if the rule is to search for the most update similar record, this deal would be helpful. Moreover, new affinity groups can be established if necessary. Establishing a new affinity group requires human interaction. However, the update of an existing affinity group can be done in an automatic way by the knowledge agent in the post-negotiation phase.

Throughout the discussion, we have seen that rules play the crucial role in negotiation automation. There are two kinds of rules, i.e. empirical rules and experiential rules about empirical and experiential knowledge, respectively. Empirical rules are obtained from the human interaction. Experiential rules are learnt from the past negotiation scenarios. Nevertheless, rules are subject to certain applicable scenarios. There is no guarantee of a better bargain following the predefined rules. In order to achieve a better bargain by the automated negotiation system, it is necessary to provide the flexibility for updating the rules. This is based on the observation on how successful the rule is when applied to a negotiation transaction.

Assume that we have established an affinity group of successful deals with late delivery date proposed but less than 5 days, and a discounted price offered. Applying certain data mining techniques, we can obtain the experiential rule specifying the relationship between the last delivery date and discount offered. Assume that it is the rule present in section 3.2.2 which says "*if the Delivery\_Date in a quote offered by a seller is late but less than 5 days, then the buyer can further ask for a reduced price with 20% off for the requested product*". However, applying the same rule for similar scenario does not guarantee a positive feedback. Thus we can further define *applicable degree* for the rules, which indicates how effective the rule is for the negotiation. For both the positive and negative feedback, the applicable result will be recorded. When the rule is first established, it is given a default applicable degree. Later, the applicable degree increases when there are more successful

applications recorded, while decreases when unsuccessful applications are recorded. If the applicable degree drops to some threshold, the rule is subject to further change. The change can be obtained by applying the same data mining technique again. For example, the new mined result indicates an around 10% off for the price in the updated affinity group. Thus, the rule will be adjusted accordingly. In this way, rules are updated to catch more negotiation knowledge every time in a different scenario. It helps the knowledge reuse and makes the automated negotiation system more practical in use. To allow most flexibility, traders can by themselves adjust the rules manually during negotiation.

## 5. CONCLUSION AND FUTURE WORK

KB allows dynamic definitions of buyer's preference and criteria that can be used as suitable building blocks for negotiation context and business intelligence during agent negotiation. Each negotiation context can be described using both knowledge and meta-knowledge represented in both KB's and meta-KB's, respectively.

When the basis of the whole e-marketplace including the participating agents is constructed using KB's, new knowledge (including business intelligence) relating to the continuous operation of e-procurements can be easily derived by using appropriate data-mining techniques. Such knowledge defined in KB's can be managed and used to do decision support, aiming at facilitating agents in the e-marketplace to do negotiation more efficiently.

Thus the agents communicate just like that they are using a common market-specific language built on KB, which gives buyers and sellers the freedom to interrelate automatically:

- Traders can trade easily, free of the tedious manual interactions with other parties that can involve adjusting the deal, submitting counter offers, incorporating business rules etc.
- Buyers can shop effectively, scan a wide array of offerings and find the offers that best match their requirements.
- Sellers can compete on their strong points: matching products to market needs, differentiating themselves and receiving instantaneous wide exposure. This increases their sales volume.
- The agents provide a high level of flexibility and interplay on a fully automated level, which allows for broad-based, simultaneous activity.

The agents hence have learning ability too. During and after the negotiation process, when an agent encounters some new "ways" of business deals, it will learn about it and be able to apply them in the future. Such new "ways" could be a new specification of a product, strategy, constraint, rule, preference or attribute represented in KB which helps to update the ontology platform among all the agents.

A more detailed model that depicts communication protocols and knowledge management mechanisms that are based on [12] will be developed. Prototype of such a trading agent system with knowledge management functions will be built. Then the prototype will be applied into a few study cases for demonstrating its feasibility and performance. We will investigate into how different sets of popular negotiation algorithms work well with the knowledge management system too.

## 6. REFERENCES

- [1] Simon Fong, Zhuang Yan, "Enabling Agent Negotiation in e-Trading Environments Through Knowledge Beads", *IEEE International Conference on Intelligent Engineering Systems 2002* (May 26-28, Opatija)
- [2] Zhuang Yan, Simon Fong, "Automated Negotiation with Multiple Concept-based Alternatives in Multi-agent Systems", *International Conference on Intelligent Systems and Control 2003* (June 25-27, Salzburg)
- [3] Patrick Hung, Ho-fung Leung, "Facilitating e-Negotiation Processes with Semantic Web Technologies", *IEEE Proceedings of the 38<sup>th</sup> Hawaii International Conference on System Sciences*, 2005, pp.1-10.
- [4] Sheng Li, Qiang Yang, 2001, "ActiveCBR: an agent system that integrates case-based reasoning and active databases", *Knowledge and Information Systems* (2001) 3: 225-251, 2001 Springer-Verlag London Ltd.
- [5] Sheng Zhang, Song Ye, Fillia Makedon, James Ford, "A Hybrid Negotiation Strategy Mechanism in an Automated Negotiation System", *ACM Electronic Commerce '04*, May 17-20, 2004, pp. 256-257
- [6] Cuihong Li, Joseph Giampapa, Katia Sycara, "Bilateral Negotiation Decisions with Uncertain Dynamic Outside Options", *IEEE Proceedings of the First International Workshop on Electronic Contracting*, 2004.
- [7] Susan Conway, Char Sligar, *Unlocking Knowledge Assets, Knowledge Management Solutions from Microsoft*, Copyright 2002 by Microsoft Corporation
- [8] Jennings, N. R. et al., "Automated negotiation: prospects, methods and challenges", *International Journal of Group Decision and Negotiation*, 10(2), pp.199-215, 2001
- [9] Raiffa, H. and R. Keeney. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. New York, Wiley, 1976
- [10] Gregory E. Kersten, "The Science and Engineering of E-negotiation: Review of the Emerging Field", *InterNeg Reports* 05/02
- [11] J. A. Rodriguez-Aguilar, A. Giovanucci, A. Reyes-Moro, F. X. Noria, J. Cerquides, "An Agent-based Decision Support for Actual-World Procurement Scenarios", *IEEE/WIC International Conference on Intelligent Agent Technology*, pp. 371, 2003
- [12] Maria Indrawan, Thalerngsak Kijthaweesinpoon, Bala Srinivasan, A. Sajeew, "Coalition Formation Protocol for E-Commerce," *IEEE ICISIP*, pp.403-408, 2004.