Negotiation Paradigms for E-Commerce Agents using Knowledge Beads Methodology

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
shi@csnet4.cs.tsinghua.edu.cn

Abstract

The technology of using computer Agents for B2B trading in Cyberworld is getting prevalent nowadays. Much of the research work has been done on the models, architectures and service provision in the past. However, agent negotiation remains as a challenge in making the whole trading process fully automated, due to its fuzzy and complex nature. The lack of interoperability and knowledge-reuse that limits to case-based reasoning pose certain drawback. An object-oriented ontology-based Knowledge Bead (KB) method for knowledge representation has been proposed in [1]. It was designed as a foundation to enable agent negotiation in e-trading environment in a systematic way. This paper continues the research work of KB on formulating its theorems and methodology. Some typical negotiation paradigms using appropriate strategies based on the KB's methodology are presented as well. In particular, KB's taxonomies and their use in the whole negotiation process are discussed. The main advantage of this approach is the ability to describe the deal that is under negotiation in an object-oriented format that in turn allows reasoning, optimizing, knowledge reuse and management.

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 [4]. This suggests a more sophisticated negotiation system is needed, which can do risk assessment, coalition formation, criteria evaluation and knowledge management. It would embrace the full automation of trading agents in the Cyberworld that are capable to cope with the rapid changes experienced by various business-to-business (B2B) commercial activities. And these changes contribute to the problem of information explosion that is anticipated in the Cyberworld in the near future. 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 description carries a criterion and a weight specifying how important that criterion is. 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. Moreover, the self-learning ability of KB enables the accumulation of business intelligence through each negotiation process, so that the subsequent procurements can be benefited from the past ones. As virtually everything in a negotiation process is specified using KB's, information discovery in the e-trading environment now becomes standard operations on the defined KB's. For the same reason, a common ontology is formed by KB's to facilitate the same points of reference for the negotiation parties.

Once we have the ontology-based foundation, namely KB, we need to further develop a complete set

of negotiation protocols based on the KB's methodology. A widely accepted model is to split the negotiation process in several phases, such that the negotiation progresses through pre-negotiation preparation, conduct of the negotiation, and post-negotiation settlement [6] [7]. Example systems include the Inspire system developed in 1995 to provide training resources and to study the use of support tools [8] [9], the WebNS that is an example of Web-based negotiation support system (WNSS) [10] [11], and the SmartSettle system which is a commercial system extended from the research system ICANS [12] [13]. The strength of the phase model is in its integration of human facilitation in the pre-negotiation phase, extensive analytical support during the negotiation, and post-settlement phases. We adopt this phase model in developing the KB-enabled negotiation protocols. This paper first focuses upon the formulation of the taxonomies of KB's, that come in three types: category-based, ontology-based, and operation-based. Then some typical negotiation paradigms and associated strategies are described to illustrate the usage of the KB's methodology

2. Methodological Foundations

In [5], Gregory E. Kersten proposed a scientific approach to negotiation that formulates four different views on the participants involved in the negotiation processes, their characteristics, roles and theories, approaches and models used for the construction of their representations. The participants involved in decision-making and negotiation include the negotiator, advisor, principal and an agent who represents the principal. Third parties and stakeholders may also be involved. To simplify illustration, we name all the participants as traders, i.e. buyers or sellers. Traders may have different set of characteristics such as their preferences, attitude to risk, attitude and concern towards others, power, negotiation style, and culture. The traders' roles define sets of activities, for example, analysis, decision-making or advice. Both characteristics and roles are generalized to formulate theories, frameworks and models. There are two categories of models. Models of problem and expert knowledge are used in pre-negotiation processes. The negotiation and post-negotiation processes are described with models that incorporate the dynamic aspect of the negotiation, namely the choice and concession models, argumentation models, expert models and models which describe the negotiation protocol. The main consideration in this paper is on establishing KB's taxonomies, and building the negotiation protocol models with the aid of KB's. Pre-negotiation and the negotiation processes are discussed in details.

3. Taxonomies of KB's

Throughout the trading process, there is a large amount of data that must be collected and processed. Each participant maintains its own database and knowledge base, and is represented by an agent that has access to the data and knowledge. The major difference between data and knowledge is that the latter is extracted from the former through the reuse and learning process. For automatic negotiation, appropriately formulated knowledge plays a crucial role. This includes trader's preference, conditions and criteria, conventional rules for negotiation, statistics of similar deals, and past trading records. As proposed in [1], KB's serve as the building blocks for building the knowledge base for e-commerce. To provide the same points of references for communicating agents, the taxonomies of KB's are built up. The goal of a corporate taxonomy is not only to provide a list of authorized terms for use in writing and information seeking [22], but also to create maps between concepts to connect traders with the right knowledge at the right time. There are three types of taxonomies of KB's: category-based, ontology-based, and operation-based.

Category-based taxonomies provide the specification for all the entities in the e-commerce system. An entity can be a *Request For Quote* (RFQ), a trader, a deal, or anything that is represented as a KB or a set of relevant KB's. Category-based taxonomies are implemented using 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 virtually carries a category name as an attribute, which helps to link RFQ between buyer and seller from the same category. A category name is represented as a sequence of labels corresponding to the edges in the path, e.g.

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

KB's describing traders and deals also belongs to category-based taxonomies. They have a different space graph other than the product space, but with the same category hierarchical structure.

Creating ontology-based KB taxonomies involves reviewing entries against an established set of category-based taxonomies, looking for similarities, affinities, and dependencies. The ontology-based KB associates the agents with business intelligence necessary for automated negotiation. To support this collaborative filtering feature, the trading agent needs further evaluation of the KB for RFQ, to determine similarity between the trader's choice and choices stored in the database. Usually there will be a pivot attribute identified to connect related KB's.

The simplest form of similarity is identified for virtually the same entity but with different category name. For example, a supplier providing electronics and computers also provides lamps that belong to the *home* category. To link this same entity under different categories, each KB category will be assigned a canonical name (CNAME), i.e. a primary category name. A KB with a different category name from the canonical one will have the pivot attribute storing the canonical category name, i.e. implemented as a symbolic link in the graph representing the category space. Another kind of similarity resides in attributes other than the category name. Under certain circumstances, a trader may find it helpful to establish a link between different KB's through their common attributes. All attributes in the common KB become pivot attributes. For example, a buyer may specify in the bill-of-material (BOM) his/her favorite color in the common KB. When the buyer decides to compromise on the favorite color during further negotiation, it's only necessary to change the color attribute in the common KB, which will be applicable to all other KB's in the BOM. Other attributes can also be specified as pivot attributes. To allow the traders to dynamically change their preferences and evaluation criteria in a changing world and situation, KB incorporates with the concept-based alternatives in the proposals to speed up the negotiation process [2]. Changing the value of a pivot attribute defined in a first KB may lead to another KB with different weights associated to the attributes. This concept-based automated negotiation model thus provides the benefits of allowing greater flexibility in describing the desired product under different concepts.

Affinities are identified to group similar KB's for future reference. They are described as various forms of business intelligence, which possibly can be mined from quotes and deals offered by suppliers 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 record or a similar preference profile in the seller's previous procurement history, then the seller's trading agent can predict the preference of the buyer based on the past KB's. 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 [16]. Another example is to group all the deals with a price discount. The relevant supplier information will also be extracted from this group, and added into an affinitive supplier list. Later when all the possible alternative quotes need to be evaluated to find out the optimal offers, the affinity can be used as an additional constraint, so that the buyer agent may give more value to the suppliers in the affinitive list.

Dependencies between KB's are the results of imposing trading rules and other constraints on KB's exchanged during quote evaluation and negotiation process. Constraints can be specified as attributes in KB, or, as a triggering event for the current KB. Dependency refers to the latter. For example, a BOM consists of two KB's, the trader can add one constraint that the suppliers of the two KB's should be the same. The buyer can add '*supplier*' as an attribute in both KB's, then establishes a link between using '*supplier*' as the pivot attribute.

The ontology-based taxonomies are illustrated in Table 2. In this example, the first tag IDENTIFY indicates the current KB's identifier. COMMON refers to the KB containing common attributes applicable to the current one. CNAME refers to the canonical name of the KB's category. CONCEPT is used to switch to another KB with a different concept on the attribute. AFFINITY refers to the KB containing certain business intelligence. DEPENDENCY refers to the KB for some triggering rule or constraint. Note that except IDENTIFY and CNAME, there may be more than one entry in the table defined for the other five tags, depending on the real situation.

Separating category-based and ontology-based taxonomies has an advantage of reducing communication cost between agents. Only category-based KB will be transferred in the first phase of asking for quotes, so to eliminate unnecessary traffic. How to make use of the related KB's, in particular, different affinities, is scenario-oriented, and is discussed as follow.

Operation-based taxonomy is a list of operations that is used to support KB's trading capabilities; in particular, to establish links among related KB's. This includes the following:

(1) Searching. The KB identifier, represented as the category name together with the name of the leaf node in the graph, is used as the key for searching. 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

(2) Classification. When a new KB is added, according to the category to which it belongs, it is inserted into the product graph, supplier graph, or deal graph, respectively.

(3) Appending common attributes. Attributes from the COMMON KB are appended to the current KB. After the preliminary RFQ has been appended, the buyer agent then contacts the seller agents based on this completed RFQ.

(4) Switching to alternative concept template. The concept-based ontology is implemented when given the new KB to be switched to, and together with the pivot attribute. When the seller agent offers quote with a different attribute from the one requested in the original KB, the buyer agent will compare the offered attribute with the pivot attribute. If a match is found, the original KB is switched to a new KB that will be proposed subsequently.

(5) Establishing affinities. Since different affinity represents different business intelligence, which often comes in various forms in real practice, the operations are thus scenario-oriented. For example, linking to the affinitive list of suppliers will assign a higher priority to the suppliers in the list. Affinitive KB can also carry pivot attributes if it is necessary to tell the agent which attributes in the KB are to be linked, for example, an attribute '*discount*' in KB of a completed deal suggests the agent make further proposal for discounted price.

(6) Imposing constraints. There are two alternatives to deal with a constraint. If the constraint is attribute-oriented, it can be embedded into the KB, and will be evaluated as a constraint for the multi-objective problem formulation. Otherwise, the constraint is exchanged together with the KB, as a justification during the negotiation process using argumentation-based strategies.

(7) Operations on KB's attributes. These include common arithmetical and logic operations performed on attributes from different KB's, e.g. comparison, addition, subtraction, etc.

4. Negotiation

The negotiation process is divided into three phases: pre-negotiation preparation, conduct of negotiation, and post-negotiation settlement. The first two phases are discussed in details in this paper.

4.1. Pre-negotiation Preparation

Defining the negotiation context is the main job in pre-negotiation analysis. In general this includes the specification of the bill of material, constraints for deal settlement, and the development of trader's preference profiles. Depending on the specific product, the trader can choose to use a predefined KB template within the same product category, with further modification on the attributes, or to build a new KB template on its own. An example of a KB template filled for purchasing digital camera is given in Table 1.

Successful paradigms allow a flexible negotiation while following the strict constraints. In particular, trader's preferences described 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 ontology-based KB in Table 2. In category-based KB, 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. Moreover, there are some dynamic issues to be considered while 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 and mini-movie are specified as 'Not Required', which means that the buyer has no specific requirements in these two attributes. 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. There are also hidden attributes that are not to be seen by the other side, usually considered as constraints on the KB. They can be used in either the automatic evaluation of offers as a constraint, or argumentation-based negotiation as a justification, or both, depending on the trader's business policy.

4.2. Conduct of the Negotiation

4.2.1. Negotiation Strategies. Negotiation is a strategy-based process governed by some explicit and implicit rules. For different negotiation situations and types, the trading agents are able to select appropriate

negotiation strategies based on the current mission constraints and the current negotiation phase, which can be further divided into three steps. The preliminary search for available offers mainly involves the rough matching of product category to which the RFQ KB belongs. The buyer agent communicates with a list of available seller agents, using only the non-negotiable attributes in the KB template. The next important negotiation activity is to choose, from the possible alternatives, one or more optimal offers. Searching the optimal solutions from a list of possible offers, multiple objectives are involved; such searching and optimization problems are known as multi-objective optimization problem (MOOP). In this phase mathematical algorithms for solving MOOP are needed to facilitate the automated selection process. Based on the resulted, qualified offers, the buyer agent then further negotiates with the seller agents using argumentation-based negotiation strategies [5]. In this approach, while agents negotiate as usual by sending each other proposals and counter-proposals, these proposals are accomplished by supporting arguments (explicit justifications). Ontology-based KB's play an important role in this last step. Now the trader agents can at the same time make use of past incidents, cases and histories, which are all specified in ontology-based KB's, to reach a best deal. Figure 2 shows how KB's are involved in different negotiation phases.

Note that in order to reach a best deal, it is necessary to repeat the activities in the second and third steps in a loop until the agreement from both sides are achieved.

4.2.2. Mathematical Algorithms for Multi-Objective Optimization. There are various kinds of choice model for evaluating and selecting optimal alternatives. Most important ones are the game-theoretic approach, Multi-Attribute Utility Theory (MAUT), Multiple Criteria Decision Making (MCDM). Game-theoretic approach has been pointed out [23] that it tends to assume that traders' preferences do not change when they negotiate, and their negotiation stances cannot be justified according to different attribute values. MAUT [14] [15] 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. They are designed to guide decision makers and help them to achieve a Pareto-optimal compromise. Both MAUT and MCDM try to solve the general multi-objective optimization problems. This section discusses in details the mathematical models we used for searching the best matches from all the offers.

The application of multi-objective optimization algorithms can be classified into two fundamentally different approaches [17]. Most classical multi-objective optimization algorithms follow the preference-based approach. Of these, the weighted approach converts multiple objectives into a single objective by using a weighted sum of objectives. The weight vector is user-defined in KB's attributes. It is possible to obtain one particular trade-off solution. For example, assume that we are faced with the three objectives of minimizing the price of the digital camera, maximizing the resolution, and minimizing the camera weight. In formulating the optimization problem, it is a usual practice to choose weights such that their sum is one. However, the user should not be worried about such a constraint when specifying his/her preferences on the attributes. Depending on the number of attributes to be optimized, and their relevant weights, the algorithm generates a set of new weights such that their sum is one. Table 3 illustrates the conversion for the example.

In this case, the buyer thinks price is most important (10), at the same time he also considers resolution (8) and product size (5). The corresponding weight for optimization is calculated as:

$$wo = \frac{wp}{\sum_{m=1}^{M} wp_m} \tag{1}$$

where wp is the weight of preference, wo is the converted weight for optimization, wp_m is the weight of preference for the m-th attribute.

Setting up an appropriate weight also depends on the scaling of each objective function. It is likely that different objectives take different orders of magnitude. In the above example, the price may vary between 100 to 2000 dollars, whereas the resolution may vary between 0.2 megapixels to 6.35 megapixels (effective), and the weight may vary between 150 grams to 600 grams (empty). When such objectives are weighted to form a composite objective function, it would be better to normalize them appropriately so that each has more or less the same order of magnitude. For this example, we may multiply the price by 10^{-3}, the resolution by 10^{-6} and the weight by 10^{-2} to make them equally important.

After the objectives are normalized, a composite objective function can be formed by summing the weighted normalized objectives and the multi-objective

optimization problem is then converted to a single-objective optimization problem as follows:

$$\text{Minimize } F(x) = \sum_{m=1}^{M} wo_m f_m(x)$$
$$\text{subject to } x_i^{(L)} \le x_i \le x_i^{(U)}, i = 1,2,...,n. \quad (2)$$

where wo_m is the weight for optimization for the *m*-th objective function. Each decision variable x_i is restricted to take a value within a lower and an upper bound. Each objective function can be either minimized or maximized. Many optimization algorithms are developed to solve only one type of optimization problems, such as e.g. minimization problems. In the above example, price and weight are to be minimized, while resolution is to be maximized. We can convert a maximization problem into a minimization one by multiplying the objective function by -1, by the duality principle [18] [19] [20], and vice versa. The example is then formulated as follows:

$$\text{Minimize } F(x) = 0.435p - 0.348r + 0.217w$$
$$\text{subject to } \begin{array}{l} 0.1 \le p \le 2, \\ 0.2 \le r \le 6.35 \\ 1.5 \le w \le 6 \end{array} \quad (3)$$

where *p*, *r*, and *w* correspond to the normalized price, resolution, and camera weight, respectively.

For problems having a convex Pareto-optimal front, this method guarantees finding solutions on the entire Pareto-optimal set. However, since weights in KB's are user-specified relative importance vector for the attributes, unless a reliable and accurate preference vector is available, the optimal solution obtained by such methods is highly subjective to the particular user. In some real scenarios, finding a reliable importance vector is difficult in the absence of any knowledge of the Pareto-optimal solutions. For example, a user, who has little knowledge about digital cameras, wants to purchase an 'ideal' camera with high quality and low price, and is sure of emphasizing cost more than quality. Instead of filling in the exact weights for optics and resolution, memory, and other sophisticated features, the user is provided with a preferred set of Pareto-optimal solutions with more solutions crowded near the minimum-cost solution. He/she then chooses one of the obtained solutions using high-level information. For such cases, evolutionary multi-objective optimization algorithms (EAs) are adopted, in which no special importance is given to any particular objective and a set of Pareto-optimal solutions are desired to be obtained. Afterwards, higher-level knowledge based on relevant ontology-based KB's is used for choosing one solution from the obtained set of solutions. Among different evolutionary algorithms, we will make use of genetic algorithms (GAs) [21]. GAs can be easily and conveniently used in parallel systems. Since in B2B trading environment, a great amount of computational time is spent in searching and evaluating offers, with multiple processors, all offers in a population can be evaluated in a distributed manner. This will reduce the overall computational time substantially. A biased sharing approach [17] can be used to find a preferred distribution in the region of interest. It is more practical and less subjective than finding one biased solution in the region of interest. There are two advantages: (i) the search effort is reduced, and (ii) better precision in solutions in the desired region is achieved. The detailed algorithms will be described in another paper.

The foregoing discussion focuses on the optimization of attributes of numeric types. However, there are circumstances when non-numeric attributes are to be considered, e.g. the color of a product is one of the concerns. In this case, the buyer/seller is required to specify which colors, for example, are their favorites or preferred, which are acceptable, and which are not acceptable. Based on the information provided, Table 4 is constructed.

As a simple example for handling constraints in optimization problem, assume that the requirement of color is added as a constraint to the optimization problem. The equation (3) is modified as follows:

$$\text{Minimize } F(x) = 0.435p - 0.348r + 0.217w$$
$$\text{subject to } \begin{array}{l} 0.1 \le p \le 2, \\ 0.2 \le r \le 6.35 \\ 1.5 \le w \le 6 \\ 2 \le c \le 4 \end{array} \quad (4)$$

where *c* represents the product color, which is required to be acceptable at least.

4.3. Post-negotiation Settlement

The main task of post-negotiation settlement is to close the deal and log the negotiation process for future reference. This makes the whole trading process a complete loop. A great amount of efforts need to devote to knowledge management in this phase. Actually, throughout the negotiation process, the KB's ontology database is successively appended and updated. How to maintain and utilize the KB's ontology database in an efficient and effective way is another vital concern for the whole architecture

construction. We will have a separate paper discussing KB's knowledge management in details.

5. Conclusions and future work

The paper is the continuing research work of defining a knowledge beads methodology for agent negotiation in B2B e-commerce. It focuses on the definition of KB's three taxonomies, and the description of negotiation paradigms based on KB's methodology. We will in the next step concentrate on developing efficient knowledge management strategies for KB's.

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] Simon Fong, Zhuang Yan, A Concept-based Model to Facilitate Automated Negotiation in Multi-agent Systems, *Second International Workshop on Intelligent Systems Design and Applications* (August 7-8, 2002, Atlanta, USA)

[3] Minghua He and Ho-fung Leung, *Agents in E-Commerce: State of the Art, Knowledge and Information Systems* (2002) 4: 257-282

[4] Collis J, Lee L (1998) Building electronic marketplaces with the ZEUS agent tool-kit. In Noriega P, Sierra C (eds). Agent-mediated electronic commerce. *Lecture notes in artificial intelligence*, vol 1571, Springer, Berlin, pp 1-24

[5] Gregory E. Kersten. The Science and Engineering of E-negotiation: Review of the Emerging Field. *InterNeg Reports* 05/02

[6] Gulliver, P. H (1979). *Disputes and Negotiations: A Cross-Cultural Perspective.* Orlando, FL, Academic Press

[7] Graham. J. L., A. T. Mintu and W. Rogers (1994). Explorations of Negotiation Behaviors in Ten Foreign Cultures Using a Model Developed in the United States. *Management Science* 40(1): 72-95

[8] Kersten, G.E. and S. Noronha (1999a). Negotiations via the World Wide Web: A Cross-cultural Study of Decision Making. *Group Decision and Negotiation* 8: 251-279

[9] Kersten, G.E. and S. J. Noronha (1999b). *WWW-based Negotiation Support: Design, Implementation, and Use. Decision Support Systems* 25: 135-154

[10] Yuan, Y., H. Ding, J. B. Rose, et al (1999). *Multi-party Interaction in a Web-Based Negotiation Support System. WebNS.* https://ecomlab.mcmaster.ca/webns/papers/Interact.html, Accessed: April 23, 2002

[11] Yuan, Y., J. B. Rose and N. Archer (1998). A Web-Based Negotiation Support System. *Electronic Markets* 8(3): http://www.mediamanagement.org/netacademy/publications.nsf/all_print_pk/1076

[12] Thiessen, E. M. and D. P. Loucks (1994). *ICANS: Interactive Computer Assisted Negotiation Support. Computer Assisted Negotiation and Mediation: Prospects and Limits*, R. Shell, Ed., Harvard, Harvard Law School, Cambridge, MA

[13] Thiessen, E. M., D. P. Loucks and J. R. Stedinger (1998). *Computer-Assisted Negotiations of Water Resources Conflict. Group Decision and Negotiation* 7(2): 109-129

[14] Raiffa, H. and R. Keeney (1976). *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*. New York, Willey

[15] French, S. (1998). *Decision Theory. An Introduction to the Mathematics of Rationality*. New York, Ellis Hoerwood

[16] Pennock, D. M., E. Horvitz, S. Lawrence, et al. (2000). Collaborative Filtering by Personality Diagnosis: A Hybrid Memory- and Model-Based Approach. *16th Conference of Uncertainty in Artificial Intelligence*, San Francisco, Morgan Kaufann

[17] Kalyanmoy Deb, *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Ltd. 2001

[18] Deb, K. (1995). *Optimization for Engineering Design: Algorithms and Examples*. New Delhi: Prentice-Hall

[19] Rao, S. S. (1984). *Optimization: Theory and Applications*. New York: Wiley

[20] Reklaitis, G. V., Ravindrann, A. and Ragsdell, K. M. (1983). *Engineering Optimization Methods and Applications*. New York: Wiley

[21] Hajela, P., Lee, E. and Lin. C. Y. (1993). Genetic algorithms in structural topology optimization. In *Proceedings of the NATO Advanced Research Workshop on Topology Design of Structures*, pp. 117-133

[22] Susan Conway, Char Sligar, *Unlocking Knowledge Assets, Knowledge Management Solutions from Microsoft*, Copyright ©2002 by Microsoft Corporation

[23] Jennings, N. R. et al. 2001, Automated negotiation: prospects, methods and challenges. *International Journal of Group Decision and Negotiation*, 10(2), 199-215

Table 1. An Example KB Template Filled for Purchasing Digital Camera

Attribute	Value	Weights	**Negotiable**
Optics&Resolution		10	
Optical Zoom	3x	8	
Autofocus	Yes	10	
Resolution	>= 340,000	10	
Removable Memory	Yes	10	
Features		7	
Display Size	>= 1.5 Inches	3	
Tripod Mount	Not Required	0	
Mini-Movie	Not Required	1	
Batteries	Lithium Ion	8	
Size	Pocket Size	5	
Cash on Delivery		N/A	NOT
Hidden			
Price	<=$400		
Quantity	[1, 5]		

Table 2. An Example ontology-based KB

IDENTIFY	*KB Identifier*		
COMMON	*KB Identifier 1*		
CNAME	*Category Name*	*Attribute*	
CONCEPT	*KB Identifier 2*	*Attribute*	
AFFINITY	*KB Identifier 3*		
DEPENDENCY	*KB Identifier 4*	*Attribute 1*	*Attribute 2*

Table 3. Weight for optimization

Attribute	Weight of Preference	Weight for Optimization
Price	10	0.435
Resolution	8	0.348
Weight	5	0.217

Table 4. Example of attributes of non-numeric type

Color	Value of Preference	Value for Optimization
Silver	Favorite	4
Pink	Preferred	3
Black	Acceptable	2
Golden	Acceptable	2
Brown	Unacceptable	1

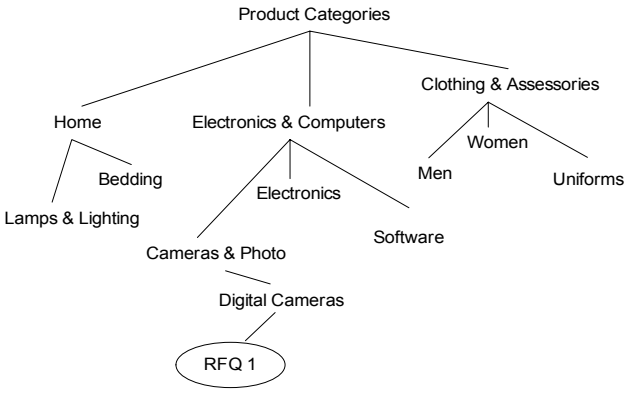

Figure 1. Part of the product space of category-based KB's

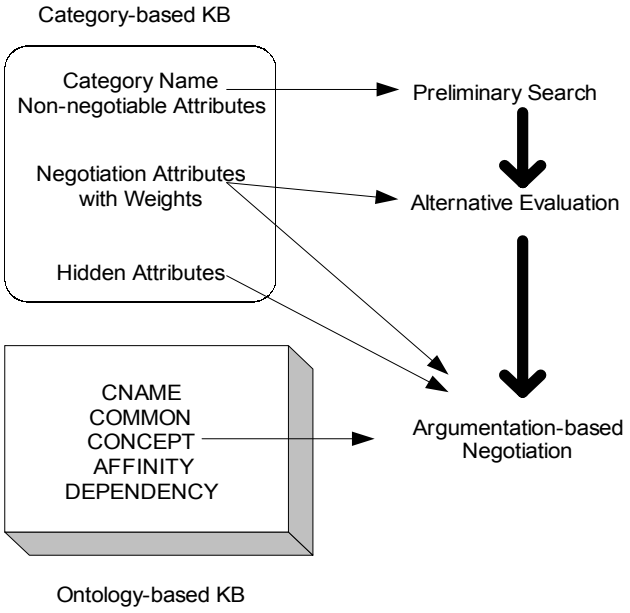

Figure 2. Using KB's in the negotiation process