4

Developing a Multi-Issue E-Negotiation System for E-Commerce with JADE

Bala M. Balachandran
Faculty of Information Sciences and Engineering
University of Canberra
Australia

1. Introduction

The rapid growth of the Internet, networking systems, and wireless and web technologies is stimulating more and more companies to provide e-commerce applications. E-commerce application developers have the problem of creating enterprise-wide and world-wide applications that must operate across corporations and continents of the world. Such e-commerce applications are difficult to produce with traditional software technologies. An agent is a software entity that applies Artificial Intelligence techniques to choose the best of actions to perform in order to reach a goal specified by the user. It should be proactive, flexible, dynamic and autonomous, and should act in an intelligent manner to the changes produced in its environment. A multi-agent system (MAS) may be defined as a collection of agents that communicate between themselves to coordinate their activities in order to be able to solve collectively a problem that could not be solved by an individual agent (Sycara, 1998; Wooldridge, 2010). Multi-agent systems have been shown to be suitable to domains with the following characteristics (Moreno and Isern, 2002):

- Knowledge is distributed in different locations.
- Several entities have to join their problem-solving abilities to be able to solve a complex problem.
- The problem in the domain may be decomposed into a set of different sub-problems, even if they have some kind of inter-dependencies.

There has been growing interest in the applications of agent and multi-agent systems to problems in the e-commerce domain (Paprzycki, 2004; Fasli, 2007; Balachandran, 2010). The characteristics and abilities of software agents have been found ideal to model problems in the e-commerce field. For example, intelligent agents can support buying and selling products and other services over the Internet for their customers. Other potential applications of agents in e-commerce include managing supply chains, conducting technical and market research, locating potential partners, negotiation, auctions, bargaining, voting protocols and coalition formation. Such agent-based e-commerce systems will deliver huge benefits to the businesses and to their customers.

In an e-commerce environment, agents should work collaboratively, resolve conflicts and come to a common conclusion to a shared problem. The agents can be friendly (in which they work together in a team), or they could be opposed to each other (in which they work against each other). Either way, multi-agent systems highlight the social complexities and capabilities of software agents (Alem et al., 2002). When considering an agent-oriented view of e-commerce, it becomes evident that most problems involve multiple agents. Having multiple agents in a system increases its complexity because of the need for the agents to interact with each other and collaborate effectively in the common environment. These interactions can be simple such as those made between a client and a web server, or they can be composed of rich social interactions such communication, collaboration and negotiation as required (Paprzycki et al, 2004).

One of the key problems in implementing an agent-based e-commerce system is the question of what agents are necessary, how they should communicate, what strategy is required, and how it should be implemented. In this chapter, we have attempted to address these questions in the context of a multi-agent system for providing personalised buying and selling services. This chapter is organised as follows. First, we review agent-based negotiation and some related works. We then present our proposed negotiation model, discussing its ability to handle customer preferences based on multiple issues. We describe the model in terms of the negotiation object, the negotiation protocol and the negotiation strategy. Then we describe details of a prototype system we have developed using JADE and the Eclipse platform. Finally, we present our concluding remarks and discuss our future work.

2. Agent-based negotiation

Negotiation is defined as a process in which two or more parties with different criteria, constraints and preferences jointly reach an agreement on the terms of a transaction (Paprzcki et al., 2004). For agents to be involved in negotiation, we need to consider the following:

- Negotiation protocols (rules) governing interactions
- Negotiation issues (objects) over which agreement must be made
- Reasoning model (mechanisms used for negotiation) depends upon the first two points

The process of bargaining is a type of negotiation which occurs between a single buyer and a single seller. Buyers and sellers bargain with each other to reach a solution which is mutually acceptable. In the traditional sense, bargaining happens between a seller and customer. The seller is skilled in the arts of bargaining and will attempt to entice the buyer into making a purchase where there is more profit for the seller. In a bargaining process, the buyer and the seller exhibit different behaviours and attributes. (Sheng 2004). In a buyers' point of view, they would experience a sense of satisfaction at getting a deal even though the price achieved may not be the lowest possible. Generally, price may not be the most important factor for the buyer. From the sellers' point of view, they can make use of the extra information they have about the supply and demand of a particular item when setting its price. They are able to better customise their products to the individual needs of customers, hence giving a greater sense of satisfaction to their customers.

Agents provide an ideal platform for bargaining because the agent oriented model is inherently responsive to a rapidly changing environment, is dynamic in nature and autonomous in its actions. Agents are also designed to meet their design objectives. In an agent-based bargaining model, there would be specialised agents which carry out the tasks of parties involved in a bargaining process. There would be a buyer, who is trying to get the best combination of a few different variables (e.g. Price, quality, delivery time). The seller would be the bargaining agent who would try to entice the buyer into buying a product at the highest possible revenue. In order to reach a mutual agreement in a bargaining situation involves finding an acceptable solution for both the buyer and seller. This implies that the buyer's behaviours and attributes should also be considered. These behaviours and attributes could be modelled into a software agent who will then commence in the bargaining process. Figure 1 shows a typical bargain protocol between two agents.

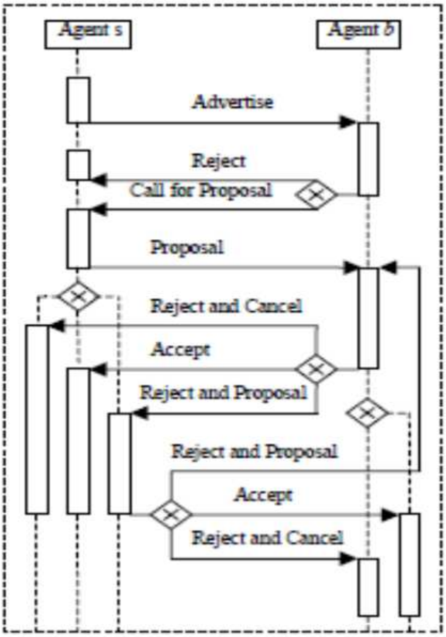


Fig. 1. A typical bargain protocol between two agents

2.1 Related work

There is a body of work on the applications of agent and multi-agent systems for electronic commerce. Recent advances in agent technology have promoted the development of intelligent e-commerce systems (Balachandran, 2010, 2011). Such systems are built upon the foundations of agent technology with a strong emphasis on the agent collaboration and negotiation capabilities. There have been several models developed for negotiation. Some examples include bargaining, fuzzy reasoning, and argumentation (Wooldridge, 2009). Broadly, these approaches have stemmed from two distinct fields of research: Game theory and artificial intelligence (Paprzycki 2004).

There have been many different mechanisms proposed for automated negotiations in literature and they each have their own advantages and drawbacks. Merlat (1999) discusses the potential of agent-based multiservice negotiation for e-commerce and demonstrates a decentralized constraint satisfaction algorithm (DCSP) as a means of multiservice negotiation. Badica et al (2006) present a rule based mechanism for agent price negotiation. They discuss the goals/challenges of developing an automated negotiation system for eCommerce. Sheng (2004) presents work that offers customers online business-to-customer bargaining service. Lai, Sycara, & Li (2007) present a Pareto optimal model for automated multi-attribute negotiations.and discuss issues, challenges and view points in agent-mediated eCommerce. Lomuscio et al (2003) provide an insightful overview of the existing research efforts on negotiation and describe a classification scheme for negotiation in electronic commerce. Liu and Feng (2007) presents work on e-commerce oriented automated negotiation based on FIPA Interaction Protocol Specification.

3. Modelling agent negotiation with multiple issues

If the bargaining process was centred over a single issue (such as the price), then it is relatively straightforward. The buyer will search for the lowest price offered for an item of their choice. Once the lowest price is found, that is the optimal solution to the problem. However, in real world bargaining situations, it is never this simple. There are always multiple issues to consider such as the price, quality, quantity, warranty, delivery date etcetera. As the number of issues being considered increases, so does the complexity of the negotiation (Yoshikawa 2006).

In this section we present a multi-issue negotiation model for e-commerce in which agents autonomously negotiate multi-issue terms of transactions in a bargaining environment. We use three agents in our model: a buyer agent, a seller agent, and a facilitator agent. The seller agent allows a seller to determine his negotiation strategies for selling merchandise. Similarly, the buyer agent allows a buyer to determine his negotiation strategies for buying merchandise. The facilitator agent serves to handle the negotiation strategies for both the buyer and the seller agents. In our approach, agents' preferences are expressed in fuzzy terms. The application domain for our prototype implementation is buying and selling laptop computers.

The negotiation model we have chosen for our study is illustrated in Figure 2. In this model, issues within both the buyer's request and the seller's offer can be split into hard constraints and soft constraints. Hard constraints are issues that have to be necessarily satisfied in the final agreement, whereas soft constraints represent issues they are willing to negotiate on. We utilise a facilitator agent which collects information from bargainers and exploits them in order to propose an efficient negotiation outcome.

The negotiation module consists of three components: negotiation object, decision making model and negotiation protocol. The negotiation object is characterised by a number of attributes for which the agents can negotiate. The decision making model consists of an assessment part which evaluates an offer received and determines an appropriate action, and an action part which generate and send a counter-offer or stop the negotiation. The assessment part is based on the fact that different values of negotiation issues are of different value for negotiating agents. We model the value of negotiating issues by scoring functions (Kurbel et al., 2004). The bigger the value of a scoring function for a certain value of an issue is, the more suitable is this value for a negotiating agent.

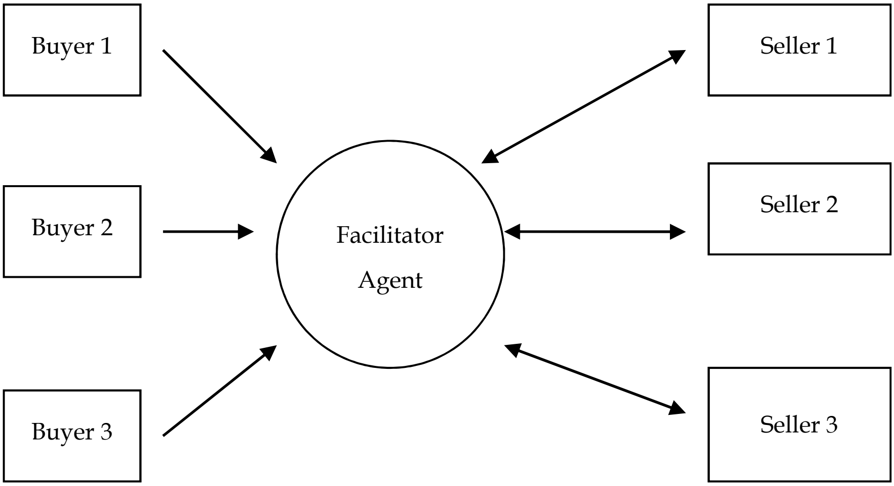


Fig. 2. One-to-many negotiation scheme

3.1 Scoring functions

The scoring functions represent private information about their preferences regarding the negotiation issues. This information is not given to other participants in the negotiation process. A scoring function is defined by four values of a negotiation issue. They are the minimal, maximal, optimum minimal and optimum maximal as illustrated in Figure 3 below:

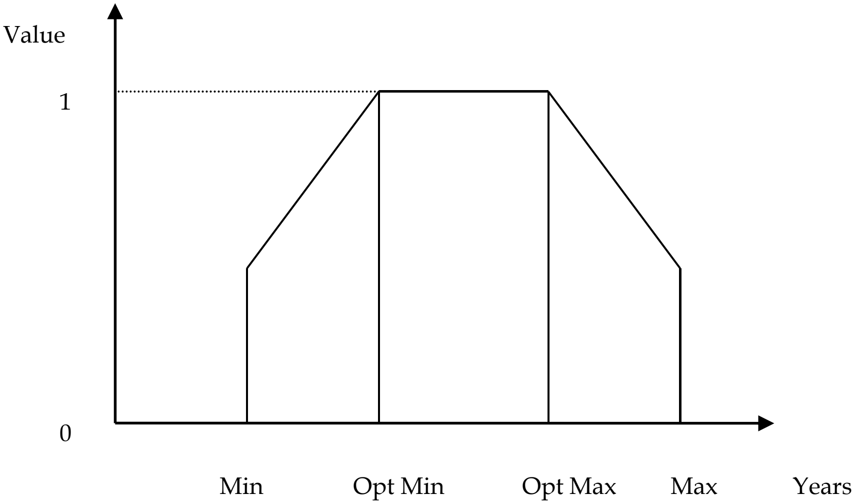


Fig. 3. Scoring Function for negotiation issue "number of years warranty"

We also consider the fact that different negotiation issues are of different importance for each participant. To model this situation, we introduce the weighting factor representing the relative importance that a participant assigns to an issue under negotiation.

During negotiation, the value of an offer received is calculated using two vectors: a vector-valued offer received by an agent and a vector of relative importance of issues under negotiation. The value of an offer is the sum of the products of the scoring functions for individual negotiation issues multiplied by their relative importance.

3.2 Pareto optimality

In order to find the most optimal solution to a multi-dimensional problem, the Pareto algorithm can be used to determine if one alternative is better than another (Keeny and Raiffa, 1976). A buyers Pareto optimal solution would be quite different to the Pareto optimal solution of a seller. In fact, they are likely to be the opposite. The negotiation process will attempt to find a mutually acceptable situation for both buyer and seller. This is likely to be in between the buyer's and the seller's Pareto optimal solutions. Figure 4 depicts a Pareto graph to show how two attributes (Price and Quality) are being optimized. The blue dots are where there are possible solutions and the red line shows where the optimal solution set is. All the possible solutions are covered under the red line. In this case, as you go towards the edges of the graph the item is being optimised. So to optimize price, one would have to go to the far right of the graph. This would have different meaning for buyer and seller because the buyer would consider the optimal price to be the lowest price, but the seller would consider the optimal price to be the highest possible price.

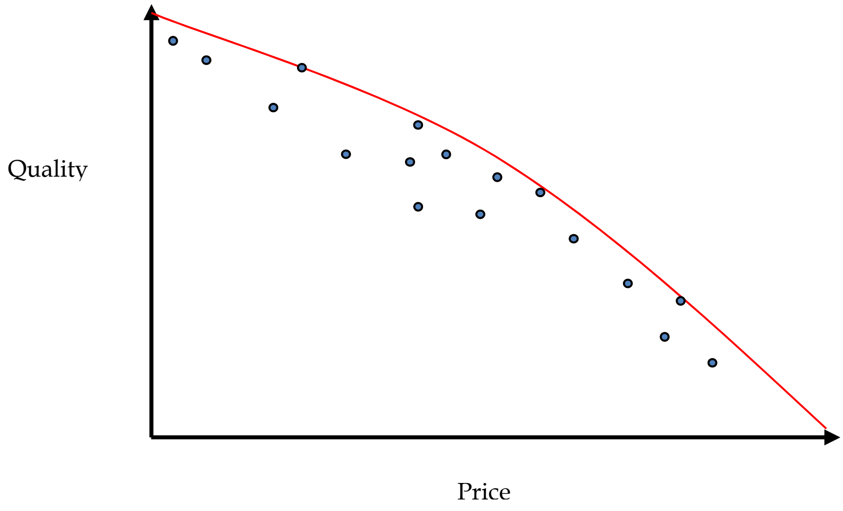


Fig. 4. Pareto Optimality Graph

3.3 The negotiation protocol

The negotiation facilitator receives this request and registers the customer. Once this is done, the negotiation process can begin with the suppliers. The negotiation facilitator requests the suppliers to provide offers conforming to the restrictions imposed by the customer agent. Please note that each restriction has an importance rating (0% to 100%), which means there is some leniency in the restrictions imposed by the customer. For example if the customer wants the colour Red, but provides an importance rating of 50%, it is quite lenient and the negotiation facilitator will request suppliers to make offers for a range of different colours. The negotiation facilitator and suppliers go through several rounds of negotiation until they reach the maximum number of rounds. Then the best offer (optimal set) is sent back to the customer agent. The customer agent then displays the results of the negotiation process to the end user who is ultimately responsible for making the decision on which item to buy.

3.4 The negotiation strategy

The facilitator's strategy is to gather a set of offers from the listed suppliers which satisfy the customer's wishes. Each offer is compared with the last offer by using a Pareto optimality algorithm. The facilitator has a utility algorithm which shows how good a particular offer is, this facilitator may modify the customers preferences (those which have an importance rating of less than 100%) in order to find other offers which may satisfy the user's needs. Once the set of optimal results are obtained, it is sent back to the customer agent.

The buyer's strategy is one which aims to maximise their profits on the goods sold to customers. They would also like to sell the goods as fast as possible, but at the highest possible price. The supplier does not want any old stock which cannot be sold.

3.5 The negotiation process

The negotiation process begins with registered buyers and sellers and a single facilitator. The seller sends a list of all items for sale to the facilitator. These items are registered for sale and available for all the buyers to bargain on and purchase. The buyer then registers with the facilitator and sends all their preferences. Once the preferences have been received by the facilitator, the negotiation process can begin between the facilitator and the supplier:

1. Facilitator runs the Pareto optimality algorithm to remove any sub-optimal solutions
2. Facilitator runs the Scoring function to get the item with the highest utility.
3. This item with the highest utility is selected as the negotiation item and set as the base item with all its properties (price, hard drive space etc.)
4. The item's properties are changed so that the property with the highest importance factor is increased. If importance factor of price was highest, it would be reduced by 10%. If the importance factor of the hard drive space or any other property was the highest, then it would be increased by 10%
5. This is sent to the supplier to see if they agree with the properties
6. This counter offer is received by the supplier who has a negotiable threshold amount (set to 10% by default) by which they are willing to negotiate on the items properties

a. If the negotiable threshold is not crossed, the counter offer is agreed to and sent back to the facilitator

b. If the negotiable threshold has been crossed, then check by how much. This difference is added to the price. If the threshold is crossed by 5%, then the price is increased by 5% and sent back to the facilitator
7. When the facilitator receives this offer, it calculates the utility of the offer and if it is greater, then it becomes the new base item. The next round of bargaining begins (back to step 4)
8. The bargaining process happens for a fixed number of rounds, 4 by default

4. An overview of the JADE platform

JADE (Java Agent Development Framework) is a software environment fully implemented in JAVA language aiming at the development of multi-agent systems that comply with FIPA specifications. This framework is provided free of charge by TILabs (http://jade.tilab.com/) and runs entirely on the Java runtime environment. JADE provides many of the base classes required for agent based software development (Bellifemine, Caire, and Greenwood, 2007). Some of them are:

- Agent
- Behaviour
- ACL Messaging
- Ontology

One goal of JADE is to simplify development while ensuring standard compliance through a comprehensive set of system services and agents. It provides the following mandatory components for agent's management:

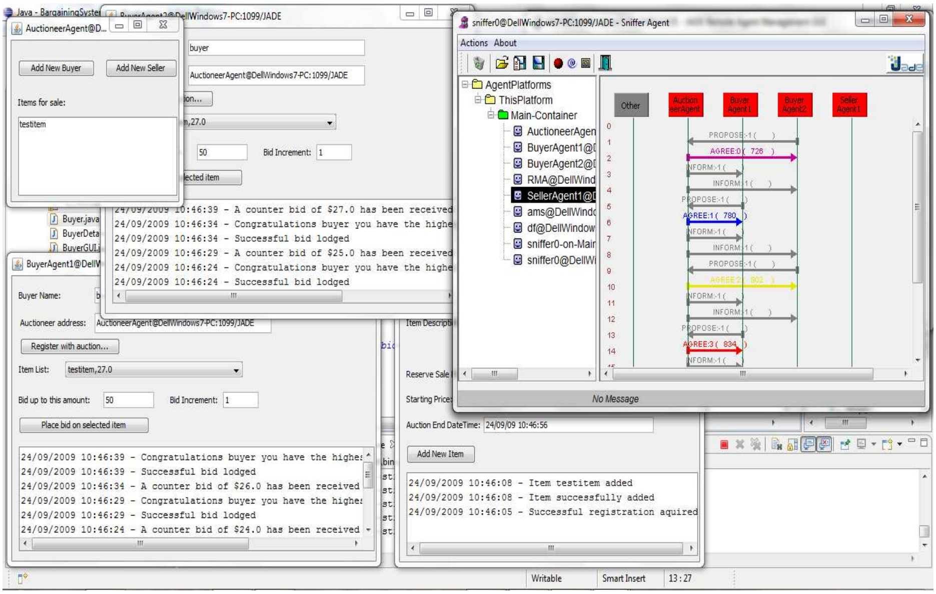
- AMS (Agent Management System), which besides providing white page services as specified by FIPA, it also plays the role of authority in the platform.
- DF (Directory Facilitator) provides yellow pages services to other agents.
- ACC (Agent Communication Channel) which provides a Message Transport System (MTS) and is responsible for sending and receiving messages on an agent platform.

Eclipse is the IDE (Integrated development environment) commonly used to develop the JADE application. It is quite easy to integrate Eclipse with JADE so that when the agent application is executed, it runs JADE and deploys the Agent into the runtime environment. Figure 5 shows a screen dump of the development environment using Eclipse and JADE.

In the following subsections we discuss the various steps involved in developing a multi-agent system using JADE.

4.1 Creating agents

Creating a JADE agent is as simple as defining a class that extends the jade.core. Agent class and overriding the default implementation of the methods that are automatically invoked by the platform during the agent lifecycle, including SetUp and TakeDown (). Consistent with the FIPA specifications, each agent instance is identified by an 'agent identifier'. In JADE an agent identifier is represented as an instance of the jade.core.AID class. The getAID () method of the Agent class allows retrieval of the local agent identifier.



buyer

AuctioneerAgent@DellWindows7-PC:1099/JADE

m,27.0

50 Bid Increment: 1

```
24/09/2009 10:46:39 - A counter bid of $27.0 has been received
24/09/2009 10:46:34 - Congratulations buyer you have the highe
24/09/2009 10:46:34 - Successful bid lodged
24/09/2009 10:46:29 - A counter bid of $25.0 has been received
24/09/2009 10:46:24 - Congratulations buyer you have the highe
24/09/2009 10:46:24 - Successful bid lodged
```

testitem

Buyer Name: b

Auctioneer address: AuctioneerAgent@DellWindows7-PC:1099/JADE

Item List: testitem,27.0

Bid up to this amount: 50 Bid Increment: 1

```
24/09/2009 10:46:39 - Congratulations buyer you have the highes
24/09/2009 10:46:39 - Successful bid lodged
24/09/2009 10:46:34 - A counter bid of $26.0 has been received
24/09/2009 10:46:29 - Congratulations buyer you have the highes
24/09/2009 10:46:29 - Successful bid lodged
24/09/2009 10:46:24 - A counter bid of $24.0 has been received
```

Auction End DateTime: 24/09/09 10:46:56

```
24/09/2009 10:46:08 - Item testitem added
24/09/2009 10:46:08 - Item successfully added
24/09/2009 10:46:05 - Successful registration aquired
```

Fig. 5. The Development environment using Eclipse and JADE

4.2 Defining agent tasks

In JADE, behaviour represents a task that an agent can carry out and is implemented as an object of a class that extends jade.core.behaviours.Behaviour. Each such behaviour class must implement two abstract methods. The action () method defines the operations to be performed when the behaviour is in execution. The done () method returns a Boolean value to indicate whether or not a behaviour has completed and is to be removed from the pool of behaviours an agent is executing. To make an agent execute the tasks represented by a behaviour object, the behaviour must be added to the agent by means of the add Behaviour () method of the Agent class.

4.3 Agent discovery process

The JADE platform provides a yellow pages service which allows any agent to dynamically discover other agents at a given point in time. A specialised agent called the DF (Directory Facilitator) provides the yellow pages service in JADE. Using this service any agent can both register (publish) services and search for (discover) services.

4.4 Agent communication

Agent communication is probably the most fundamental feature of JADE and is implemented accordance with the FIPA specifications. The JADE communication paradigm is based on asynchronous message passing. Each agent is equipped with an incoming message box and message polling can be blocking or non-blocking. A message in JADE is implemented as an object of the jade.lang.acl.ACLMessage object and then calling the send () method of the Agent class.

4.5 Defining ontologies

Agents must share semantics if the communication is to be effective. Therefore, exchanged messages must have written in a particular language and must share the same ontology. An ontology in JADE is an instance of the jade.content.onto.Ontology class to which schemas have been added that define the types of predicates, agent actions and concepts relevant to the addressed domain. These schemas are defined as instances of the PredicateSchema, AgentAction Schema and ConceptSchema classes included in the jade.content.schema package.

5. Developing multi-issue negotiation system with JADE

The model of a hypothetical system developed in the previous section is used to implement a prototype system capable of demonstrating the bargaining negotiation strategy. The proposed multi-issue negotiation system was implemented using the JADE environment. The system provides graphical user interfaces for users (buyers and sellers) to define scoring functions, weighting factors, negotiation tactics. It also has a customer management system for the system administrator.

One of the most useful tools to use when developing a multi-agent system with JADE is the Sniffer agent. This is another agent built into JADE which allows the user to see the message interactions taking place in real time. It can be seen in Figure 6 below that the interactions depicts the type of message, the sender and receiver and when it was sent within the lifetime of the system. If more information is required about any of the ACL messages, the user can double click the specific arrows and it will display full details.

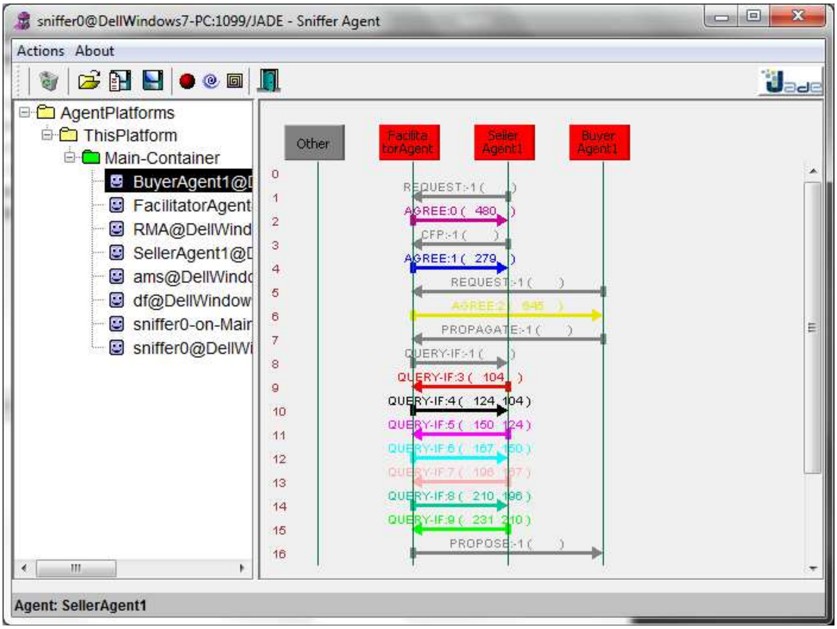


Fig. 6. A screenshot of the JADE sniffer agent

5.1 A case study

When someone wishes to buy a computer/laptop, they have to go through a process to determine which one is best suited to their needs and requirements. One of the buyer's requirements could be that the price is below $1500. Likewise the following could be the set of requirements for the buyer:

- 15.6" screen
- 500GB hard drive
- 4 USB ports
- Brand quality

For each of these requirements, there is a priority associated which shows how much the importance of fulfilling that particular requirements is for the buyer. For example, the laptop must be below or equal to $1500. This is not a negotiable item, so its importance factor is 100%. The hard drive, however, can be of a lower size as long as there is a reasonable adjustment in the other requirements of the laptop. For example, if the laptop has less than 500GB hard drive, then price of the product should be less than one which has a 500GB hard drive. In these ways, there are many different rules which assist the user in determining the value of one deal compared to another. The buyer is basically calculating optimality of one solution/possibility compared to another. The goal is to maximise the optimality so that there is no other solution which is of better suitability to the buyer.

In order for a customer with basic knowledge of computer systems to choose the best available computer, the buyer needs to do comprehensive research online and at electronic stores. There is often conflicting and confusing information on display to consumers. Because it is such a daunting experience to go through all the computers for sale and compare one to the other purely on a factual level, consumers can often make purchases which do not optimise the money they are spending.

Computer systems make decisions purely based on the facts available to it. They are able to sift through large amounts of data and use complex algorithms to compare and contrast the different options for the user. All this should be done based on the 'value' to the customer, rather than just the lowest price. Price is only one of the factors which influence a buyer's decision on whether or not to make a purchase. There are many other factors which must be considered. The majority of computer systems currently work based on the concept that a lower priced good is always better than a higher priced good. Web sites like EBay and Amazon allows the user to negotiate the price so that they may achieve the highest value. However, buying a product is rarely a black and white decision based purely on money. Admittedly the price is a very important factor, but it is definitely not the only factor which should be considered.

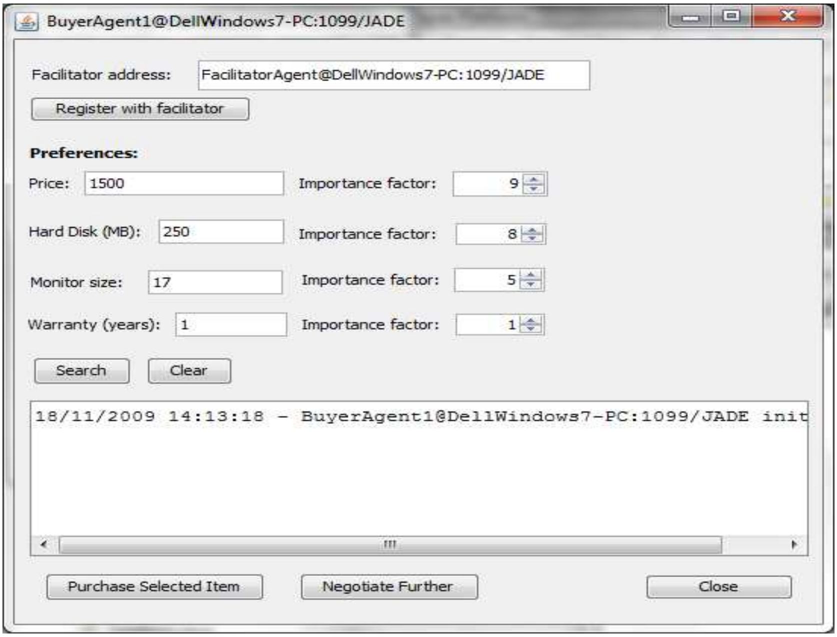
In this chapter, we present a prototype system which will demonstrate the use of multi-issue negotiation in order to achieve the optimal value for the user. So far, the buyer has been of our main concern, but similar considerations can be made for the seller in order to increase their revenue.

5.2 Agents in the system

This section will describe the agent screens as developed as part of the prototype multi agent system. There are three agents in total. Although there can be more than one instance of the buyer and seller, there can only be one instance of the facilitator running at any one time. This is a limitation on the system imposed to reduce the complexity of the application. The main aspect of the system that this project is interested is the negotiation component. The prototype system implements multi-attribute bargaining.

5.2.1 Buyer agent

The buyer agent is designed to get the preferences from the user, register with the facilitator and then receive the results of the negotiation process. From the point the user clicks on search, there is no interaction between this agent and the end user, until the negotiation results are returned. Using the buyer agent's screen , the end user selects their preference values and importance factors. This information is used by the facilitator during the bargaining process. Figure 7 shows the buyer agent screen.



BuyerAgent1@DellWindows7-PC:1099/JADE

Facilitator address: FacilitatorAgent@DellWindows7-PC:1099/JADE

Preferences:

Price: 1500 Importance factor: 9

Hard Disk (MB): 250 Importance factor: 8

Monitor size: 17 Importance factor: 5

Warranty (years): 1 Importance factor: 1

```
18/11/2009 14:13:18 – BuyerAgent1@DellWindows7-PC:1099/JADE init
```

Fig. 7. The buyer agent screen

5.2.2 Facilitator agent

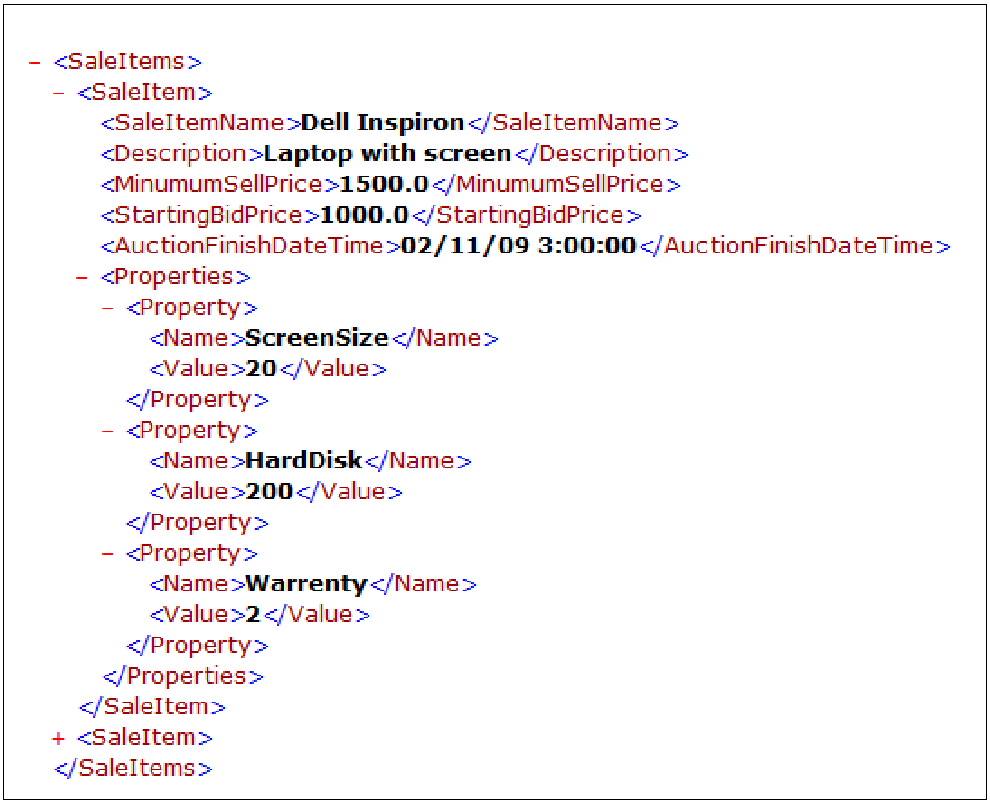
The facilitator agent receives registration requests from both the buyer and seller and then process the request (either accepts or denies the registration request). Once all the information for a round of negotiation is available, the facilitator looks after the bargaining process with the supplier. Once the maximum number of negotiation rounds has been completed, the facilitator sends the best offer back to the buyer.

5.2.3 Seller agent

The seller agent is responsible for registering with the facilitator and sending a list of sale items which are available. This agent also manages the counter offers received from the facilitator. The agent has a threshold limit as to how much it is able to negotiate. All offers where it needs to negotiate more incur an increase in the price of the good.

5.3 The negotiation object

The negotiation object in our model is the item which is being negotiated upon. This item, A, has several properties and each property has a name and value. For the purposes of this project, the name is a string and the value is an integer whole number. These item details are read in by the agents and the properties manipulated during the negotiation process. Figure 8 shows an example XML document showing a sale Item and its properties.



```
- <SaleItems>
  - <SaleItem>
      <SaleItemName>Dell Inspiron</SaleItemName>
      <Description>Laptop with screen</Description>
      <MinumumSellPrice>1500.0</MinumumSellPrice>
      <StartingBidPrice>1000.0</StartingBidPrice>
      <AuctionFinishDateTime>02/11/09 3:00:00</AuctionFinishDateTime>
    - <Properties>
      - <Property>
          <Name>ScreenSize</Name>
          <Value>20</Value>
        </Property>
      - <Property>
          <Name>HardDisk</Name>
          <Value>200</Value>
        </Property>
      - <Property>
          <Name>Warrenty</Name>
          <Value>2</Value>
        </Property>
      </Properties>
    </SaleItem>
  + <SaleItem>
  </SaleItems>
```

Fig. 8. A sale item representation in XML

5.4 The negotiation protocol

The negotiation facilitator receives this request and registers the customer. Once this is done, the negotiation process can begin with the suppliers. The negotiation facilitator requests the suppliers to provide offers conforming to the restrictions imposed by the customer agent. Please note that each restriction has an importance rating (0% to 100%), which means there is some leniency in the restrictions imposed by the customer. For example if the customer wants the colour Red, but provides an importance rating of 50%, it is quite lenient and the negotiation facilitator will request suppliers to make offers for a range of different colours. The negotiation facilitator and suppliers go through several rounds of negotiation until they reach the maximum number of rounds. Then the best offer (optimal set) is sent back to the customer agent. The customer agent then displays the results of the negotiation process to the end user who is ultimately responsible for making the decision on which item to buy.

5.5 The negotiation strategy

The facilitator's strategy is to gather a set of offers from the listed suppliers which satisfy the customer's wishes. Each offer is compared with the last offer by using a Pareto optimality algorithm. The facilitator has a utility algorithm which shows how good a particular offer is, and the facilitator may modify the customers preferences in some circumstances (those which have an importance rating of less than 100%) in order to find other offers which may satisfy the user's needs. Once the set of optimal results are obtained, it is sent back to the customer agent.

The seller's strategy is one which aims to maximise their profits on the goods sold to customers. They would also like to sell the goods as fast as possible, but at the highest possible price. The supplier does not want any old stock which cannot be sold.

6. Summary and conclusions

In this chapter, we have described a multi-issue negotiation protocol which allows agents to follow a process, in order to end up with an optimal decision. Further we have described design and implementation of automated negotiations in an e-commerce modelling multi-agent system. We have implemented a working prototype of the system using the JADE and ECLIPSE platforms. JADE is very easy to learn and use. Moreover we found that JADE is quite suitable for developing multi-agent systems as it supports many agent concepts such as agent communication, protocol, behaviour and ontology. We have done some evaluation to investigate the satisfaction of using the prototype system. The System has met the functionalities expected and has been tested for the domain of buying a laptop computer for given user preferences. The concepts and models utilised in this chapter are very promising for the future e-commerce applications.

There are a number of areas of further investigation. We would like to compare our negotiation strategy with other decision theoretic approaches to determine the relative strengths and weaknesses of these methods. The use of scoring function is purely number based and the users' preferences currently have to be numerated into a number value indicating their importance. This may not be possible in many situations. We intend to use fuzzy techniques to model user preferences. Furthermore the current negotiation strategy is heavily based upon the views of the buyer, and attempts to negotiate a deal with the supplier which would see greater value for the buyer. A better supplier strategy could be implemented which could also take multiple issues into consideration (such as stock items for sale, and time before stock replacement).

In conclusion, there are opportunities for future improvements and further development in this area of e-commerce. The current system focuses on utility based bargaining on different items for sale. It aims to provide the best possible deal for the customer. Work could be done in the future to further enhance the model and the system to incorporate new tools and techniques used in the electronic marketplace. In the future, we would like to extend our work by implementing more sophisticated individual strategies and knowledge sharing ability between agents. Another important area of extension is the ability to handle non-linear scoring functions to represent agents' preferences.

7. Acknowledgment

We would like to thank Mr Tauhid Tayeb, for his efforts in implementing the prototype system described in this chapter.

8. References

Alem, L, Kowalczyk, R, and Lee, M.R. (2002). Supporting Electronic Negotiation for Intelligent Trading, Idea Group Publishing

Badica, C., Badita, A., & Ganzha, M. (2006). Implementing rule-based mechanisms for agent-based price negotiation. ACM, 96-100

Balachandran, B.M, Gobbin, R, and Sharma, D. (2011). Development of a Multi-Issue Negotiation System for E-Commerce, KES-IDT'2011, Athens, Greece

Balachandran, B.M, Tayeb, T, Sharma, D, and Mohammadian, M. (2010). An Implementation of a Multi-Attribute Negotiation Protocol for E-Commerce, KES-IDT'2010, Baltimore, Maryland, USA

Bellifemine F, Caire G, and Greenwood, D. (2007) Developing Multi-Agent Systems with JADE, John Wiley & Sons, UK

Fasli, M. (2007). "Agent Technology for e-Commerce", John Wiley and Sons, UK

FIPA (2006) The Foundation for Intelligent Physical Agents. See http://www.fipa.org/
JADE: Java Agent Development Framework: http://jade.tilab.com

Keeny,R. and Raiffa,H.(1976). Decisions with Multiple Objectives: Preferences and Value Tradeoffs, John Willey & Sons

Kurbel, K., Loutchko, I, and Teuteberg, F (2004). FuzzyMAN: An agent-based electronic market place with a multilateral negotiation protocol, G. Lindemann et al. (Eds), MATES 2004, LNAI 3187, pp.126-140, 2004

Lai, G., Sycara, K., & Li, C. (2007). A Pareto Optimal Model for Automated Multi-attribute Negotiations. IFAAMAS

Liu, K. and Feng, Y. (2007). E-Commerce Oriented Automated negotiation Based on FIPA Interaction Protocol Specification, Proceedings of the Sixth International Conference on machine Learning and Cybernetics, Hong Kong, 19-22 August 2007

Lomuscio, A.R., Wooldridge, M, and Jennings, N.R. (2003). A Classification Scheme for Negotiation in Electronic Commerce, Group Decision and Negotiation, Kluwer Academic Publishers, 12:31-56

Merlat, W. (1999). An Agent-Based Multiservice Negotiation for Ecommerce, BT technical Journal, Vol 17 No 4 pp:168-175

Moreno, A and Isern, D. (2002). Accessing distributed health-care services through smart agents, Proceedings of the 4th IEEE International Workshop on Enterprise Networking and Computing in the Health Care Industry, Nancy, France, pp34-41

Paprzycki M et al., (2004) Implementing Agents Capable of Dynamic Negotiations, in D. Petcu et. al. (eds) Proceedings of SYNASC04: Symbolic and Numeric Algorithms for Scientific Computing, Mirton Press, Timisoara, Romania, pp. 369-380

Sycara,K.P. (1998). "Multiagent Systems," AI Magazine vol. 19(2), pp. 79-92

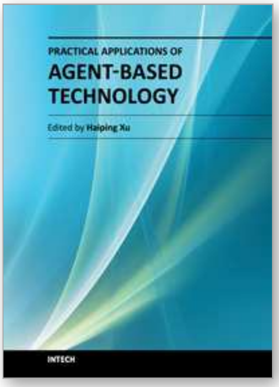
Sheng, Y.P. (2004) A dynamic adaptive bargaining algorithm for intelligent software agents in electronic commerce, International Journal of Computers, Systems and Signals, Vol. 5, No. 1, 2004

Takayuki, I., Hattori, H., and Klein, M. (2007). Multi-Issue Negotiation Protocol for Agents: Exploring Nonlinear Utility Spaces, IJCAI-07, pp1347-1352.

The Eclipse Platform, See http://www.eclipse.org/

Wooldridge M (2009) Introduction to Multiagent Systems, 2nd Edition, John Wiley and Sons, UK

Yoshikawa, S., Kamiryo, T., Yasumura, Y., & Uehara, K. (2006). Strategy Acquisition of Agents in Multi-Issue Negotiation. ACM International Conference of Web Intelligence



Practical Applications of Agent-Based Technology
Edited by Dr. Haiping Xu

ISBN 978-953-51-0276-2
Hard cover, 136 pages
Publisher InTech
Published online 18, March, 2012
Published in print edition March, 2012

Agent-based technology provides a new computing paradigm, where intelligent agents can be used to perform tasks such as sensing, planning, scheduling, reasoning and decision-making. In an agent-based system, software agents with sufficient intelligence and autonomy can either work independently or coordinately with other agents to accomplish tasks and missions. In this book, we provide up-to-date practical applications of agent-based technology in various fields, such as electronic commerce, grid computing, and adaptive virtual environment. The selected applications are invaluable for researchers and practitioners to understand the practical usage of agent-based technology, and also to apply agent-based technology innovatively in different areas.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Bala M. Balachandran (2012). Developing a Multi-Issue E-Negotiation System for E-Commerce with JADE, Practical Applications of Agent-Based Technology, Dr. Haiping Xu (Ed.), ISBN: 978-953-51-0276-2, InTech, Available from: http://www.intechopen.com/books/practical-applications-of-agent-based-technology/developing-a-multi-issue-e-negotiation-system-for-e-commerce-with-jade

INTECH
open science | open minds

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821