

# Towards a web services and intelligent agents-based negotiation system for B2B eCommerce

Raymond Y.K. Lau \*

*Department of Information Systems, City University of Hong Kong, Tat Chee Avenue, Kowloon, Hong Kong, China*

Received 2 April 2006; accepted 22 June 2006

Available online 14 July 2006

## Abstract

With the explosive growth of the number of transactions conducted via electronic channels, there is a pressing need for the development of intelligent support tools to improve the degree and sophistication of automation for eCommerce. With reference to the BBT business model, negotiation is one of key steps for B2B eCommerce. Nevertheless, classical negotiation models are ineffective for supporting multi-agent multi-issue negotiations often encountered in eBusiness environment. The first contribution of this paper is the exploitation of Web services and intelligent agent techniques for the design and development of a distributed service discovery and negotiation system to streamline B2B eCommerce. In addition, an effective and efficient integrative negotiation mechanism is developed to conduct multi-party multi-issue negotiations for B2B eCommerce. Finally, an empirical study is conducted to evaluate our intelligent agents-based negotiation mechanism and to compare the negotiation performance of our software agents with that of their human counterparts. Our research work opens the door to the development of the next generation of intelligent system solutions to support B2B eCommerce.

© 2006 Elsevier B.V. All rights reserved.

*Keywords:* Automated negotiations; Intelligent agents; Web services; eCommerce

## 1. Introduction

With the rapid growth of the number of transactions conducted via electronic channels such as the Internet, there has been an ever increasing demand to develop advanced computational tools to streamline business-to-consumer (B2C) and business-to-business (B2B) eCommerce. Although most of the initial Internet-based eCommerce was in the form of B2C, B2B now constitutes a much larger portion of the overall eCommerce landscape. It is widely believed that B2B will continue to grow and will be the predominant means of doing business in the near future [43]. Accordingly, there is a pressing need of applying sophisticated tools to enhance B2B eCommerce [17,32]. It is generally believed that the “Negotiation” stage is crucial for B2B eCommerce since whether a transaction is

profitable or not for an organization is highly dependent on the outcomes from this stage [14,35]. This paper focuses on leveraging Web services technology and intelligent agent techniques for the development of automated negotiation systems to streamline B2B eCommerce.

### 1.1. Background

By increasing the degree and sophistication of automation, B2B eCommerce will become much more dynamic, efficient, and hence more widely adopted by organizations. Intelligent software agents are promising to enhance the degree of automation and sophistication of B2B eCommerce [15,17,34,53]. Software agents are encapsulated computer systems situated in some environments such as the Internet and are capable of flexible, *autonomous* actions in that environment to meet their design objectives [21,51]. The notion of agency can be applied to build robust architectures for automated negotiation systems

\* Tel.: +852 2788 8495; fax: +852 2788 8694.

E-mail address: [raylau@cityu.edu.hk](mailto:raylau@cityu.edu.hk).

within which a group of software agents communicate and autonomously make negotiation decisions on behalf of their human users. It is believed that *negotiation agents* can reduce human negotiation time and identify optimal or near optimal solutions from combinatorially complex negotiation spaces [15,27,29,31].

Negotiations are ubiquitous and conducted in various contexts such as the formation of virtual enterprises [17], marketing, establishing business contracts, managing labour disputes [45], resolving border conflicts, handling hostage crisis [50], etc. Negotiation refers to the process by which group of agents (human or software) communicate with one another in order to reach a mutually acceptable agreement on resource allocation (distribution) [33]. Given the ubiquity and importance of negotiations in various contexts, research into negotiation theories and techniques has attracted attention from multiple disciplines such as distributed artificial intelligence (DAI) [24,26,45], social psychology [2,38,39], game theory [37,48,42], operational research [10,18], and more recently in agent-mediated electronic commerce [17,53]. Despite the variety of approaches towards the study of negotiation theory, a negotiation model consists of four main elements: *negotiation protocols*, *negotiation strategies*, *negotiation environments* and *agent environments* [12,27,33].

### 1.2. Requirements of practical negotiation systems for B2B eCommerce

Practical negotiation mechanisms for B2B eCommerce must be computationally efficient. Therefore, negotiation agents should be developed based on the assumption of *bounded* rather than *perfect rationality* [33]. In addition, since individual businesses have the freedom to employ their own negotiation strategies and mechanisms to negotiate with their business partners, it implies that a practical negotiation mechanism should be based on a distributed rather than a centralized decision making model. As an agent (i.e., the representative of a business) only knows its own preferences (utility function) but not the preferences of its opponents [21,11] in typical B2B negotiation settings, negotiation agents must be able to make sensible decisions based on incomplete and uncertain information. Since B2B negotiations often involve multiple parties who will exploit many issues (e.g., price, quantity, product quality, etc.) simultaneously, practical negotiation mechanisms must be able to support multi-lateral multi-issue negotiations. In particular, B2B relationships and interactions should be modeled as *integrative* negotiations [14] (i.e., the cooperative process of resolving multiple interdependent but not mutually exclusive goals). This is in stark contrast to *distributive* negotiation in B2C where win–lose type of negotiations are not uncommon. In short, the goal of an automated negotiation system is to search for a win–win situation efficiently given the specific preferences of the participating human negotiators.

### 1.3. Contributions of the paper

To harness the full potential of eCommerce, particularly B2B eCommerce, a new model of software is required to enhance the degree and sophistication of automation in the exchanges of B2B transactions. The first contribution of this paper is the demonstration of the design and implementation of an intelligent agents-based negotiation Web service to streamline B2B eCommerce. Although many theoretical negotiation models have been reported in the literature [16,25,41,42,54], these models have limitations in supporting B2B negotiations. The second contribution of this paper is the development of an effective multi-party multi-issue negotiation mechanism which are based on appropriate assumptions for B2B negotiation situations. With the emerge of agent-mediated eCommerce, several agent-based negotiation prototype systems have been developed. Nevertheless, few empirical studies were conducted to directly compare the performance of the autonomous negotiation agents with that of their human counterparts. The third contribution of this paper is the empirical study of our intelligent agents-based negotiation system and a direct comparison between the negotiation performance of software agents and that of their human counterparts.

### 1.4. Outline of the paper

The remainder of the paper is organized as follows. Section 2 highlights previous research in the related area and compare these research work with ours. The system architecture of the Web services and intelligent agents-based B2B service discovery and negotiation system called WSDNS is illustrated in Section 3. The details of our multi-party multi-issue negotiation mechanism which guarantees Pareto optimal results are described in Section 4. Section 5 reports the experimental results of the intelligent agents-based multi-party multi-issue negotiation system. A qualitative analysis of our empirical test results is conducted in Section 6. Finally, we offer concluding remarks and describe future direction of our research work.

## 2. Related research work

The Kasbah e-marketplace is one of the early attempts at exploiting agent technology for automated negotiations in eCommerce [35]. A group of buyer agents and seller agents meet at the centralized Kasbah e-marketplace. These agents proactively seek out potential buyers or sellers and negotiate with each other on behalf of their owners. The objective of each agent is to complete an acceptable deal based on the user-specified constraints such as initial asking (or bidding) price, a reservation price, a date by which to complete the transaction, and restrictions on which parties to negotiate with and how to change the price over time. A Kasbah agent is restricted to exercising one of the three negotiation strategies: anxious, cool-headed, and

frugal, corresponding to a linear, quadratic, and exponential function respectively for increasing (decreasing) its bid for a product over time. Unfortunately, the Kasbah agents can only negotiate over the single issue of price. However, B2B negotiations often involve multiple issues. Moreover, the Kasbah agents can only act according to one of the three pre-defined negotiation strategies which may not lead to the optimal negotiation results. The negotiation agents proposed in this paper can negotiate multiple issues at the same time. Moreover, our agents can autonomously search for a Pareto optimal solution in a multi-dimensional negotiation space. Such a feature allows a win-win type of negotiation process and is particularly suitable for B2B eCommerce. Since our service discovery and negotiation system adopts the Web services system architecture and software development standard, an organization will find it much easier to connect to our negotiation Web service by invoking their own client application. This open system architecture is a desirable for the development of practical B2B system solutions, and it is in sharp contrast with the proprietary protocol employed by the Kasbah e-marketplace.

The Michigan AuctionBot is a general purposed Internet-based auction server hosted by the University of Michigan [52]. Sellers can create new auctions on AuctionBot by choosing from a set of pre-defined auction types and then enter their specific auction parameters such as clearing time, minimum bid increment and whether proxy bids are allowed. In general, a seller will set a reservation price after creating an auction, and let AuctionBot manage the bidders and enforce the bidding rules according to the chosen auction protocol. One distinct advantage of AuctionBot is that it offers Application Programming Interfaces (APIs) so that buyers can create their software agents to bid on their behalf at the AuctionBot virtual auction house. Nevertheless, as with other commercial auction sites such as eBay ([www.ebay.com](http://www.ebay.com)), such an e-market only allows negotiations over a single issue of price. Although these kind of e-markets or auction houses are popular for B2C eCommerce, they are ineffective for B2B eCommerce where multiple negotiation issues are often explored. Our negotiation system differs from the Michigan AuctionBot in that multi-party multi-issue negotiations can be conducted. This feature makes our system more suitable for the B2B type of negotiations. In addition, the proposed dynamic business solution in this paper not only provides an effective B2B negotiation mechanism but also supports business service discovery and transaction processing.

MAGNET is a secure multi-agent marketplace which supports a variety of types of transactions, from simple buying and selling of goods and services to complex multi-agent negotiation of contracts with temporal and precedence constraints [20]. The MAGNET agents are self-interested which attempt to gain the greatest possible profits from their endeavors. In this sense, MAGNET is more suitable for B2C eCommerce rather than B2B eCommerce where cooperative negotiation behaviour is

possible. A MAGNET agent can take one of two different roles: customer or supplier, and an agent can act as a customer in one negotiation and as a supplier in another negotiation process at the same time. To trade in the market, a customer agent generates a plan which is a collection of tasks with time and precedence constraints, and then submits one or more Requests for Quotes (RFQs) to suppliers via the MAGNET market. Any supplier agent who wants to bid will respond. After receiving the bids, the customer agent decides which bids to accept. Finally, the winning supplier agents execute the tasks included in their winning bids. The MAGNET market administrator mediates all communication among agents, and its trust model is somewhat different from other on-line auction system. Three standard cryptographic techniques are used: a publish/subscribe system to provide simple and general messaging, a time-release cryptography to provide guaranteed non-disclosure of the bids, and an anonymous communication system to hide the identity of the bidders until the end of the auction. As a whole, MAGNET is only an e-Auction mechanism with enhanced security facilities, whereas the negotiation system illustrated in this paper can perform automated multi-party multi-issue negotiations on behalf of individual organizations. In addition, the proposed dynamic business solution provides an open and scalable platform to automate and execute each crucial business process with reference to the BBT business model.

Rubenstein–Montano and Malaga have reported a Genetic Algorithm (GA)-based negotiation mechanism for searching optimal solutions for multi-party multi-objective negotiations [41]. Basically, a negotiation problem is treated as a multi-objective optimization problem. Apart from the standard genetic operators such as selection, crossover and mutation, the GA is enhanced with a new operator called *trade*. The trade operator simulates a concession making mechanism which is often used in negotiation systems. However, the main problem of their particular GA-based negotiation mechanism is that the preferences (i.e., the utility functions) of all the negotiation parties are assumed available to a central negotiation mechanism. Such an assumption does not correspond to the reality often found in B2B eCommerce. Our multi-party multi-issue negotiation mechanism discussed in this paper does not assume complete knowledge about a negotiation space. In other words, the preferences of the negotiation opponent (i.e., a business partner) are assumed unavailable to our negotiation agents. Nevertheless, the proposed sequential negotiation mechanism can still lead to Pareto optimal solutions if they exist in a negotiation space. Moreover, our automated negotiation service is based on the multi-agent system approach where each agent representing individual business can make autonomous negotiation decisions. Our automated negotiation mechanism represents a distributed decision making model when compared with the centralized decision making model presented in [41].

Genetic algorithm has also been applied to learn effective rules to support the negotiation processes [36]. A chromosome represents a negotiation (classification) rule rather than an offer. The fitness of a chromosome (a rule) is measured in terms of how many times the rule has contributed to reach an agreement. In order for the system to determine if an agreement is possible, each negotiator's preferences including the reservation values of the negotiation attributes are assumed known or hypothesized. Therefore, this approach also suffers from the same common problem found in the other negotiation models which assume complete information about negotiation spaces. Our proposed negotiation mechanism does not assume the availability of the opponents' negotiation preferences, and therefore it is more suitable for the development of an automated negotiation service for real-world B2B applications.

In many real-life negotiation settings such as business process management and eCommerce, negotiation agents have only limited information about their opponents and limited computational resources (bounded rationality) to deliberate solutions. Therefore, employing heuristic approach to develop negotiation agents' decision making mechanism is desirable because it is impossible to predict or specify equilibrium strategies at design time given the above constraints. A fuzzy similarity-based trade-off mechanism is proposed to search for near optimal negotiation solutions based on realistic assumptions [11]. Nevertheless, the fuzzy similarity trade-off mechanism is demonstrated based on a bilateral negotiation situation only, whereas negotiations in B2B eCommerce often involves multiple parties. The proposed negotiation supports multi-party multi-issue negotiations typically found in B2B environment. The negotiation research reported by Faratin et al. [11] focuses on a theoretical discussion and so a concrete system implementation is not available. However, the design, implementation, and evaluation of our business service discovery and negotiation system is discussed in this paper. In particular, our negotiation system is developed based on the emerging Web services standard which enables intra as well as inter-enterprises communication and transaction processing.

Defeasible logic has been used to model negotiation strategies based on argumentation semantics [9]. Different classes of arguments are identified based on the notions of strict arguments, defeasible arguments, and supportive arguments [13]. According to these notions, a negotiation agent can evaluate incoming arguments or generate new arguments with various strength with respect to the requirements of a particular negotiation situation. It is proposed that defeasible rules are not only suitable for specifying negotiation strategies but they are also useful for expressing offers and counter-offers exchanged among agents during a negotiation session. It is believed that by enabling agents to exchange rules, instead of just exchanging simple communication performatives, the flexibility of the proposed negotiation protocol can be improved considerably [9]. Nevertheless, how multi-agent multi-issue nego-

tiation is supported by the defeasible logic-based negotiation system is not clearly explained in the paper [9]. Moreover, the computational complexity of the defeasible negotiation system will be a major obstacle for its application to real-world B2B eCommerce applications. In fact, an implementation of the proposed defeasible negotiation framework is not yet available. In this paper, the design, implementation, and evaluation of the proposed negotiation system is reported. The effectiveness and the efficiency of the proposed automated negotiation mechanism are evaluated based on our empirical study.

Zeng and Sycara [54,55] have developed a sequential negotiation model called Bazaar which allows agents to revise their beliefs about the negotiation environment and the preferences of the opponents. The multi-agent learning mechanism in Bazaar is underpinned by a probabilistic framework which utilizes Bayesian representation and learning. Although Bayesian belief network was proposed to represent the uncertainty of a changing negotiation environment, only an example based on naive Bayesian probability for learning single negotiation issue was given. In their naive Bayesian learning model, it was assumed that the probability distribution  $\Pr(H_i)$  of the reservation price of the opponent was public information. Moreover, it was assumed that domain knowledge in the form of the conditional probabilities  $\Pr(o|H_i)$  describing the chance of an offering price given the opponent's reservation price was available to the system. Nevertheless, such assumptions are generally not valid in B2B eCommerce where each company tends to keep their preference information private. Furthermore, their experimentation and the corresponding illustration is limited to learning single negotiation issue (e.g., price) involving two parties only. The negotiation system discussed in this paper allows multiple agents to negotiate over multiple issues at the same time. In addition, our negotiation mechanism does not assume the availability of the opponents' preference information which tends to be kept private by individual business in B2B encounters. Even without such information, our negotiation mechanism can guarantee Pareto optimal results if solutions do exist in a negotiation space.

Other negotiation models such as case-based reasoning (CBR) [45], fuzzy constraint-based negotiation [23], belief revision [30] and argumentative logic [25] have also been explored to develop automated negotiation systems. However, these models either suffer from the lack of operational semantics to conduct multi-party multi-issue negotiations, computationally inefficient, or the lack of a concrete system design and implementation. The negotiation mechanism discussed in this paper is just able to alleviate all of the above problems.

### 3. System architecture

Web services are emerging to provide a systematic and extensible framework for application-to-application interaction, built on top of existing Web protocols and based



on open XML standards [4,47,19]. Major software vendors such as IBM, Microsoft, SAP, SUN and Oracle are all embracing Web services standards [7,49]. We leverage both Web services and intelligent agent techniques to develop the next generation of *dynamic* eBusiness systems. In fact, the design of our dynamic eBusiness solution is based on a sound B2B eCommerce model called Business-to-Business Transaction Model (BBT) [17].

With reference to the BBT model [17], B2B business process can be divided into six important stages. The first stage is “Partnership Formation” which involves finding the business partners that provide products or services in a supply chain as well as the forming of a virtual enterprise; the second stage is “Brokering” which is the process of matching sellers who supply goods or services to the buyers who require them; the third stage is “Negotiation” where the traders aim to reach an agreement about what actions should be performed under what conditions; the fourth stage “Contract Formation” marks the end of negotiation and involves the agreed terms being put into a legally binding contract; the fifth stage “Contract Fulfilment” refers to the parties executing the agreed transactions according to the terms specified in the contract; “Service Evaluation” is the final stage where traders evaluate their satisfaction with the transactions so as to prepare for another partnership formation in the future. The architecture of our Web Service Discovery and Negotiation System

(WSDNS) is depicted in Fig. 1. The proposed WSDNS system architecture directly supports “Partnership Formation”, “Brokering”, “Negotiation” and “Contract Formation”. Moreover, it can be easily extended to support the last two stages of the BBT model. The real-world scenario of applying the WSDNS platform to streamline B2B eCommerce can be illustrated as follows:

1. Service providers such as automobile manufacturers can employ their software agents (i.e., client applications) to publish their product/service information, application binding information and application interface specification on a directory server (Web Server 1 in Fig. 1) such as the Universal Description, Discovery and Integration (UDDI) Server [6,46]. The application interface specification and the application binding details are described by the universal Web Service Description Language (WSDL) [7].
2. When a wholesale car dealer want to purchase a batch of family cars from some automobile manufacturers (i.e., a B2B scenario), the car dealer can delegate this task to its intelligent software agent. The software agent (i.e., another client application) can query the UDDI Server to search for relevant products and to obtain the Web service invocation information. The car dealer can instantiate their software agent on a desktop computer or a mobile service. The software agents from the car

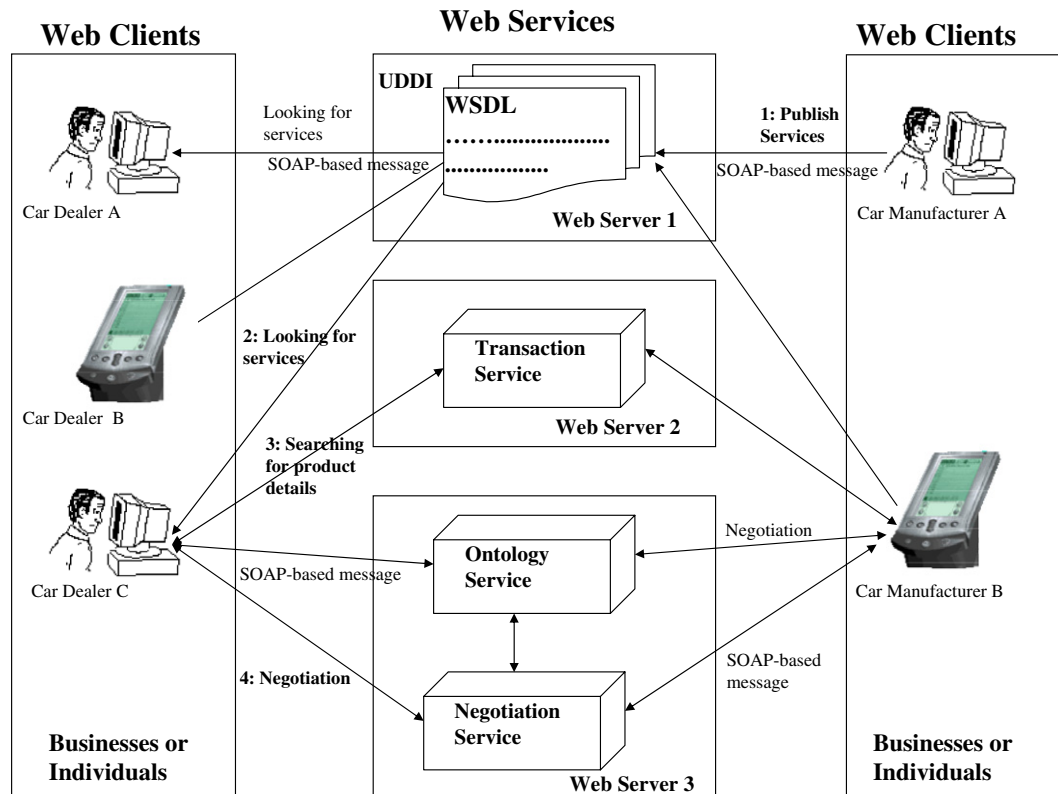


Fig. 1. A web service-based service discovery and negotiation system.

- dealers or the automobile manufacturers can connect to WSDNS via the standard XML-based Simple Object Access Protocol (SOAP) communication standard [6,3].
3. If the service interface of the transactional Web service is recognized by the software agent of the car dealer (i.e., the service consumer), connection to the transaction service (Web Server 2 in Fig. 1) can be established immediately between the car dealer and the automobile manufacturer (i.e., the service provider). If the Web service interface is of an unknown type to the service consumer's agent, adaptation of the service interface will take place before a connection is made. In this example, the software agent of the car dealer can examine the catalog of automobiles provided on the transaction server of the car manufacturer once a successful service binding is made.
  4. After obtaining relevant automobile information from the automobile manufacturer's Web site, the car dealer can start to negotiate with the manufacturer in terms of the price of the automobile, the delivery time of the automobiles, payment term, number of automobiles shipped, etc. Both the car dealer and the automobile manufacturer can obtain the service invocation information of the negotiation service via the UDDI server. After obtaining the relevant WSDL-based interface specification, the client applications representing the respective parties can consult the ontology server (Web Server 3 in Fig. 1) to share a common set of negotiation attributes. After such a hand-shaking process, the buyer and the seller can start to negotiate by instantiating the corresponding software agents on the negotiation server. Once a consensus is reached, the respective agents will alert their human users to confirm or reject the deal via the desktop machine or a mobile device.

The roles of the various components of WSDNS can be understood in terms of the traditional way of purchasing a car. The UDDI directory service [46] is like the yellow pages where one can locate a particular kind of business (e.g., wholesale car sales). WSDL is an analogy of a purchase order that specifies what information is required to complete an order, and SOAP is the protocol that defines a format for transmitting an electronic copy of the purchase order. With reference to the BBT model [17], Web Services provide the system standards and technology to support the stages of Partnership Formation and Brokering. The ontology service as well as the negotiation service (i.e., an e-marketplace) can be provided by the service provider such as the automobile manufacturer or a trusted third party such as a commercial bank. In either case, the negotiation service is independent of the transaction processing service to realize a highly distributed and scalable system architecture. As a result, the run-time performance of B2B transaction processing is optimized. Typically, the transaction service is provided by the seller to advertise product/service information and to support standard order processing function.

Based on the Web services standard, discovery and invocation of the negotiation service (i.e., entering the e-marketplace) is similar to that of the transaction service. After a buyer (seller) identifies some potential sellers (buyers), they can invite them to enter the e-marketplace with automated negotiation support. Once a negotiation service is successfully invoked, a *negotiation agent* will be instantiated on the negotiation server (i.e., the open e-marketplace) to conduct bargaining on behalf of the respective business (the negotiation stage of the BBT model). However, before negotiation can take place between a service consumer and a service provider, the ontology server must be consulted by all the negotiation parties to establish the common vocabulary (e.g., the attributes used to describe a car deal). In the proposed WSDNS system architecture, the negotiation server, the ontology server, and the transaction server are all implemented based on the Web services standards. Such an architecture allows service consumers and service providers coming from heterogeneous environments (e.g., various computer platforms and programming languages) to be able to communicate with each other based on the SOAP protocol. In fact, such an open system architecture is crucial for the success of B2B eCommerce because each company may operate under a quite different computing environment, and there is no way to enforce that they must use exactly the same computing platform and programming language.

On the negotiation server, a negotiation agent will inform their respective human negotiator to confirm a deal or to quit a negotiation session after an agreement is made or a timeout status is reached. If the human negotiators confirm the agreement, the Contract Formation stage can begin on the transaction server (Web server 2) where standard order processing function can be extended to lodge and archive business contracts. The current system architecture of WSDNS can also easily be extended to support the Contract Fulfilment stage and the Service Evaluation stage of the BBT model. For example, both car dealers (consumers) and automobile manufacturers (suppliers) can connect to the transaction server to track the status of purchase orders to verify if certain contacts are fulfilled. Moreover, service consumers can send their feedback about products and services to the transaction server for further analysis by the service suppliers (Service Evaluation). However, this paper will focus on the intelligent agents-based automated negotiation service.

Our prototype system [28] was developed using JWSDP – the Java Web Service Developer Pack provided by Sun Microsystems. In particular, the Java API for XML Processing (JAXP) and Java Architecture for XML Binding (JAXB) were used to manipulate the XML documents (e.g., the offers). The Java API for XML Messaging (JAXM) was used to send messages over the Internet in a standard format using the SOAP method calls. Essentially, this is how the offer information is communicated between our negotiation server and the client application programs. The Java API for XML Registries (JAXR) was used by a

client program to query a service registry to identify the appropriate transaction services or negotiation services.

#### 4. A Pareto optimal negotiation mechanism

A sequential alternate-offering negotiation protocol, a variant of the *monotonic concession protocol* [40], is adopted by the proposed negotiation agents. Automated negotiation proceeds in a discrete series of rounds as depicted in Fig. 2. In each round, each agent puts forward an offer in alternate. If these offers overlap, it means that an agreement is reached. If the offers do not overlap, negotiation proceeds to the next round where the agents make a concession. If there is no agreement after the deadline is reached, an agent decides to quit and the negotiation ends with a conflict. The optimal negotiation mechanism illustrated in this section is based on multi-attribute utility theory (MAUT) [22] and is first discussed in [1].

A negotiation space  $Neg = \langle P, A, D, U, T \rangle$  is a 5-tuple which consists of a finite set of negotiation parties (agents)  $P$ , a set of attributes (i.e., negotiation issues)  $A$  understood by all the parties  $p \in P$ , a set of attribute domains  $D$  for  $A$ , and a set of utility functions  $U$  with each function  $U_p^o \in U$  for an agent  $p \in P$ . An attribute domain is denoted  $D_{a_i}$  where  $D_{a_i} \in D$  and  $a_i \in A$ . An utility function pertaining to an agent  $p$  is defined by:  $U_p^o : D_{a_1} \times D_{a_2} \times \dots \times D_{a_n} \mapsto [0, 1]$ . Each agent  $p$  has a deadline  $t_p^d \in T$ . It is assumed that information about  $P, A, D$  is exchanged among the negotiation parties during the *ontology sharing* stage before negotiation actually takes place. A *multi-lateral* negotiation situation can be modelled as many one-to-one *bi-lateral* negotiations where an agent  $p$  maintains a separate negotiation dialog with each opponent. In a negotiation round, the agent will make an offer to each

of its opponents in turn, and consider the most favourable counter-offer from among the set of incoming offers evaluated according to its own payoff function  $U_p^o$ .

##### 4.1. Representing offers

An offer  $\vec{o} = \langle d_{a_1}, d_{a_2}, \dots, d_{a_n} \rangle$  is a  $n$ -tuple of attribute values (intervals) pertaining to a finite set of attributes  $A = \{a_1, a_2, \dots, a_n\}$ . An offer can also be viewed as a vector of attribute values in a geometric negotiation space with each dimension representing a negotiation issue. Each attribute  $a_i$  takes its value from the corresponding domain  $D_{a_i}$ . Generally speaking, a finite set of candidate offers  $O_p$  acceptable to an agent  $p$  (i.e., satisfying its hard constraints) is constructed via the Cartesian product  $D_{a_1} \times D_{a_2} \times \dots \times D_{a_n}$ . As human agents tend to specify their preferences in terms of a range of values, a more general representation of an offer is a tuple of attribute value intervals such as  $o_i = \langle 20 - 30 (\$), 1 - 2 (\text{years}), 10 - 30 (\text{days}), 100 - 500 (\text{units}) \rangle$ .

##### 4.2. Representing negotiation preferences

While offer representation deals with the issue of representing the content of the potential offers, preference representation is concerned about rating a set of potential offers according to an agent's specific negotiation interests. The *valuations* of individual attributes and attribute values (intervals) are defined by the valuation functions  $U_p^A : A \mapsto [0, 1]$  and  $U_p^{D_{a_i}} : D_{a_i} \mapsto [0, 1]$  respectively, whereas  $U_p^A$  is an agent  $p$ 's *valuation function* for each attribute  $a_i \in A$ , and  $U_p^{D_{a_i}}$  is an agent  $p$ 's valuation function for each attribute value  $d_{a_i} \in D_{a_i}$ . In addition, the valuations of attributes are assumed normalized, that is,  $\sum_{a_i \in A} U_p^A(a_i)$

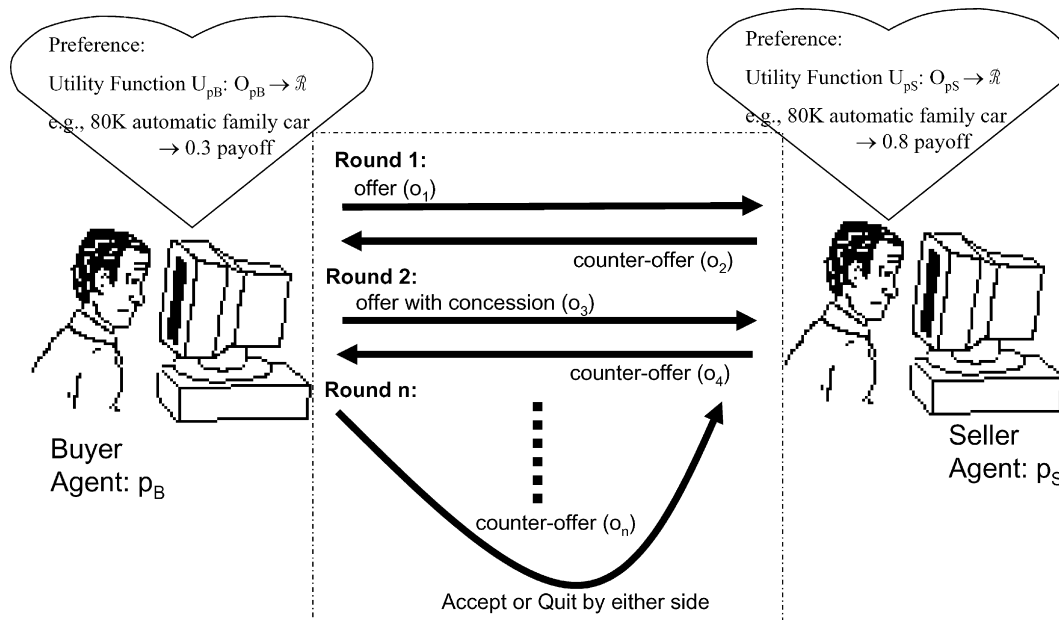


Fig. 2. An overview of the negotiation process.

= 1. One common way to quantify an agent’s preference (i.e., the utility function  $U_p^o$ ) for an offer  $o$  is by a linear aggregation of the valuations [1,44]:  $U_p^o(o) = \sum_{a_i \in A} U_p^A(a_i) \times U_p^{D_{a_i}}(d_{a_i})$ .

4.3. Computing concessions and generating offers

If an agent’s initial proposal is rejected by its opponent, it needs to propose an alternative offer with the least utility decrement (i.e., computing a concession). An agent will maintain a set  $O'_p$  which contains the offers it has proposed before or the offer to be proposed in the current round. In a negotiation round, an alternative offer with a concession can be determined based on:

$$\exists o_{\text{counter}} \in \{O_p - O'_p\} \forall o_x \in \{O_p - O'_p\} : [o_x \preceq_p o_{\text{counter}}]$$

$o_x \preceq_p o_y$  denotes that an offer  $o_y$  is more preferable than another offer  $o_x$ . The above statement defines that the alternative offer  $o_{\text{counter}}$  is the most preferable offer from among the set of feasible offers which have not been proposed before. The preference relation  $\preceq_p$  is a total ordering induced by an agent  $p$ ’s utility function  $U_p^o$  over the set of feasible offers  $O_p$ . In other words, the feasible offers of an agent  $p$  are ranked in descending order of utility driven by  $(\preceq_p, O_p)$ . An alternative offer with concession is picked up from the top of the list ranked by  $(\preceq_p, \{O_p - O'_p\})$  in a negotiation round.

4.4. Evaluating incoming offers

When an incoming offer  $o$  is received from an opponent, an agent  $p$  first evaluates if  $o \in O_p$  is true (i.e., the offer satisfying all its hard constraints). To do this, an *equivalent offer*  $o_{\simeq}$  should be computed.  $o_{\simeq}$  represents agent  $p$ ’s interpretation about the opponent’s proposal  $o$ . Once  $o_{\simeq}$  for  $o$  is computed, acceptance of the incoming offer  $o$  can be deter-

mined with respect to  $p$ ’s own preference  $(\preceq_p, O_p)$ . An offer  $o_{\simeq} \in O_p$  is equivalent to  $o$  iff every attribute interval of  $o_{\simeq}$  intersects each corresponding attribute interval of  $o$ . Formally, any two attribute intervals  $d_x, d_y$  intersect if the intersection of the corresponding sets of points is not empty (i.e.,  $\{d_x\} \cap \{d_y\} \neq \emptyset$ ). The acceptance criteria for an incoming offer  $o$  (i.e., the equivalent  $o_{\simeq}$ ) is defined by:

1. If  $\forall o_x \in O_p, o_x \preceq_p o_{\simeq}$ , an agent  $p$  should accept  $o$  since it produces the maximal payoff.
2. If  $o_{\simeq} \in O'_p$  is true, an agent  $p$  should accept  $o$  because  $o_{\simeq}$  is one of its previous proposals or is the one to be proposed in the current round.

It can be proved that if each participating agent  $p \in P$  employs their preference ordering  $(\preceq_p, O_p)$  to compute concessions and uses the offer acceptability criteria described above to evaluate incoming offers, *Pareto optimal* [40] result is always found if it does exist in a negotiation space.

**Definition 1.** Given a negotiation space  $\text{Neg} = \langle P, A, D, U, T \rangle$ , a negotiation solution  $o_x$  is a Pareto optimal solution if it is impossible to find another solution  $o_y$  such that at least one agent will be better off ( $\exists p \in P : U_p^o(o_y) > U_p^o(o_x)$ ) but no other agent will be worse off ( $\neg \exists p' \in \{P - p\} : U_{p'}^o(o_y) < U_{p'}^o(o_x)$ ).

It should be noted that a Pareto optimal solution does not necessarily lead to maximal joint payoff. Joint payoff is defined as the sum of each agent’s payoff obtained at the end of a negotiation session. Fig. 3 depicts how a Pareto optimal result is derived according to our negotiation model. With reference to the negotiation example demonstrated in Fig. 3, the decision making processes of agents  $p_B$  and  $p_S$  are illustrated in Table 1. The aforementioned sequential negotiation mechanism can be expressed by Theorem 1.

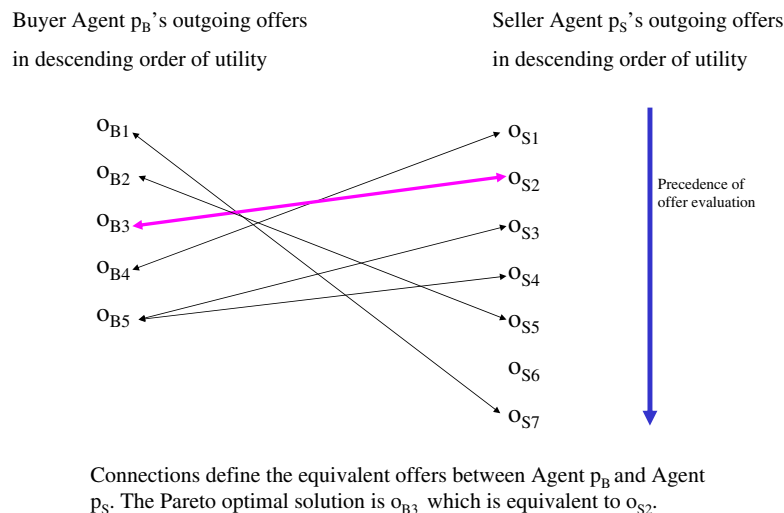


Fig. 3. Deriving a Pareto optimal solution.



Table 1  
The decision making processes of agents

Buyer agent $p_B$	Seller agent $p_S$
1 Generate offer: $O'_{p_B} = \{o_{B1}\}$ Send out offer $o_{B1}$ Evaluate $o_{S1} \cap O'_{p_B} = \emptyset$	Generate offer: $O'_{p_S} = \{o_{S1}\}$ Send out offer $o_{S1}$ Evaluate $o_{B1} \cap O'_{p_S} = \emptyset$
2 Generate offer: $O'_{p_B} = \{o_{B1}, o_{B2}\}$ Send out offer $o_{B2}$ Evaluate $o_{S2} \cap O'_{p_B} = \emptyset$	Generate offer: $O'_{p_S} = \{o_{S1}, o_{S2}\}$ Send out offer $o_{S2}$ Evaluate $o_{B2} \cap O'_{p_S} = \emptyset$
3 Generate offer: $O'_{p_B} = \{o_{B1}, o_{B2}, o_{B3}\}$ Send out offer $o_{B3}$ Evaluate $o_{S3} \cap O'_{p_B} = \emptyset$	Generate offer: $O'_{p_S} = \{o_{S1}, o_{S2}, o_{S3}\}$ Send out offer $o_{S3}$ Evaluate $o_{B3} \cap O'_{p_S} \neq \emptyset$ Agent $p_S$ accepts offer $o_{B3}$

**Theorem 1.** Given a negotiation space  $Neg$ , if each agent  $p \in P$  employs the utility function  $U_p^o$  to rank its offers  $o \in O_p$  in descending order of utilities and then sends out offers strictly according to the ordering  $(\succeq_p, O_p)$ , the first overlapping offer among the agents is a Pareto optimum.

A formal proof of the Pareto optimal feature of the sequential negotiation method can be conducted as follows:

**Proof.** Assuming that the sequential negotiation method discovers the first overlapping offer (i.e., a solution)  $(o_t, o_u, \dots, o_z)$  where offers  $o_t, o_u, \dots, o_z$  are equivalent offers from agents  $p_t, p_u, \dots, p_z$ 's perspectives. If the solution  $(o_t, o_u, \dots, o_z)$  is not Pareto optimal and there is another deal  $(o_i, o_j, \dots, o_m)$  which is Pareto optimal, this Pareto optimal deal must satisfy the condition that  $(U_{p_t}^o(o_i) \leq U_{p_t}^o(o_t)) \wedge (U_{p_u}^o(o_j) \leq U_{p_u}^o(o_u)), \dots, \wedge (U_{p_z}^o(o_m) \leq U_{p_z}^o(o_z))$  is true because the offers are evaluated in descending order of utilities by each agent  $p_t, p_u, \dots, p_z \in P$ . However, the deal  $(o_i, o_j, \dots, o_m)$  cannot be a Pareto optimum according to Definition 1.  $\square$

## 5. The experiments

The simulated negotiation environment of our experiments was characterized by multi-lateral negotiations among two buyer agents ( $B1, B2$ ) and two seller agents ( $S1, S2$ ). These agents negotiated over some virtual services or products described by some attributes with each attribute domain containing discrete values such as some natural numbers. The valuation of an attribute or a discrete attribute value fell in the interval of  $(0, 1]$ . Each negotiation case was developed based on the valuation functions  $U_p^A, U_p^{D_{a1}}, U_p^{D_{a2}}, \dots, U_p^{D_{an}}$ , etc. defined for each agent  $p$  participating in the negotiation process. Each buyer (seller) participating in a negotiation process was assumed to have a product to buy (sell). For each negotiation case, an agreement zone always existed since the difference between a buyer and a seller only lay on their valuations against the same set of negotiation issues (e.g., attributes and attribute values).

As described in Section 4, the alternate offering protocol was adopted in our system. At the beginning of every negotiation round, each agent would invoke its own decision making mechanism to generate an offer for that round. The order of deliberation of out-going offers among the agents was randomly chosen by the simulation controller in each negotiation round. At the message exchange phase, each agent sent the offer messages to its opponents (e.g.,  $S1 \rightarrow B1$  and  $S1 \rightarrow B2$ ). After the message exchange phase, the simulation controller randomly selected a sequence of offer evaluation such as  $\langle B2, B1, S1, S2 \rangle$ . Then, each agent in turn followed the pre-defined order to evaluate its incoming offers. For instance, with reference to the above order, agent  $B2$  would evaluate its incoming offers sent from  $S1$  and  $S2$ , then agent  $B1$  would evaluate its incoming offers. Each agent selected the best incoming offer (evaluated according to its private utility function) as the opponent offer  $o_{\text{opponent}}$  in a negotiation round. If there was a tie, an opponent would be selected randomly by an agent. If an agreement was made between a pair, they would be removed from the negotiation table by the simulation controller, and the remaining two agents would continue their negotiation until either an agreement was made or the deadline was due. Our negotiation server was installed on a PC with a Pentium-4 2.2 GHz single processor and 512 MB main memory. To avoid the communication overheads, all the experiments were conducted under an Intranet environment. Figs. 4 and 5 show the client interfaces of a buyer and a seller respectively.

### 5.1. Experiment one

The objective of our first experiment was to evaluate the general performance of the agent-based negotiation system. Both the effectiveness (in terms of joint payoff) and the efficiency (in terms of execution time in seconds) of the system were evaluated. These agents negotiated over some virtual services or products described by five attributes (i.e.,  $|A| = 5$ ) with each attribute domain containing 5 discrete values represented by the natural numbers  $D_a = \{1, 2, 3, 4, 5\}$ . For each agent  $p$ , the size of the feasible offer set is:  $|O_p| = 5^5 = 3,125$ . There were six negotiation groups with each group containing 10 negotiation cases; each case was uniquely defined by the valuation functions  $U_p^A, U_p^{D_{a1}}, U_p^{D_{a2}}, \dots, U_p^{D_{an}}$ , etc. for each participating agent  $p$ . For the first simulation group, each negotiation case contained identical buyer/seller preferences (i.e., the same weights for the attributes and the same valuations against the same set of attribute values). Each case in the succeeding group was injected a 20% increment of preferential difference between a buyer agent and a seller agent. The negotiation deadline was set to 3125 rounds in this experiment. If an agreement was not reached after the deadline was due, zero utility was assumed for each agent.

The experimental results are depicted in Table 2. The average execution time and joint payoffs of each simulated negotiation group is plotted in Fig. 6. The execution time

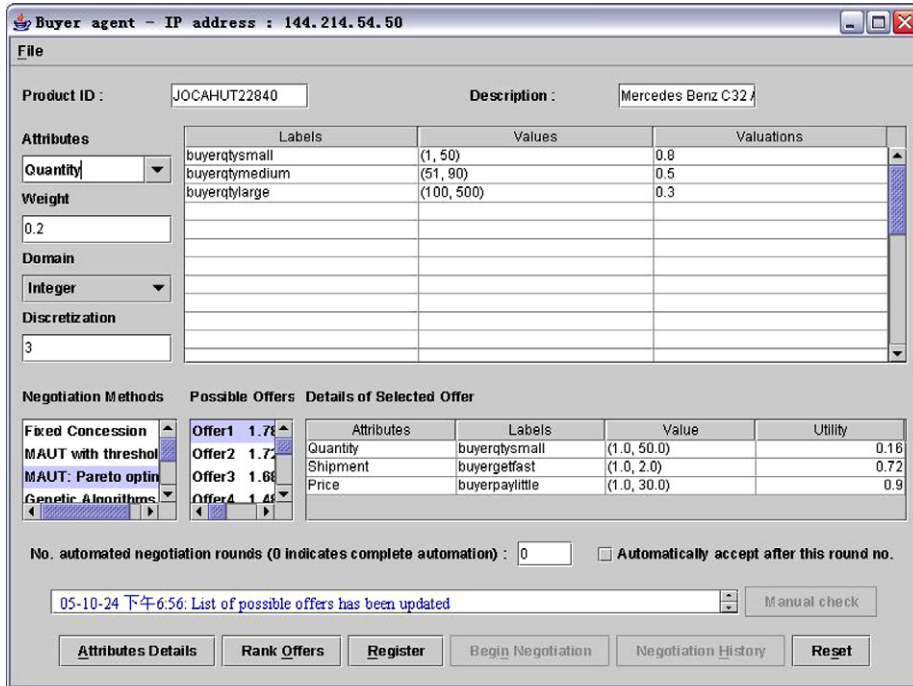


Fig. 4. The client interface of a buyer.

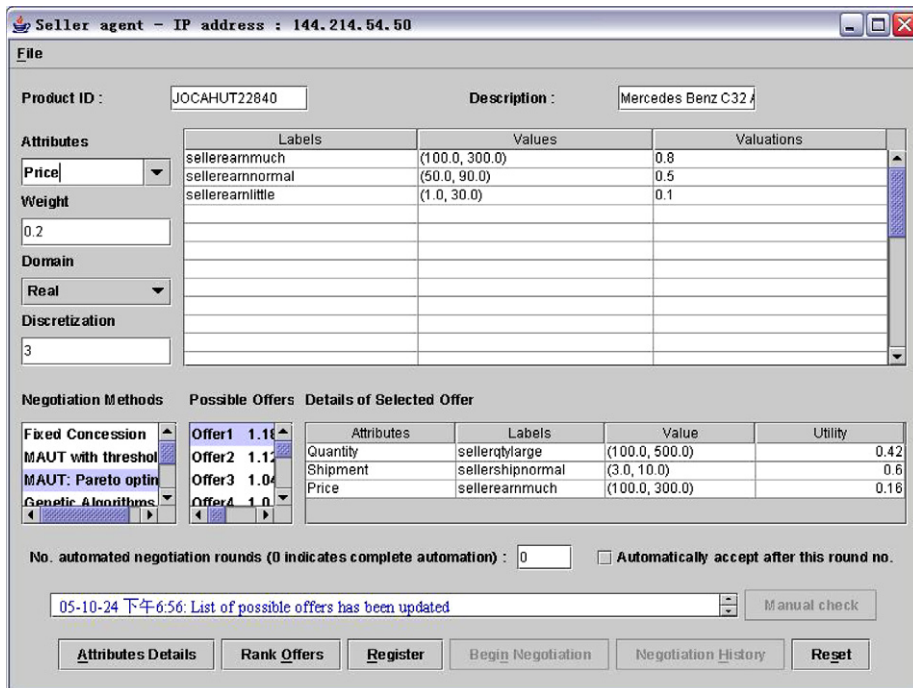


Fig. 5. The client interface of a seller.

of a negotiation case represents the time sent by the negotiation server to identify a solution; it was counted after every client had submitted their negotiation preferences to the server (i.e., after every user clicked the “Begin Negotiation” button). Since the buyer/seller agents in the first simulation group were characterized by the same valuation function, they were able to find agreements in the first

negotiation round. Therefore, the execution time of this simulation group is the fastest among the other simulation groups. It should be noted that the execution time of our agent-based negotiation system only grew linearly with respect to the increased conflicts (i.e., preferential differences) between the buyer agents and the seller agents. It was also found that the automated negotiation system

Table 2  
General performance of agent-based negotiation system

Group	Preferential difference (%)	Average joint utility	Average negotiation time (s)
1	0	2.30	1.13
2	20	2.87	2.31
3	40	2.88	2.67
4	60	2.41	3.12
5	80	2.56	3.13
6	100	1.93	3.22
Average		2.49	2.60

was able to locate the Pareto optimal negotiation solution in each negotiation case for all the simulation groups. In other words, optimal joint utility was produced by the system in the sense that it was impossible to find an improvement (in terms of joint utility) without any one of the four agents begin suffered from decreased individual utility.

5.2. Experiment two

Since our general hypothesis is that intelligent negotiation agents can help human negotiators make better deals faster, the second experiment was designed to test if the proposed intelligent negotiation agents can identify negotiation solutions more effectively (i.e., achieving higher payoffs) and more efficiently (i.e., reaching an agreement with fewer negotiation rounds) when compared with their human counterparts. The subjects involved in this experi-

ment are 20 postgraduate students. These subjects were randomly chosen and they attended lectures in automated negotiation for 2 weeks before the experiment began. The subjects were divided into five groups with each group comprising of two buyers and two sellers. The same negotiation protocol used in experiment one was adopted for this experiment. In each negotiation round, each human negotiator delivered sealed-bid to the opponents only. The order of evaluating incoming offers by the subjects (negotiators) was determined by lottery that was commonly used in playing card games. If a subject found that the incoming offer was acceptable, a deal was made and the number of negotiation rounds for reaching the agreement and the payoff obtained would be recorded for both the buyer and the seller. The remaining pair in a group would continue to negotiate until either an agreement was found or the deadline exceeded.

As the negotiation space of experiment one was too complicated for the average human negotiators, a smaller negotiation space ( $|A| = 4$  and  $|D_a| = 3$ ) was created. Five negotiation cases were developed with each one simulating the trading of some products such as computer, car, property, etc. As in experiment one, the valuation functions  $U_p^A$ ,  $U_p^{D_{a1}}$ ,  $U_p^{D_{a2}}, \dots, U_p^{D_{an}}$ , etc. were created for each agent  $p$  in a simulated case. Tables 4 and 5 depicts an example of the valuation functions for the buyer (B1) and the seller (S1) respectively. It should be noted that each subject knew only their own preference since they were not allowed to

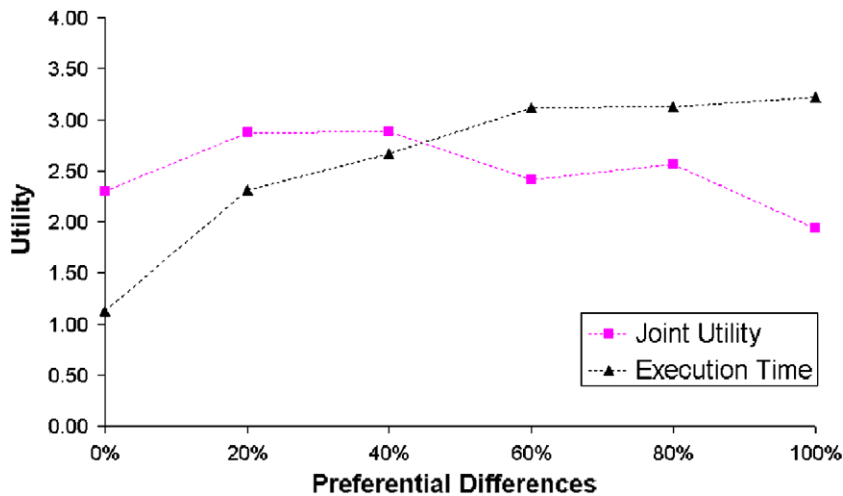


Fig. 6. Average execution time and joint payoff of simulated negotiations.

Table 3  
Comparative negotiation performance (Agent) vs. (Human)

Negotiation case	(Agent)		(Human)		Comparison	
	Joint payoff	Time (Rounds)	Joint payoff	Time (Rounds)	$\Delta$ Payoff	$\Delta$ Time
1	2.53	15	2.02	22	+0.51	-7
2	3.06	8	1.98	25	+1.08	-17
3	3.11	12	1.68	16	+1.43	-4
4	2.17	14	1.89	33	+0.28	-19
5	1.98	9	1.33	24	+0.65	-15

Table 4  
Valuation functions for attributes

Attribute	$U_{B1}^A$	$U_{S1}^A$
Delivery time	0.2	0.3
Warranty period	0.3	0.2
Payment method	0.2	0.1
Price	0.3	0.4

Table 5  
Valuation functions for the domain – warranty period

$D_{\text{warranty} - \text{period}}$	$U_{B1}^{D_w}$	$U_{S1}^{D_w}$
1 Year	0.1	0.9
3 Years	0.8	0.2
5 Years	0.9	0.1

exchange the preference information. The deadline was set to 81 negotiation rounds in this experiment. Each human negotiation group was randomly assigned to deal with one of the five simulated negotiation cases. The same set of negotiation cases was tried by the intelligent negotiation agents. The joint-payoff and the negotiation time (in rounds) obtained by the agent-based negotiation system (Agent) for each simulated negotiation case are depicted under the (Agent) columns in Table 3. Similarly, the joint-payoff and the negotiation time achieved by the human negotiators are depicted under the (Human) columns in Table 3. From Table 3, it is clear that the intelligent negotiation agents are able to produce a higher average joint payoff and consume fewer number of negotiation rounds in each negotiation case when compared with their human counterparts. Therefore, according to this study, our general hypothesis is supported in the sense that the agent-based negotiation system can identify better deals faster when compared with human negotiators.

## 6. Discussion

According to the results of our empirical test, the proposed negotiation agents can achieve Pareto optimal outcomes in each negotiation case. The efficiency of these negotiation agents are also satisfactory since they only take 2.6 s on average to reach a Pareto optimal agreement for a moderately complex negotiation session. When compared with other agent-based multi-party multi-issue negotiation systems such as Kasbah [35], the distinct advantage our negotiation system is that each negotiation agent can identify the Pareto optimal outcome which naturally leads to a win-win situation in the e-Marketplace. Such a feature is desirable in B2B eCommerce situations where mutual benefits is the basis for long-term partnership. Another observation is that the negotiation time only grows linearly with respect to the increasing complexity of a negotiation session (e.g., the increasing conflicts between a buyer and a seller). Such a feature is essential to develop scalable automated negotiation system for eBusiness.

The average joint utility reported in Table 2 is computed based on a linear utility aggregation function often adopted in automated negotiation research [1,36,44]. The limitation is that the dependencies among negotiation issues (if they exist) may not be properly captured by the linear utility function. For example, while a buyer may prefer cheaper price, they may be satisfied with an expensive product if extended warranty period is available. Such a preference relation may not be properly represented by the linear combination of the valuations of the particular purchase price and the warranty period. There was an attempt to employ non-linear utility function to estimate negotiators' payoffs [5]. Nevertheless, the merits of such a non-linear utility function were not empirically tested [5]. Further study can be performed to utilize various classes of utility functions to compute an agent's payoff and empirically evaluate the effectiveness of each class of utility functions. To make a better trade-off between computational complexity and computational efficiency for the development of practical negotiation systems to streamline eCommerce, our current approach is to make use of the efficient and widely used linear utility function [35,41,54,55] to compute an agent's payoff.

Even though computers are more efficient in carrying out numerical computations in general, they may not be as effective and efficient as human in approximate reasoning which is applied to solve real-world negotiation problems. Therefore, it is important to compare the performance of our negotiation agents with that of their human counterparts. In fact, previous research in agent-based negotiation [5,35,41,54,55] has done little in directly comparing the performance of automated negotiation systems with that of human negotiators. Furthermore, from a previous empirical study in negotiation support system, it was found that comprehensive automated negotiation support system did not reduce negotiation time when compared with that of a primitive decision support system for negotiations [8]. Therefore, our second experiment is important. According to our findings, the negotiation agents performed much better than their human counterparts. For instance, the mean and the standard deviation of the negotiation time (rounds) consumed by the agents are 11.6 and 3.05, respectively, in our second experiment. On the other hand, the mean negotiation time and the standard deviation of the human negotiators were 24 and 6.12, respectively. Such a difference is statistically significant (e.g.,  $p < 0.01$ ). Apart from achieving better performance, the negotiation agents' performance was more consistent (e.g., a much lower STD). However, there are limitations of our second experiment. Since the human negotiators are only college students receiving basic training in automated negotiation, they may not perform as well as the experienced human negotiators. In addition, the second experiment was structured as a game with the goal of optimizing one's self payoff. Therefore, the human subjects might tend to spend more time to exploit the negotiation space. Further experiments involving experienced human



negotiators can help improve the internal validity of our research.

## 7. Conclusions

Negotiation is one of the crucial stages with reference to the BBT business model. Intelligent agent techniques and Web services technology can be applied to develop the next generation of dynamic eBusiness systems in general and automated negotiation systems in particular. Since B2B negotiations are characterized by combinatorially complex negotiation spaces, tough negotiation deadlines and limited information about the opponents, practical negotiation mechanisms must be able to address these issues. The proposed Web services and intelligent agents-based negotiation system fulfils most of the requirements of practical negotiation systems for B2B eCommerce because it supports multi-party multi-issue negotiations based on a distributive decision making model, and it can deliberate Pareto optimal negotiation solutions with incomplete information about the opponents. Our experiments show that the intelligent negotiation agents outperform their human counterparts in terms of increased joint payoff and reduced negotiation time. Our research work opens the door to the development of intelligent system solutions to streamline B2B eCommerce. Future work includes the improvement of the adaptiveness of our intelligent agents so that they can take into account the changing negotiation behaviour of their negotiation opponents during bargaining. Moreover, non-linear utility functions can be explored to see if they can improve the effectiveness of automated negotiation systems.

## Acknowledgements

The author thanks the track chairs, anonymous reviewers, and audience at our presentation at the 2005 International Conference on Electronic Commerce, Xi'an, China, for helpful input on an earlier version of this work. We would also like to acknowledge Zhangxi Lin and Karl Lang, co-editors of the special issue of eCommerce Research and Applications, and three anonymous reviewers for valuable comments.

## References

- [1] Mihai Barbuceanu, Wai-Kau Lo, Multi-attribute utility theoretic negotiation for electronic commerce, in: Frank Dignum, Ulises Cortés (Eds.), *Agent-Mediated Electronic Commerce III – Current Issues in Agent Based Electronic Commerce Systems*, Number 2003 in Lecture Notes in Artificial Intelligence, Springer-Verlag, Heidelberg, Germany, 2001, pp. 15–30.
- [2] M. Bazerman, T. Magliozzi, M. Neale, Integrative bargaining in a competitive market, *Organization Behavior and Human Decision Processes* 35 (1985) 294–313.
- [3] D. Box, Simple Object Access Protocol (SOAP) 1.1, May 2000. Available from <http://www.w3.org/TR/SOAP/>.
- [4] Minder Chen, Andrew N.K. Chen, Benjamin B.M. Shao, The implications and impacts of web services to electronic commerce research and practices, *Journal of Electronic Commerce Research* 4 (4) (2003) 128–139.
- [5] Samuel P.M. Choi, Jiming Liu, Sheung-Ping Chan, A genetic agent-based negotiation system, *Computer Networks* 37 (2) (2001) 195–204.
- [6] Francisco Curbera, Matthew Duftler, Rania Khalaf, William Nagy, Nirmal Mukhi, Sanjiva Weerawarana, Unraveling the web services web: an introduction to SOAP, WSDL, and UDDI, *IEEE Internet Computing* 6 (2) (2002) 86–93.
- [7] N. Davies, D. Fensel, M. Richardson, The future of web services, *BT Technology Journal* 22 (1) (2004) 118–130.
- [8] M. Delaney, A. Froughi, W. Perkins, An empirical study of the efficacy of a computerized negotiation support system (nss), *Decision Support Systems* 20 (1) (1997) 53–64.
- [9] M. Dumas, G. Governatori, A.H.M. ter Hofstede, P. Oaks, A formal approach to negotiating agents development, *Electronic Commerce Research and Applications* 1 (2) (2002) 29–50.
- [10] H. Ehtamo, E. Ketteunen, R. Hämäläinen, Searching for joint gains in multi-party negotiations, *European Journal of Operational Research* 130 (2001) 54–69.
- [11] P. Faratin, C. Sierra, N.R. Jennings, Using similarity criteria to make issue trade-offs in automated negotiations, *Artificial Intelligence* 142 (2) (2002) 205–237.
- [12] S. Fatima, M. Wooldridge, N.R. Jennings, An agenda based framework for multi-issues negotiation, *Artificial Intelligence* 152 (1) (2004) 1–45.
- [13] Guido Governatori, Michael Maher, An argumentation-theoretic characterization of defeasible logic, in: Werner Horn (Ed.), *Proceedings of the 14th European Conference on Artificial Intelligence*, IOS Press, Amsterdam, the Netherlands, 2000, pp. 469–474.
- [14] R. Guttman, P. Maes, Agent-mediated integrative negotiation for retail electronic commerce, in: Pablo Noriega, Charles Sierra (Eds.), *Proceedings of the First International Workshop on Agent Mediated Electronic Commerce (AMET-98)*, LNAI, vol. 1571, Springer, 1999, pp. 70–90.
- [15] R. Guttman, A. Moukas, P. Maes, Agent-mediated electronic commerce: a survey, *Knowledge Engineering Review* 13 (2) (1998) 147–159.
- [16] J. Harsanyi, R. Selten, A generalised nash solution for two-person bargaining games with incomplete information, *Management Science* 18 (5) (1972) 80–106.
- [17] M. He, N.R. Jennings, H. Leung, On agent-mediated electronic commerce, *IEEE Transactions on Knowledge and Data Engineering* 15 (4) (2003) 985–1003.
- [18] P. Heiskanen, Decentralized method for computing Pareto solutions in multiparty negotiations, *European Journal of Operational Research* 117 (1999) 578–590.
- [19] Ying Huang, Jen-Yao Chung, A web services-based framework for business integration solutions, *Electronic Commerce Research and Applications* 2 (1) (2003) 15–26.
- [20] A. Jaiswal, Y. Kim, M. Gini, Design and implementation of a secure multi-agent marketplace, *Electronic Commerce Research and Applications* 3 (4) (2004) 355–368.
- [21] N.R. Jennings, P. Faratin, A.R. Lomuscio, S. Parsons, C. Sierra, M. Wooldridge, Automated negotiation: prospects, methods and challenges, *Journal of Group Decision and Negotiation* 10 (2) (2001) 199–215.
- [22] R. Keeney, H. Raiffa, *Decisions with Multiple Objectives: Preferences and Value Tradeoffs*, Cambridge University Press, Cambridge, UK, 1993.
- [23] Ryszard Kowalczyk, Van Bui, On constraint-based reasoning in e-negotiation agents, in: Frank Dignum, Ulises Cortés (Eds.), *Agent-Mediated Electronic Commerce III – Current Issues in Agent Based Electronic Commerce Systems*, Number 2003 in Lecture Notes in Artificial Intelligence, Springer-Verlag, 2001, pp. 31–46.
- [24] Sarit Kraus, Negotiation and cooperation in multi-agent environments, *Artificial Intelligence* 94 (1–2) (1997) 79–97.
- [25] Sarit Kraus, Katia Sycara, Amir Evenchik, Reaching agreements through argumentation: a logical model and implementation, *Artificial Intelligence* 104 (1–2) (1998) 1–69.



- [26] P. Krause, S. Ambler, M. Elvang-Goransson, J. Fox, A logic of argumentation for reasoning under uncertainty, *Computational Intelligence* 11 (1995) 113–131.
- [27] R. Krovi, A. Graesser, W. Pracht, Agent behaviors in virtual negotiation environments, *IEEE Transactions on Systems, Man, and Cybernetics* 29 (1) (1999) 15–25.
- [28] R.Y.K. Lau, Adaptive Negotiation Agents for e-Business, in: Proceedings of the Seventh International Conference on e-Commerce (ICEC'05), ACM Press, Xi'an, China, 2005, pp. 271–278.
- [29] R.Y.K. Lau, S.Y. Chan, Towards belief revision logic-based adaptive and persuasive negotiation agents, in: J.G. Carbonell, J. Siekmann (Eds.), Proceedings of the Eighth Pacific Rim International Conference on Artificial Intelligence (PRICAI'04), Lecture Notes in Artificial Intelligence, vol. 3157, Springer, Auckland, New Zealand, 2004, pp. 605–614.
- [30] R.Y.K. Lau, B. Essam, S.Y. Chan, Z. Huang, Belief revision for adaptive negotiation agents, in: Proceedings of the Third IEEE/WIC International Conference on Intelligent Agent Technology, IEEE Computer Society, Halifax, Canada, 2003, pp. 196–202.
- [31] R.Y.K. Lau, M. Tang, O. Wong, Towards genetically optimised responsive negotiation agents, in: Proceedings of the Fourth IEEE/WIC International Conference on Intelligent Agent Technology, IEEE Computer Society, Beijing, China, 2004, pp. 295–301.
- [32] F. Leymann, D. Roller, M.-T. Schmidt, Web services and business process management, *IBM Systems Journal* 41 (2) (2002) 198–211.
- [33] A.R. Lomuscio, N.R. Jennings, A classification scheme for negotiation in electronic commerce, *Journal of Group Decision and Negotiation* 12 (1) (2003) 31–56.
- [34] M. Ma, Agents in eCommerce, *Communications of the ACM* 42 (3) (1999) 79–80.
- [35] P. Maes, R. Guttman, A. Moukas, Agents that buy and sell, *Communications of the ACM* 42 (3) (1999) 81–91.
- [36] S. Matwin, T. Szapiro, K. Haigh, Genetic algorithm approach to a negotiation support system, *IEEE Transactions on Systems Man and Cybernetics* 21 (1) (1991) 102–114.
- [37] J.F. Nash, The bargaining problem, *Econometrica* 18 (1950) 155–162.
- [38] D. Pruitt, *Negotiation Behaviour*, Academic Press, 1981.
- [39] H. Raiffa, *The Art and Science of Negotiation*, Harvard University Press, 1982.
- [40] J. Rosenschein, G. Zlotkin, Task oriented domains, in: *Rules of Encounter: Designing Conventions for Automated Negotiation among Computers*, MIT Press, Cambridge, Massachusetts, 1994, pp. 29–52.
- [41] Bonnie Rubenstein-Montano, Ross A. Malaga, A weighted sum genetic algorithm to support multiple-party multi-objective negotiations, *IEEE Transactions on Evolutionary Computation* 6 (4) (2002) 366–377.
- [42] A. Rubinstein, Perfect equilibrium in a bargaining model, *Econometrica* 50 (1) (1982) 97–109.
- [43] M. Shaw, Electronic commerce: state of the art, in: M. Shaw, R. Banning, T. Strader, A. Whinston (Eds.), *Handbook on Electronic Commerce*, Springer, 2000, pp. 3–24.
- [44] C. Sierra, P. Faratin, N.R. Jennings, Deliberative automated negotiators using fuzzy similarities, in: Proceedings of the EUSFLAT-ESTYLF Joint Conference on Fuzzy Logic, Palma de Mallorca, Spain, 1999, pp. 155–158.
- [45] K. Sycara, Multi-agent compromise via negotiation, in: L. Gasser, M. Huhns (Eds.), *Distributed Artificial Intelligence II*, Morgan Kaufman, 1989, pp. 119–139.
- [46] Paul Tremblett, Java and UDDI registries, *Dr. Dobb's Journal of Software Tools* 27 (9) (2002) 34, 37–40.
- [47] Willem-Jan van den Heuvel, Zakaria Maamar, Moving toward a framework to compose intelligent web services, *Communications of the ACM* 46 (10) (2003) 103–109.
- [48] J. von Neumann, O. Morgenstern, *The Theory of Games and Economic Behaviour*, Princeton University Press, 1994.
- [49] W3C. Web Services Activity, 2004. Available from <http://www.w3.org/TR/2004/>.
- [50] J. Wilkenfeld, S. Kraus, K. Holley, M. Harr, Genie: a decision support system for crisis negotiations, *Decision Support Systems* 14 (4) (1995) 369–391.
- [51] M. Wooldridge, N. Jennings, Intelligent agents: theory and practice, *Knowledge Engineering Review* 10 (2) (1995) 115–152.
- [52] P. Wurman, M. Wellman, W. Walsh, The Michigan Internet AuctionBot: a configurable auction server for human and software agents, in: Katia P. Sycara, Michael Wooldridge (Eds.), Proceedings of the Second International Conference on Autonomous Agents (Agents'98), ACM Press, New York, 1998, pp. 301–308.
- [53] Y. Ye, J. Liu, A. Moukas, Agents in electronic commerce, *Electronic Commerce Research* 1 (1-2) (2001) 9–14.
- [54] D. Zeng, K. Sycara, Bayesian learning in negotiation, *International Journal of Human-Computer Studies* 48 (1) (1998) 125–141.
- [55] Dajun Zeng, Katia Sycara, Benefits of learning in negotiation, in: Proceedings of the 14th National Conference on Artificial Intelligence and 9th Innovative Applications of Artificial Intelligence Conference (AAAI-97/IAAI-97), AAAI Press, Menlo Park, 1997, pp. 36–42.