uOttawa

L'Université canadienne
Canada's university

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTOCTORALES

uOttawa

L'Université canadienne
Canada's university

FACULTY OF GRADUATE AND POSDOCTORAL STUDIES

Xin Wang

AUTEUR DE LA THÈSE / AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)

GRADE / DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT / FACULTY, SCHOOL, DEPARTMENT

A Fuzzy Logic Based Intelligent Negotiation Agent

TITRE DE LA THÈSE / TITLE OF THESIS

N. Georganas

DIRECTEUR (DIRECTRICE) DE LA THÈSE / THESIS SUPERVISOR

CO-DIRECTEUR (CO-DIRECTRICE) DE LA THÈSE / THESIS CO-SUPERVISOR

EXAMINATEURS (EXAMINATRICES) DE LA THÈSE / THESIS EXAMINERS

C. Joslin

E. Petriu

Gary W. Slater

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES /
DEAN OF THE FACULTY OF GRADUATE AND POSTDOCORAL STUDIES

A Fuzzy Logic Based Intelligent Negotiation Agent

Xin Wang

Thesis submitted to the

Faculty of Graduate and Postdoctoral Studies

in partial fulfillment of the requirements for the degree of a

Master of Applied Science

in

Electrical Engineering

The Ottawa-Carleton Institute for

Electrical and Computer Engineering

School of Information Technology and Engineering

University of Ottawa

Your file Votre référence
ISBN: 0-494-14968-X
Our file Notre référence
ISBN: 0-494-14968-X

Abstracts

With the evolution of electronic commerce (eCommerce) on the web and the rise of interests in intelligence of software agents, automated negotiation is becoming an increasingly popular method for an eCommerce system to be efficient; however, negotiation, which takes place in transaction, is complicated, time-consuming and costly for participants to reach an agreement. This thesis presents a model of an intelligent negotiation agent based on fuzzy logic methodology in order to alleviate the complexity of negotiation. The proposed negotiation agent model is particularly suitable to open environments, such as the Internet. The conventional methods, such as game theory, are incapable of handling an open environment where the information is sparse and full of uncertainty, while the fuzzy approaches are suitable to elegantly deal with this problem.

The fuzzy logic based intelligent negotiation agent, presented in this thesis, is able to interact autonomously and consequently save human labor in negotiations. The aim of modeling a negotiation agent is to reach mutual agreement efficiently and intelligently. The negotiation agent is able to negotiate with other such agents, over various sets of issues, on behalf of the real-world parties they represent, i.e. it can handle multi-issue negotiation.

The reasoning model of the negotiation agent has been implemented partially by using c# based on Microsoft .NET. The reliability and the flexibility of the reasoning model are finally evaluated. The results show that performance of the proposed agent model is acceptable for negotiation parties to achieve mutual benefits.

Acknowledgements

It was my pleasure and privilege to work under Professor Nicolas D. Georganas. My profound thanks for Prof. Georganas's supervision and guidance of my research and thesis..

I am very grateful to Dr. Xiaojun Shen for his helpful suggestions and corrections of this thesis and this project.

It was wonderful to work in the DISCOVER laboratory. I would like to thank my friends and colleagues, especially Mr. Francois Malric, for their support and advice.

Xin Wang

Ottawa, December 2005

Table of Contents

List of Figures

Chapter 1

Introduction

1.1 Background and motivation

Electronic Commerce (eCommerce) provides businesses, efficiency, cost savings, productivity in their business process. With today's improvements in internet technology, for example, reliability, security, higher speeds (broadband) and cheaper costs, eCommerce over the web is quickly growing. E-Commerce is now globally flourishing and growing rapidly. E-commerce is helping transform business into a network structure thus providing greater value for their products, less costs and access to their customers. E-commerce brings production and consumption closer and enterprises gain a wider and fairer competitive market while consumers gain more choices and more personalized services [1].

People have used negotiation as a means of compromise, in order to reach mutual agreements since the early days of civilization. Today negotiation is still primitive in eCommerce. There is a need for more sophisticated automated negotiation in eCommerce. Negotiation is a critical activity in business transactions. In general "negotiation" is defined as an iterative process which aims to achieve a mutually beneficial deal for the seller and buyer [2]. Negotiations can be done manually in eCommerce, for example, by emails, but its not, for example, time, cost effective. There is a need for automated negotiation, for example, negotiation process using two intelligent agents negotiate a solution autonomously would be more efficient and objective.

Software agents are currently being used for information retrieval and for offering recommendations such as finding product information, comparing product prices, and offering suggestions on product and services based on customer's interest and preferences [3].

Software agents are highly customizable and personalization enhances interactivity.

Software agents also interact with other software agents to achieve mutually agreeable terms and conditions of a business transaction. For example, in a simple auction, a buyer agent may negotiate with one or more seller agents, or even can take part in bidding in an online auction.

Although an electronic market could eliminate the geographic obstacles to a large extent, the barriers of, for example, culture and other constraints necessitate human involvement. But human based negotiation is subjective, time consuming and generally inefficient.

This thesis proposes an intelligent negotiation mechanism for software agents that use fuzzy logic that to a large extent eliminates the need for human involvement in the eCommerce negotiations.

1.2 Research Objective

To alleviate much of today's problems, such as high inefficiency, subjectivity, etc., inherent in human negotiations, we are proposing an intelligent negotiation technology for eCommerce.

The objectives for the above proposed technology are:

1. To find appropriate methodology to model an automated negotiation agent in order to alleviate the complexity and inefficiencies of existing negotiations technologies.

2. The automated negotiation agent should be suitable for an open environment, such as the Internet, where the available information may lack context, quantification. .

3. The automated negotiation agent should be able to handle multi-issue negotiation, for example, that is not just costs for products, but also, for example, deliver time, return policy, quality of products, etc.

4. The negotiation agent should be capable to interact and achieve mutual agreement intelligently i.e. the negotiation agent can make decisions autonomously.

1.3 Structure of thesis

The thesis is broken into six chapters. The reminder of the thesis is organized as follows.

Chapter Two provides an overview of eCommerce and software agent technology; the concepts and definitions related to eCommerce and software agents are also discussed.

Chapter Three gives detailed explanation about the process of negotiation and reviews the current research trends on modeling negotiation.

Chapter Four reviews the basics of fuzzy logic and fuzzy expert systems in the first part. The proposed negotiation agent model is described in the second part.

Chapter Five discusses implementation of a specific example of reasoning model first. Tests of its performance and feasibility are shown secondly.

Chapter Six, finally, concludes and summarizes the contributions and advantages of the proposed negotiation agent model.

Chapter 2

Overview of eCommerce and Agent Technology

2.1 E-Commerce

2.1.1 Traditional commerce

The basics of commerce can be viewed as trade between buyer and seller. One member of society created something of value that another member of society desired. Commerce, or doing business, is a negotiated exchange of desired product from the seller and money or other product from the buyer. The above exchange involves desired objects or services between at least two parties and includes all activities that each of the parties undertakes to complete the transaction [4].

The Buyer

Any commerce transaction can be examined from either the buyer's or seller's viewpoint. The elements of traditional commerce for buyers can be specified as: 1) identifying specific need, 2) searching for products or services that will satisfy the above specific need, 3) selecting a vendor from a set of vendors that appears to provide the above buyers need, 4) negotiating a purchase transaction, 5) making payment, and 6) performing regular maintenance and if required making warranty claims [4].

A buyer begins by identifying a need. The complexity of the need-identification varies. The need may be simple, for instance, an individual client decides, "I'm hungry and would like to find some lunch now." The need also might be complex, such as a city council trying to decide "We must find a way to generate clean power that will meet our city's needs in 25 years." The need-identification process for a hungry individual may require, for example, no more than a quick consideration of which fast-food outlets are nearby and open. Identifying the specific needs for the power generation example could require a team of experts with expertise in, for example, demographics, industry growth and present/future power requirements, cost, distribution, environmental considerations, etc. Most need-identification processes fall between the above two extremes.

Once buyers identify their specific needs, they must find products or services that will satisfy those needs. In traditional commerce, buyers use a variety of search techniques. They may consult catalogs, ask friends, read advertisements, or examine directories. The Yellow Pages is a good example of a directory that buyers often use to find products and services. Buyers may consult salespersons to gather information about specific features and capabilities of the products they are considering. Companies often have highly structured procedures for finding products and services that satisfy recurring needs of their businesses [5].

After buyers select a product or service that will meet the identified need, they must select a vendor who can supply the product or service. Buyers in traditional commerce contact vendors in a variety of ways, including by telephone, by mail, and by trade shows. After choosing a vendor, the buyer negotiates a purchase transaction. This transaction may have many elements, including delivery date, method of shipment, price, warranty, and payment terms, and will often include detailed specifications to be confirmed by inspection when the product is delivered or the service is performed. When the buyer is a business, the negotiation of a purchase transaction can be very complicated. Imagine, for example, the complex ordering, delivery, and inspection activities that must occur when an airline buys a new airplane from an aircraft manufacturer. Businesses often have entire departments devoted to negotiating purchase transactions with their suppliers. These departments are usually named supply management or procurement.

When the buyer is satisfied that the purchased product or service meets the requirements and conditions agreed to by both buyer and seller the buyer pays for the purchase. After the transaction is complete, the buyer may have further contact with the seller regarding warranty claims, upgrades, and regular maintenance.

The Seller

The seller has the required product that satisfies the buyers need. The seller aims to make a profit on the above product and strives to strike a mutually beneficial deal with the buyer.. The elements of commerce from a seller's viewpoint can be divided as: 1) Conducting market research to identify customer needs, 2) Creating products or services that will meet customers' needs, 3) Advertising and promoting products or services, 4) Negotiating a sale transaction, 5) Shipping goods and invoicing customer, 6) Receiving and processing customer payments, and 7) Providing after-sale support, maintenance, and warranty services.

Sellers often undertake market research to identify potential customers' needs. Even businesses that have been selling the same product or service for many years are always looking for ways to improve and expand their offerings. Firms conduct surveys, have salespeople talk with customers, run focus groups, and hire outside consultants to help them in this identification process.

Once customer needs are identified, sellers create the products and services that they believe will meet those needs within the constraints of their budgets, manpower, etc.. This product creation activity includes product design, testing, and production and marketing activities, like to make potential customers aware of the availability of the above product, by different kinds of advertising and promotional activities to find and target potential buyers [6].

Once a buyer responds to the seller's promotional activities, the two parties negotiate the details of a purchase transaction. In some cases, this is simple; for example, many retail transactions involve nothing more than a buyer entering a seller's store, selecting and inspecting items to purchase, and paying for them. In other cases, purchase transactions require prolonged negotiations to settle the terms of delivery, inspection, testing, and acceptance.

After the seller and buyer resolve delivery logistics, the seller ships the goods, or provides the service, and sends an invoice to the buyer. In some businesses, the seller also provides a monthly billing statement to each customer, summarizing the invoicing and payment activity of that customer. In some cases, the seller requires payment before or at the time of shipment. However, most businesses sell to each other on credit, so the seller must keep a record of the sale and wait for the customer to pay. Most businesses maintain sophisticated systems for receiving the processing customer payments. They want to track the amounts they are owed, and ensure that payments they receive are credited to the proper customer and invoice.

Following the conclusion of the sale transaction, the seller often provides continuing after-sale support for the product or service. In many cases, the seller is bound by contract or statute to guarantee or warrant that the product or service sold will perform satisfactorily for a specific period of time. The seller provides support, maintenance, and warranty work to help ensure that the customer is satisfied and will return to buy again [4].

2.1.2 Why electronic commerce

The definition of electronic commerce, mentions the use of electronic data transmission to implement or enhance business processes [4]. Some people use the term Internet commerce to mean electronic commerce that specifically uses the internet or the web as its data transmission medium.

Businesses are interested in electronic commerce because it can help increase profits by reducing transaction costs, advertising costs and global reach. A business can use electronic commerce to reach narrow market segments that are geographically scattered. The Web is particularly useful in creating virtual communities that become ideal target markets for specific types of products or services. A virtual community is gathering of people who share a common interest, but instead of this gathering occurring in the physical world, it takes place on the Internet.

A business can reduce the costs of handling sales inquiries, providing price quotes, and determining product availability by using electronic commerce in its sales support and order-taking processes. Since no customer service representatives are involved in making these sales, businesses can save money and operate more efficiently.

Just as electronic commerce increases sales opportunities for the seller, it increases purchasing opportunities for the buyer. Businesses can use electronic commerce to identify new suppliers and business partners. Negotiating price and delivery terms is easier in electronic commerce because the Internet can help companies efficiently obtain competitive bid information speedily. Electronic commerce increases the speed and accuracy with which businesses can exchange information, which reduces costs on both sides of transactions.

Electronic commerce provides buyers with a wider range of choices than traditional commerce because buyers can consider many different products and services from a wider variety of sellers [4]. This wide variety is available for consumers to evaluate 24 hours a day, every day. Some buyers prefer a great deal of information in deciding on a purchase; others prefer less. Electronic commerce provides buyers with an easy way to customize the level of detail in the information they obtain about a prospective purchase. Instead of waiting days for the mail to bring a catalog or product specification sheet, or even minutes for a fax transmission, buyers can have instant access to detailed information on the web. Some products, such as software, audio clips, or images, can even be delivered through the Internet, which reduces the time buyers must wait to begin enjoying their purchases.

The benefits of electronic commerce extend to the general welfare of society. Electronic payments of tax refunds, public retirement, and welfare support cost less to issue and arrive securely and quickly when transmitted over the Internet. Furthermore, electronic payments can be easier to audit and monitor than payments made by check, providing protection against fraud and theft losses. To the extent that electronic commerce enables people to work from home, we all benefit from the reduction in commuter-caused traffic and pollution. Electronic commerce can also make products and services available in remote areas. For example, distance education is making it possible for people to learn skills and earn degrees no matter where thy live or which hours they have available for study.

2.2 Agent technology

2.2.1 Definition

In this thesis, an agent is a software system capable of flexible, autonomous behavior in dynamic, unpredictable, typically multi-agent domains [5].

Some advantages can be found for using an agent in implementing an electronic commerce system. An agent provides codeless implementation of some of the electronic commerce system that greatly simplifies "personalization" of electronic commerce systems.

Software agents are particularly important when it comes to distributed environments. Agents are given rules as to how to interpret the current situation and, given a common goal, so that they adjust their response accordingly [5].

2.2.2 Agent properties

Software agent properties as applicable to electronic commerce are given below.

Autonomy

An agent should be an independent system that can make its own decision and execute tasks on by itself. [6]. Moreover, an agent has to have the ability of acting autonomously in unknown scenarios where it is difficult to control the behavior of the system.

Personalization

An agent can be equipped with a personal profile to reflect the preferences of its user. It can be taught what to do for each individual or a group of users.

Social behavior

An agent can interact with human user or other software agents in order to accomplish its goals better.

Intelligence

An agent can learn and hence perform better over. In electronic commerce scenarios this may equate to making more money. Most of artificial intelligence can be applied [6], such as Neural Network, Bayesian Networks, and Expert System etc.

2.2.3 Types of agents

The three categories of agents are:

1. Distributed Artificial Intelligence (DAI) - Multi-Agent Systems (MAS)
2. Much broader notion of "agent" (from the 90's till now)
3. Agent-based modeling and yet another type of agents (the future) [4]

Nwana's classification (Figure 1) is a good way for the overview of roles for different agents.

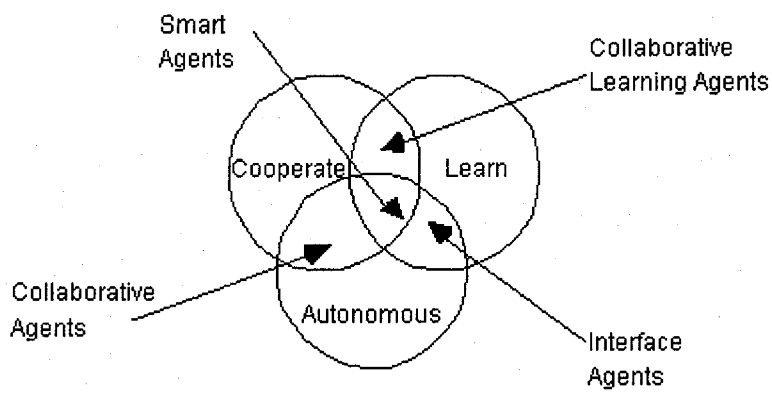


Figure 1: Nwana's classification [7]

Collaborative Agents

Collaborative agents are the first type agents. Agent research starts here. The agents can negotiate in order to resolve conflicts (e.g. meeting time), and some of them will collaborate to integrate information. The agents will also provide solution to inherently distributed problems such as air traffic control or telecommunications network management [7]. Contract Net is one of the most popular protocols collaborative agents use [41].

Interface agents

Interface agents are the second type of agents. They support and provide assistance to the user and cooperate with the user in accomplishing some task in an application. Interface agents can learn by observing and imitating the user, through receiving feedback from the user, by receiving explicit instructions, and by asking other agents for advice.

The interface agents' architecture is shown in Figure 2.

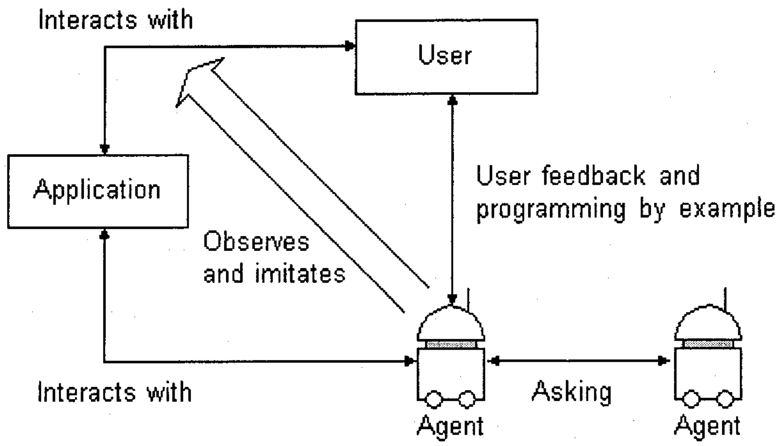


Figure 2: Interface agents' architecture [4]

Reactive agents

Reactive agents do not have internal symbolic models, and they act by stimulus-response to the current state of the environment. The agents are based in subsumed architecture, and each reactive agent is simple and interacts with others in a basic way. The benefit for using reactive agents is that the robust agents can provide a fast response time.

Mobile agents

Mobile agents are some programs that can migrate from one machine to another. They can be divided into one-hop and multi-hop mobile agents. In practice, the agents can be used to reduce communication cost and for asynchronous computing.

The mobile agents' architecture is shown in Figure 3.

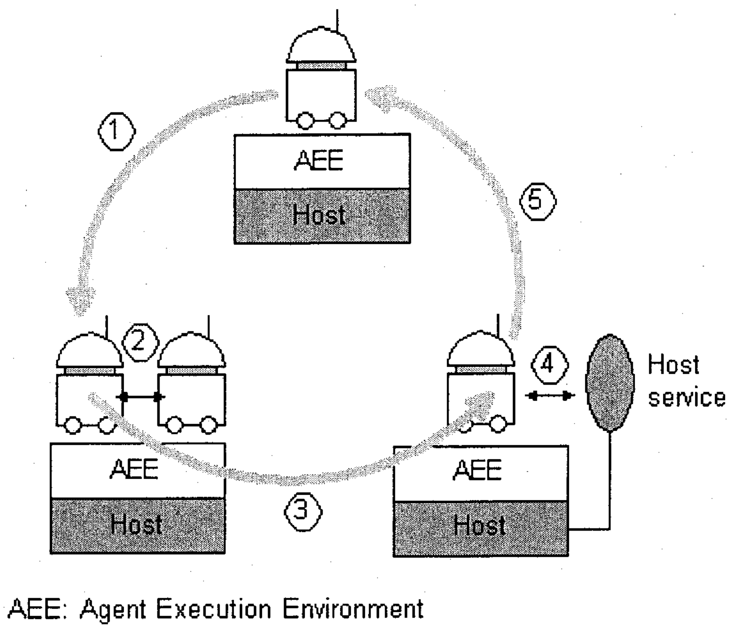


Figure 3: The mobile agents' architecture [7]

Information agents

Information agents are used to manage the explosive growth of information and manipulate or collate information from many distributed sources. Information agents can be mobile or static. A good example for the agents is that Jasper [43] works on behalf of a user or community of users and stores retrieves and informs other agents of useful information on the Internet.

The information agents' architecture is shown in Figure 4.

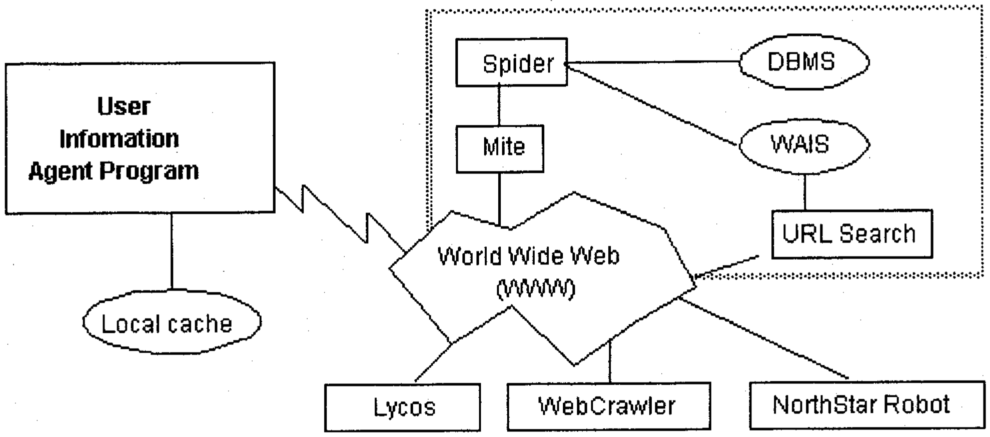


Figure 4: The information agents' architecture [7]

Economic agents

Economic agents are the third type of agents. They represent the latest development in agent research. Economic agents evolved out of various separate research efforts, e.g. social sciences, artificial life, complex adaptive systems, market-oriented programming, and pricing of Internet services. The agents share the following characteristics:

- Emergence
- Evolution of social norms
- Self-regulating
- Adaptive to changes
- Value added (efficiency)

Chapter 3

Negotiation and Modeling the Negotiation Problem

3.1 Introduction to Negotiation

This chapter describes an electronic negotiation (e-negotiation) system that facilitates the commercial transaction. The above system is constructed on the technique of multi-agent system. This chapter is organized as follows. Section 3.1.1 presents the definition, the type and the objective of negotiation. Section 3.1.2 discusses related work in the literature, Section 3.1.3 states the research issues based on the e-negotiation process. Lastly, Section 3.1.4 gives detailed ideas on a future implementation model of the e-negotiation system.

3.1.1 Overview

Negotiation helps achieve a deal between a buyer and seller. There are two types of negotiation: distributive negotiations (also known as zero-sun and competitive negotiations) and integrative negotiations (also known as collaborative and cooperative negotiations). The former type follows the win-lose policy. One negotiation party can reach its goals and the other negotiation parties fail. In eCommerce, this type of negotiations is mainly used in electronic auctions, which are popular in current on-line trading. On the other side, an integrative negotiation is the communication process of a group of agents in order to reach a mutually acceptable agreement on some matter [8]. It follows the win-win policy. The objectives of an integrative negotiation are to gain greater understanding of the other side's motives, objectives, and constraints, allow the other side to learn some of your motives, objectives, constraints, seek common ground (mutual success) and build relationships (building a good relationship facilitates effective future negotiations). This type of negotiation is very complex and hard to construct an analytic model and in turns a support system. In most of current eCommerce applications, there is no support for integrative negotiation. As negotiation is a key stage in commercial transactions, providing intelligent support is now an important research area in such applications. This thesis focuses on the e-negotiation system for integrative negotiations.

3.1.2 Existing research

Internet technology introduces new tools for conducting electronic negotiations (E-negotiations). It reduces the cost of negotiations in term of coordination and information exchange. Computer applications first were proposed for negotiation support in the 1960s [9]. Computer-based Negotiation Support Systems (NSS) appeared in 1980s. They were used for laboratory research but seldom for the practical business world. NSS is designed to aid and provide strategies to negotiators. For example, NEGOTIATOR [10] helps the negotiators to make decision by sharing information, rearranging the problem. INSPIRE [11] focuses on the evaluation of the goodness of the offer. NegoPlan [12] is constructed by an expert system. It formulizes the negotiation problems by using rule-based models. However, these NSSs have been deployed for teaching and research purposes.

Recently, software agents have been playing important roles in eCommerce to support negotiation activities. Software agents are programs that are operated by a user or another program with some degree of independent or autonomy. A set of tasks are designed for their goals. Agents are capable of reasoning which is realized by a set of "if-then" rules, or complicated machine learning algorithms such as neural networks or Bayesian networks [13]. Software agents have certain level of autonomy and can be personalized. Some agents which designed for negotiation help to conduct customer-to-customer transaction for example, the MIT Media Lab's Kasbah [14]. Tete-a-tete[15] is another example of agent-based system that seeks the mutual goal on the multiple terms of transactions including warranties, delivery times, service contracts, return policies, loan options, gift services, and other merchant value-added services.

Negotiation on business information is a much more complex subject. At present and in the near future, the social aspect of the negotiation is in the establishment of a relationship and understanding. The former might be done with the help of agents. However, the latter requires the users' participation. Users need to predefine their negotiation strategies. An e-negotiation system will make eCommerce more powerful, interesting, practical, and personalized. We propose to engage buyer-owned *shopping agents* and merchant-owned *sales agents* in integrative negotiations over the value of each product to optimize their owners' benefits. It also helps merchants differentiate themselves along dimensions other than just price, such as delivery time, return policies, support and repair services, etc.

3.1.3 On the negotiation process

The negotiation process goes through three distinct phases [16]: (1) Pre-negotiation (2) conduct of negotiation, (3) post-settlement. The support of the e-negotiation system is illustrated in Figure 5.

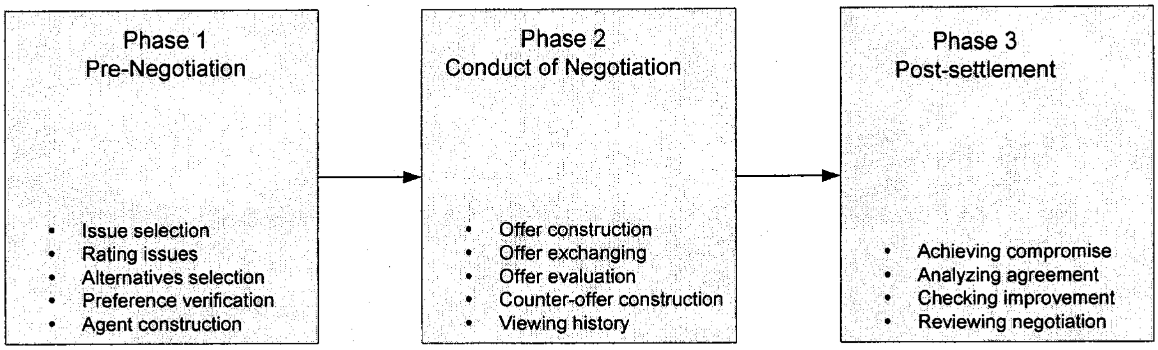


Figure 5: Activities in Phases of Negotiation Process

Pre-Negotiation:

Users or negotiation parties are capable of personalizing their negotiation plan. The e-negotiation system helps the user to understand the negotiation problem, choose the main negotiable issues such as price, deliver time, and warranty etc., and set the importance rate for each issue. Users also set the alternatives of the each issue they can accept and the rate of the acceptance. Send the initial information to each side and help the negotiation party to know the opponent. Both parties shake hand and get to know each other by exchanging messages. Personalized buyer agent and the sales agent are constructed and start to negotiate.

Conduct of Negotiation:

The sales agent provides an opening offer. There are pre-defined offers with certain format. The offers should show which issues are fixed and which issues are negotiable. The merchant-owner in the first phase selects the options for each of the negotiable issues. The shopping agent evaluates the goodness of the offer based on the strategy chosen by the buyer. A counter offer might be generated by the shopping agent. Both sides can generate offer or counter-offer in the process of the negotiation. The agents communicate very often in this phase. The messages they use are described by the pre-formatted XML.

These messages are exchanged among negotiation parties. Users may also review the history of the negotiation and revise their preference rating during the negotiation phase.

Post-settlement:

The outcome of negotiation is achieved and the system may act as a mediator. The E-negotiation system may help to assess compromise, perform efficiency analysis and provide the result to both sides. The buyer and the merchant-owner may or may not reach a final agreement. The result of the negotiation is stored by each side of negotiation parties. The relationship of negotiation parties is established. This may further facilitate similar negotiation between the same parties.

3.1.4 Structures

The generalized architecture for e-negotiation system is illustrated in Figure 6 and the architecture for an agent is illustrated in Figure 7. Shopping agent and sales agent exchange messages in the format of offers during negotiation. The users may supervise the negotiation process through interfaces. When an agreement is reached the goods may be sent to the buyer. In agent architecture negotiation provide the functions of offer evaluation and generation under the rule predefined by user. An agent communicates with other agents through its interface.

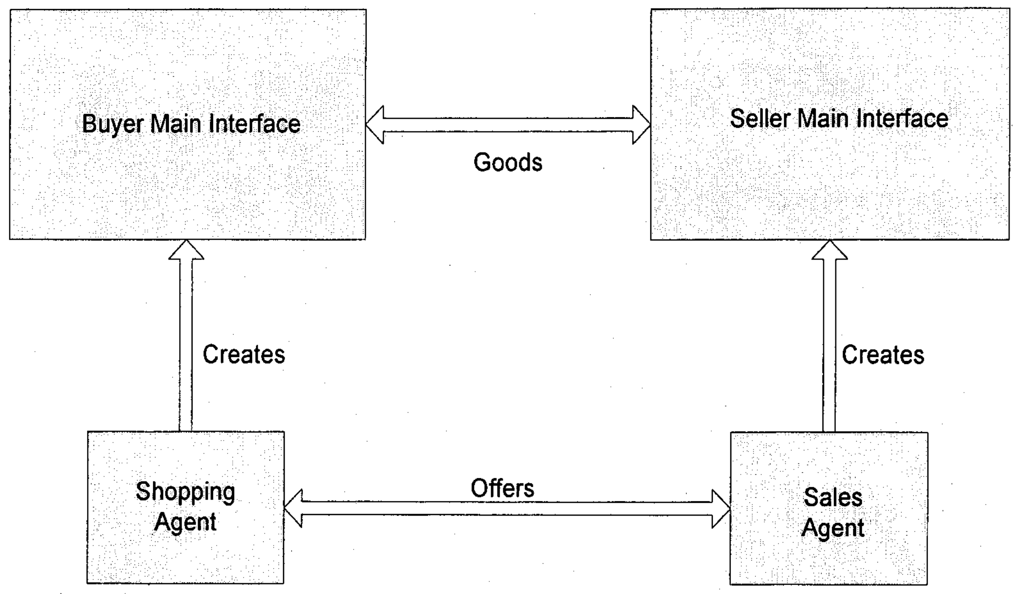


Figure 6: The Architecture of the E-negotiation System

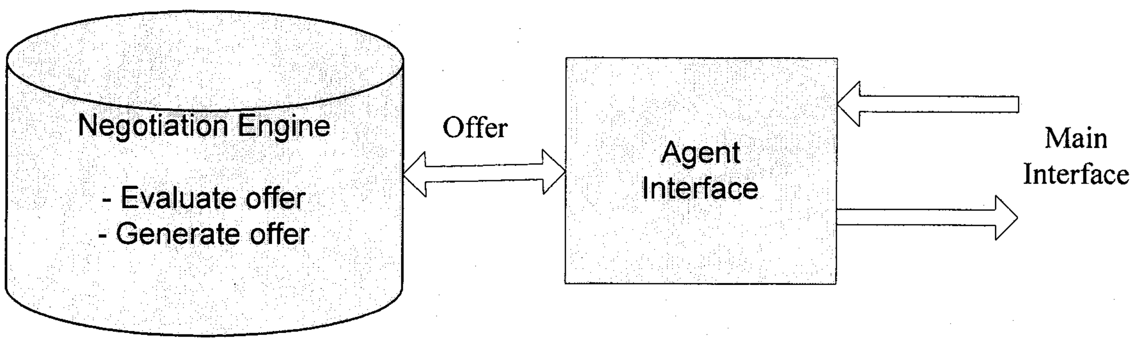


Figure 7: The Architecture of the Agent

The e-negotiation system contains three main parts: negotiation message, negotiation protocol and decision making. The relationship among these three parts is shown in Figure 8.

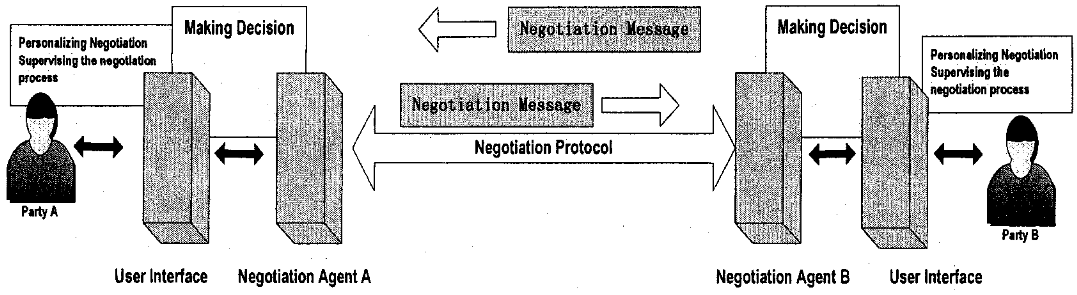


Figure 8: The structure of the e-negotiation system

Negotiation Message: is the medium for the conversation between the negotiation partners. The messages are "common language" between agents. The messages are written in XML with certain predefined format. The content of the messages indicate the intention of the agents. Figure 9 shows a simple example of the negotiation message of an offer.

```
<NegotiationMessage ID="5050" type="offer">
  <content goods="laptop">
    <issue>Price
      <alternative weight = 50>$5000</alternative>
      <alternative weight = 30>$4950</alternative>
      <alternative weight = 20>$4900</alternative>
      <alternative weight = 10>$4850</alternative>
    </issue>
    <issue>Warranty
      <alternative weight = 50> Three years
      </alternative>
      <alternative weight = 30> Two years
      </alternative>
      <alternative weight = 20>One year
      </alternative>
      <alternative weight = 10> No warranty
      </alternative>
    </issue>
    ...
  </content>
</NegotiationMessage>
```

Figure 9: An XML message of an offer

Negotiation Protocol: When implemented, it controls the negotiation workflow. The negotiation protocol consists of exchanges of negotiation messages that contain offers, counter-offers and etc. For example, a shopping agent may receive a proposal from a sales agent. Each proposal defines a complete product offering including a product configuration, price, and the merchant's value-added services. The shopping agent evaluates these proposals based on its owner's criteria and makes a suggestion. The buyer can still critique the offer along one or more dimensions. The customer's shopping agent continues negotiation until an agreement can be reached.

Decision making: It is a private decision process based on the different strategies selected by the user. Each negotiation party uses its own negotiation strategy. The system works as a mediator which does not have any preference to either side. The user's privacy is protected by the system. The other side may only view the common information of the agreement.

An e-negotiation system is explained in this section. It enables the personalized multi-issues negotiation between buyer and sales person. This system is constructed on the technique of software agent. The software agents may conduct the negotiation process of certain level autonomy.

3.2 On automating negotiation

In this thesis, the term "negotiation" refers to "automated negotiation". The automated negotiation means all parties involved are software agents while most current negotiation online still depends on human activities. Automated negotiation is a very complex process because it includes reasoning, proposals, and counter-proposals. In the future, all online negotiations will evolve to automated negotiations. There are several theoretical approaches for automated negotiations. In this section, an overview of three approaches will be analyzed, including game theory based negotiation, auction based negotiation, and multi-attribute utility theory based negotiation. These three approaches are very important in present eCommerce and will be widely used in the future business fields.

3.2.1 Game theory based bargaining

The game theory is widely acknowledged to provide a useful set of tools for the design of multi-agent architectures. Two main branches of game theory are distinguished as cooperative and non-cooperative. Non-cooperative game theory deals largely with how intelligent individuals interact with one another in an effort to obtain their own goals. The non-cooperative game theory will be discussed here.

To describe a game, one can list all players who attend the game; furthermore, each player will be provided by a list of alternative choices or strategies, and the play consists of selecting certain strategies by the players. The outcome of the play will be a pair of numbers representing the utilities of the players. The widely used equilibrium strategy is called Nash equilibrium and "dominant" strategy [17]. This strategy is optimal for all players independent of what the strategies of the other players are. Unfortunately, not every game follows a dominant strategy and the strategy in Nash equilibrium cannot always be interpreted as the most suitable one. There is a very famous example called prisoner's dilemma game [18]. This game demonstrates that game theory cannot always be applied to practical problems. Another limitation appearing by the application of game theory is that, at least in classical game theory, many frequently simplifying assumptions are made and such assumptions limit the practical applicability of the game theory results. For example, some private information such as fixed prices; preference for different features of the products is usually hidden from the opponent in real-life negotiation. The alternative methodologies on modelling the bargaining can be found in reference [19].

3.2.2 Auction based negotiation

Presently, the most effective way to solve "one–to-many" bargaining problems is the auction [20]. Four basic types of single auctions can be found: the ascending bid auction, the descending bid auction, the first price sealed bid auction, and the second price sealed bid auction.

In the ascending bid auction, the buyers keep raising the bids up to their reservation prices until only one buyer remains. In the standard terminology, the buyer's reservation price is the maximum price he/she is willing to pay for the item and the seller's reservation price is the lowest acceptable sale price for the item. If a buyer's maximum bid reaches or even exceeds the seller's reservation price, the transaction between the seller and the buyer will be completed. All buyers can see the highest bid and in some other auctions even can see the other buyer's bid. The rules of the ascending auction are easy to understand and easy to implement, this auction is the most popular type among all online auctions with or without agent support.

In the descending bid auction, [21] the seller will keep lowering the price of the item from the highest price. All buyers can see the current price and then decide if the price is still too high or if they can accept the price and purchase the item. The bidder will win the auction.

In the first price sealed bid auction, each buyer should submit in a secret bid, then all bidding prices will be opened and the item will be sold to the bid with the highest price. No one will be allowed to change or update the bid once they submit it.

In the second price sealed bid auction, the rules are similar with the first price sealed bid auction except the buyer with the highest bid will pay the price of the second highest bid.

Auction theory provides us with an understanding of the conditions as to which auction type should be the optimal solution. In the real world, for all types of the auctions above, all buyers are seldom equally informed, skilled, or perfectly rational, so that the result from the mathematical analysis of the auction cannot always be the best bidding in practice.

3.2.3 Multi-attribute utility theory

Some e-marketplaces have developed certain mathematical strategy for product and merchant brokering, so the consumer can decide what to buy or from whom to buy the item based on some preferences; however, most of the preferences are vague. In this case, the multi-attribute utility theory [22] is introduced to rank the crisp proposals coming from the merchants according to the consumer's vague preferences.

In multi-attribute theory, it is assumed that *m* participants take part in the negotiation and *n* issues will be used to characterize the negotiation subject. All *m* and *n* are of numerical nature. Let x_j^i indicate the value for issue *j* (*j* = [1, *n*]) and offered to the negotiation participant *i* (*i* = [1, m]). In general, an interval of values is accepted by each participant such as $a_j^i \le x_j^i \le b_j^i$. Different values in this interval will have different worth for each participant. The worth for each participant with respect to the values of negotiation issues is modeled by the function of scoring: $S_j^i : [a_j^i, b_j^i] \rightarrow [0,1], j = [1,n], i = [1,m]$. The bigger the value of a scoring function for certain value of an issue is, the more suitable value for the negotiation participant.

In a real situation, different negotiation issues have different level of importance for each negotiation participant. To model this property, the notion of relative importance is introduced as $\sum_{j=1}^{n} \omega_j^i = 1,\ i = [1, m]$. Now, assume that negotiation i is given an offer. The negotiation will be characterized by n issues. The offer then can be represented by a vector of dimension n ([$x = [x_1, x_n]$]). Using a linear function to model the scoring function, the scoring function can expressed as $S^i(x) = \sum_{j=1}^{n} \omega_j^i S_j^i(x),\ i = [1, m]$. This is the simplest example from the mathematical point of view. If all negotiators are using the linear scoring model, it will be possible to compute the optimal value of x by a given value as the best deal.

In real negotiation, most of cases are of non-linear properties. For example, two related negotiation issues may be taken into account together by certain negotiation participant, but considered separately by other negotiation participants. It will be very difficult to model this scenario by multi-attribute utility theory and find the optimal agreement for each participant.

The above popular theories have advantages and disadvantages for some real negotiation situations. However, they still have inherent problems. For example, auction based negotiation can not follow win-win policy to find mutual benefited agreement for both negotiation party, and the classical game theory model are not good for the open environment where the information is not complete and full of uncertainty. In this thesis, we introduce a new automated negotiation process. The above proposed negotiation process will help to optimize all advantages of the present negotiation theory and make the negotiation be easier and more user friendly.

The above proposed negotiation technology is derived from multi-attribute utility theory. We choose fuzzy inference system, to provide for vague information modeling and non-linear problem solutions.

Chapter 4

Fuzzy Logic Based Intelligent Negotiation Agent

In this chapter, the overview of the basics and core concepts of fuzzy logic is presented in Section 4.1. Our proposed model will be described in Section 4.2. There are three areas considered in designing model: negotiation issues, negotiation protocol, and reasoning model. The Last two areas will be described in Section 4.2.2 and Section 4.2.3 separately.

4.1 Basics of Fuzzy Logic

"*All traditional logic habitually assumes that precise symbols are being employed. It is therefore not applicable to this terrestrial life, but only to an imagined celestial one. The law of excluded middle is true when precise symbols are employed but it is not true when symbols are vague, as, in fact, all symbols are.*" Bertrand Russell (1923)

4.1.1 The dawn of fuzzy set

The conventional set theory was proposed by George Cantor (1845). . The conventional sets are crisp. The boundary of a set is rigid and well defined. However, things in the real world are rather fuzzy, uncertain, and vague rather than crisp.

For example, the dilemma posed by the ancient Greece caused serious problems to logicians and mathematicians. "When we take a grain from a heap, the heap is still there. But if we keep taking grains from the heap until one grain is left, is it still a heap?" The similar paradoxes are called "sorites". In conventional set theory, this problem is solved by setting a bound. A heap can be formed by more than *n* grains but it is not a heap when only *n*-1 grains are left. However, in common sense the border is not clear. It is obvious that the conventional set theory fails to describe realistically borderless cases.

The appearance of fuzzy sets solved the above problem. Lotfi Zadeh, the founder of fuzzy sets, introduced the concept of graded membership function. The graded membership function extended the characteristic function of conventional sets which only can take two values, 1 and 0, indicating whether an element is a member of a set.

A membership function of a fuzzy set A, $\mu_A(x)$, can take any value in the interval [0, 1]. Therefore, the concept of membership is not crisp any more. In this thesis, fuzzy sets are denoted by capital letters A, B, C … and corresponding membership functions by $\mu_A(x)$, $\mu_B(x)$, $\mu_C(x)$…

A fuzzy set A is defined by a set of ordered pairs.

$$A = \{(x, \mu_A(x)) | x \in A, \mu_A(x) \in [0,1]\} \tag{4.1}$$

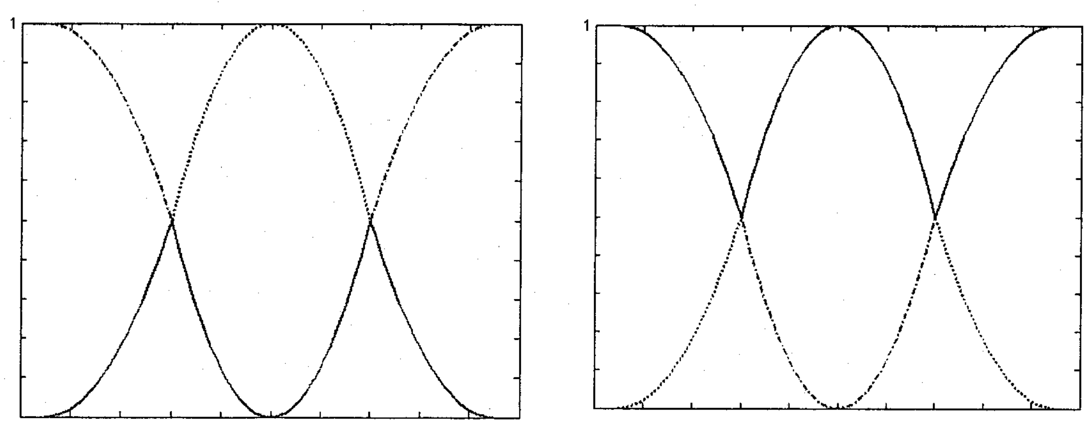
Definition (4.1) maps each element x to $\mu_A(x)$ in the interval [0, 1]. The larger the value of $\mu_A(x)$, the higher the degrees of membership. The fuzzy sets talked in the thesis are all convex and normalized.

The operations on fuzzy sets are listed in the following table. A and B are two fuzzy sets defined in the universe *U*.

Equality: $A = B$	$\mu_A(x) = \mu_B(x), x \in U$
Inclusion: $A \subseteq B$	$\mu_A(x) \le \mu_B(x), x \in U$
Proper subset: $A \subset B$	$\mu_A(x) \le \mu_B(x), x \in U$
Complementation: $\overline{A}$	$\mu_{\overline{A}}(x) = 1 - \mu_A(x), x \in U$
Intersection: $A \cap B$	$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)), x \in U$
Union: $A \cup B$	$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)), x \in U$

Table 1: Basic operations on fuzzy sets

The law of excluded middle, expressed by $A \cap \overline{A} = \phi$ and $A \cup \overline{A} = U$, is a critical property of classical sets. However it is not valid for fuzzy sets any more. It is illustrated in Figure 10 by means of Venn diagrams.



$A \cap \overline{A} \neq \phi$ $A \cup \overline{A} \neq U$

Figure 10: The law of excluded middle is invalid for fuzzy sets

Fuzzy sets are more general and flexible than classical sets because of the invalidity of the law of excluded middle [24]. They are more suitable for describing vagueness and processes with incomplete and imprecise information.

4.1.2 Fuzzy logic

Fuzzy logic can be seen as a generalized classical logic. Classical logic (two-valued logic) concerns propositions that are either true or false. The truth value set T_2 of classical logic has two elements: 0 representing false and 1 representing true. In 1923, Lukasiewicz introduced many-valued logic where the truth value set T_n has more than two elements besides 0 and 1 [25]. By using fuzzy sets and fuzzy relations in the system of many-valued logic, fuzzy logic is derived from many-valued logic. The relations among classical sets, classical logic, fuzzy sets, many-valued sets and fuzzy logic are illustrated in Figure 11.

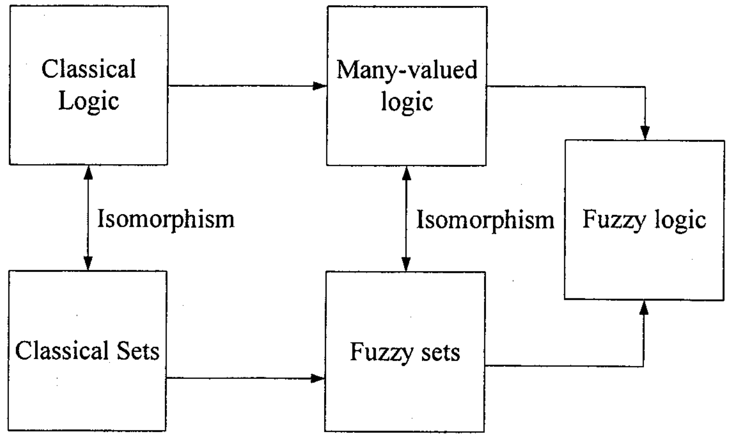


Figure 11: Evolvement of fuzzy logic [24]

A methodology is given by fuzzy logic for treating linguistic variables and expressing modifiers like *very*, *fairly*, *not* and so on. Fuzzy logic makes reasoning with imprecise and vague propositions which deal with the natural language easily. It reflects both the rightness and vagueness of natural language in common-sense reasoning.

Linguistic variable

Linguistic variables are variables which take values of words or sentences in natural or artificial languages [24]. For example, *age* is a word in natural language. Let *age* be a linguistic variable taking values from a set of words {*very young*, *young*, *middle age*, *old*, *very old*}. These values are called *terms* of linguistic variable *age* and described by fuzzy sets with corresponded membership functions on a universal set $U \subset R^{+}$ or called *operating domain*. Figure 12 describes the linguistic variable age on the universal set $U =$ [0, 100] i.e. operating domain of x.

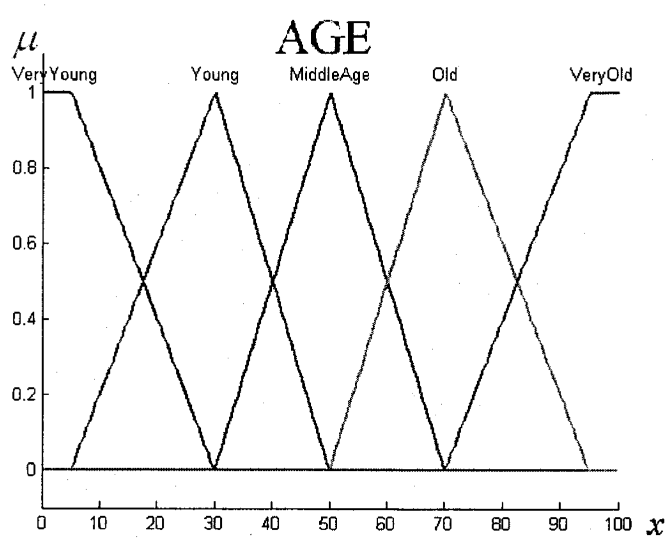


Figure 12: Terms of the linguistic variable *age*

Composition rules

Composition is an operation occurring between two propositions p and q joined by logical connectives. The proposition p states that x is A, and proposition q states that y is B, where A is a fuzzy set $A = \{(x, \mu_A(x)) | x \in A \subset U_1\}$, B is a fuzzy set $B = \{(y, \mu_B(y)) | y \in B \subset U_2\}$. Membership functions ($\mu_A(x)$, $\mu_B(y)$) represent the truth values of the propositions p and q. The compositions are listed in the Table 2.

Conjunction: $p \wedge q$	$\mu_{A \dot{\times} B}(x, y) = \min(\mu_A(x), \mu_B(y)), (x, y) \in A \times B$
Disjunction: $p \vee q$	$\mu_{A \dot{\times} B}(x, y) = \max(\mu_A(x), \mu_B(y)), (x, y) \in A \times B$
Implication: $p \to q$	$tr(\text{if p then q}) = \min(1, 1 - \mu_A(x) + \mu_B(y)), (x, y) \in A \times B$
Complementation: $\overline{A}$	$\mu_{\overline{A}}(x) = 1 - \mu_A(x), x \in U$
Intersection: $A \cap B$	$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)), x \in U$
Union: $A \cup B$	$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)), x \in U$

Table 2: The composition rules for fuzzy propositions

4.1.3 Fuzzy expert (inference) system

An Expert system is defined as a type of application program that makes decisions or solves problems in a particular field by using knowledge and analytical rules defined by experts in the field [26]. A fuzzy expert system (fuzzy inference system) is an expert system that uses fuzzy logic instead of conventional Boolean logic [27]. In other words, a fuzzy expert system is developed for reasoning based on a collection of membership functions and inference fuzzy rules. Unlike conventional expert systems, which are mainly symbolic reasoning engines, fuzzy expert systems are oriented toward numerical processing. The decisions are made by using and aggregating inferential rules. The counterpart of fuzzy system in industrial engineering is fuzzy logic control system.

There are five steps to model a problem using fuzzy expert system.

Step 1 -- Define linguistic variables described by fuzzy sets (Fuzzification)

Step 2 -- Define inference fuzzy rules

Step 3 -- Evaluate each case for all inference fuzzy rules

Step 4 -- Combine information from inference rules (Rule aggregation)

Step 5 -- Defuzzify the crisp values of outputs

The generic modelling process is depicted in Figure 13. The function of each block is explained in following sections using a two inputs and one output example.

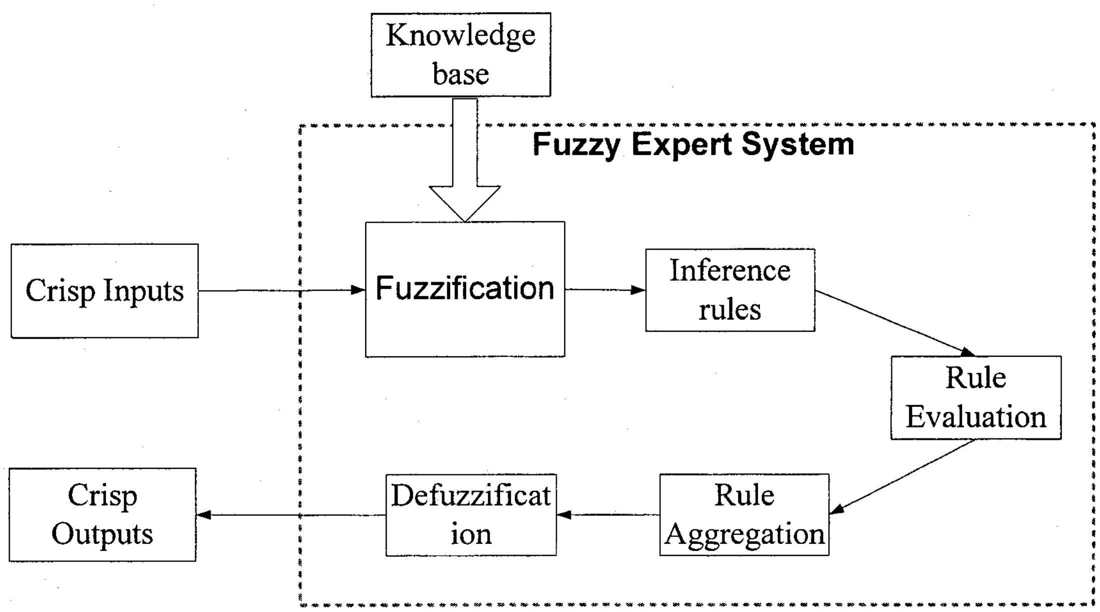


Figure 13: Block diagram of a generic fuzzy expert system

Fuzzification

Any system has inputs and outputs. In fuzzy inference system inputs and outputs are linguistic variables. These linguistic variables are modeled by fuzzy sets A, B, C:

$$\begin{aligned} A &= \{A_1,..,A_i,...A_n\}, \\ B &= \{B_1,..,B_j,...B_m\}, \qquad (4.2) \\ C &= \{C_1,..,C_k,...C_l\} \end{aligned}$$

The terms A_i, B_j and C_k are fuzzy sets determined by the corresponding membership functions $\mu_{A_i}(x), \mu_{B_j}(y), \mu_{C_k}(z)$. We assume that A and B are two inputs of system and C is only output of system.

The design of linguistic variables needs knowledge about inputs, outputs and fuzzy partitions. The design steps are listed in below.

- Find the universals U_1, U_2, U_3 of bases variable x, y, z for the linguistic variables A, B, C i.e. quantize the values of linguistic variables.
- Select the reasonable membership functions for the terms A_i, B_j, and C_k. Triangular and trapezoidal membership functions are most popular choices.
- Determine the number of terms in (4.2) i.e. n, m, and l.

Inference rules

Inference rules are a set of "if … and ... then" rules. If there are n terms in A and m terms in B the total number of the inference rules is nm. The rules make l different outputs. A typical rule is shown in the following.

If x is A_i and y is B_j then z is C_k. (4.3)

The inference rules can be summarized into a table called decision table (Table 3) where C_{ij} renamed C_k is an element of the set $\{ C_1,..,C_k,...C_l \}$.

	B_1	$\cdots$	B_j	$\cdots$	B_m
A_1	C_{11}	$\cdots$	C_{1j}	$\cdots$	C_{1m}
$\vdots$	$\vdots$		$\vdots$		$\vdots$
A_i	C_{i1}	$\cdots$	C_{ij}	$\cdots$	C_{im}
$\vdots$	$\vdots$		$\vdots$		$\vdots$
A_n	C_{n1}	$\cdots$	C_{nj}	$\cdots$	C_{nm}

Table 3: Decision table for two input and single output fuzzy rule based system

Let propositions p_i, q_j and r_k represent "x is A_i", " y is B_j", "and z is C_k" respectively. Rewrite (4.3) as

If x is p_i and y is q_j then z is r_k. (4.4)

The "and" part is called *precondition*. The truth of the precondition is defined to be composition conjunction (Table 2).

$$p_i \wedge q_j = \min(\mu_{A_i}(x), \mu_{B_j}(y)), (x, y) \in A \times B \subset U_1 \times U_2 \quad (4.5)$$

The truth value of the precondition is computed by Equation (4.5). The "if … then …" part is implication (Table 2). The truth value of the rule can be presented in the following equation.

$$p_i \wedge q_j \wedge r_k = \min(\mu_{A_i}(x), \mu_{B_j}(y), \mu_{C_k}(z)), (x, y, z) \in A \times B \times C \subset U_1 \times U_2 \times U_3 \quad (4.6)$$

In a fuzzy expert system, inference rules stem from the knowledge of human experts, the preference of clients, or common sense of everyday life. They can be redesigned at anytime when knowledge base changes.

Rule evaluation

When specified crisp inputs x_0 and y_0 are observed, measured or estimated, we have to give a corresponding output z. These crisp inputs have to be matched against related membership functions to find corresponding fuzzy inputs. This is expressed in Figure 14 where the crisp input x_0 is mapped into fuzzy inputs $\mu_{A_i}(x_0)$ and $\mu_{A_{i+1}}(x_0)$, crisp input y_0 is mapped into fuzzy inputs $\mu_{B_j}(y_0)$ and $\mu_{B_{j+1}}(y_0)$.

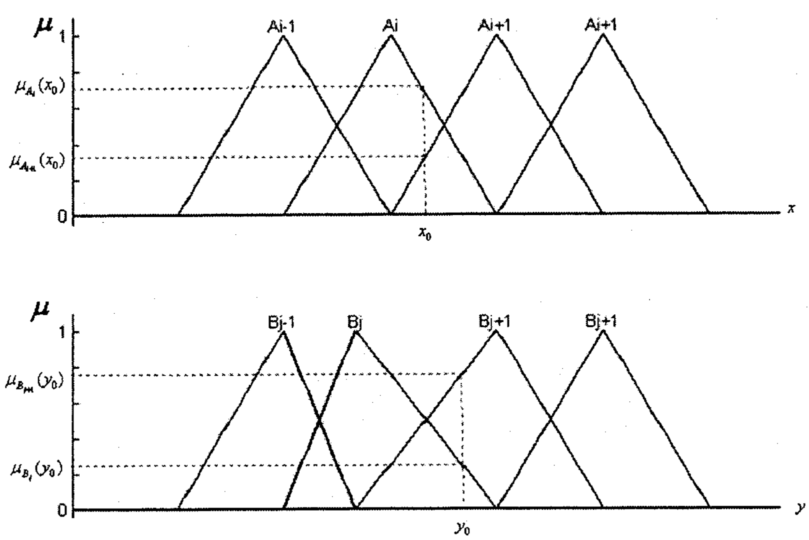


Figure 14: Fuzzy inputs corresponding to crisp inputs

The decision table reduces to four non-zero cells with specific crisp inputs $x = x_0$ and $y = y_0$.

	$\cdots$	$\mu_{B_j}(y_0)$	$\mu_{B_{j+1}}(y_0)$	$\cdots$
$\vdots$	**0**	**0**	**0**	0
$\mu_{A_i}(x_0)$	**0**	$\mu_{C_{ij}}(z)$	$\mu_{C_{ij+1}}(z)$	0
$\mu_{A_{i+1}}(x_0)$	**0**	$\mu_{C_{i+1j}}(z)$	$\mu_{C_{i+1j+1}}(z)$	0
$\vdots$	0	0	0	0

Table 4: Reduced decision table with respect to specific inputs x_0, y_0

The four rules represented by the four active cells are *fired*.

<u>*Rule aggregation*</u>

Rule aggregation is the methodology to find the final fuzzy output. We will state the process of rule aggregation by using the previous four fired rules. The truth values of the four preconditions are computed by the following equations, where α_{ij} is called the *strength* of the rule$_{ij}$.

$$\begin{aligned} \alpha_{ij} &= \min(\mu_{A_i}(x_0), \mu_{B_j}(y_0)) \\ \alpha_{ij+1} &= \min(\mu_{A_i}(x_0), \mu_{B_{j+1}}(y_0)) \\ \alpha_{i+1j} &= \min(\mu_{A_{i+1}}(x_0), \mu_{B_j}(y_0)) \\ \alpha_{i+1j+1} &= \min(\mu_{A_{i+1}}(x_0), \mu_{B_{j+1}}(y_0)) \end{aligned} \qquad (4.7)$$

The fuzzy output of each rule is defined in the following equation.

$$\begin{aligned} \alpha_{ij} \wedge \mu_{C_{ij}}(z) &= \min(\alpha_{ij}, \mu_{C_{ij}}(z)) \\ \alpha_{ij+1} \wedge \mu_{C_{ij+1}}(z) &= \min(\alpha_{ij+1}, \mu_{C_{ij+1}}(z)) \\ \alpha_{i+1j} \wedge \mu_{C_{i+1j}}(z) &= \min(\alpha_{i+1j}, \mu_{C_{i+1j}}(z)) \\ \alpha_{i+1j+1} \wedge \mu_{C_{i+1j+1}}(z) &= \min(\alpha_{i+1j+1}, \mu_{C_{i+1j+1}}(z)) \end{aligned} \qquad (4.8)$$

These fuzzy outputs are combined together to produce one aggregated fuzzy output with membership function.

$$\mu_{agg}(z) = \max((\alpha_{ij}, \mu_{C_{ij}}(z)), (\alpha_{ij+1}, \mu_{C_{ij+1}}(z)), (\alpha_{i+1j}, \mu_{C_{i+1j}}(z)), (\alpha_{i+1j+1}, \mu_{C_{i+1j+1}}(z))) \quad (4.9)$$

$\mu_{agg}(z)$ is the combination of clipped membership functions: $\mu_{C_{ij}}(z)$, $\mu_{C_{ij+1}}(z)$, $\mu_{C_{i+1j}}(z)$ and $\mu_{C_{i+1j+1}}(z)$. In order to find the crisp output we have to defuzzify the aggregated fuzzy output.

Defuzzification

There are several methods to perform the operation defuzzification. Hellendoorn and Thomas [28] have described and analyzed six defuzzification methods. Most of methods consider the shape of $\mu_{agg}(z)$ and height of the clipped membership functions and complexity of computation.

We are going to talk about three popular methods: centroid of area method (CAM), mean of maximum method (MMM), and height defuzzification method (HDM).

Assume $\mu_{agg}(z)$ is shown in Figure 15.

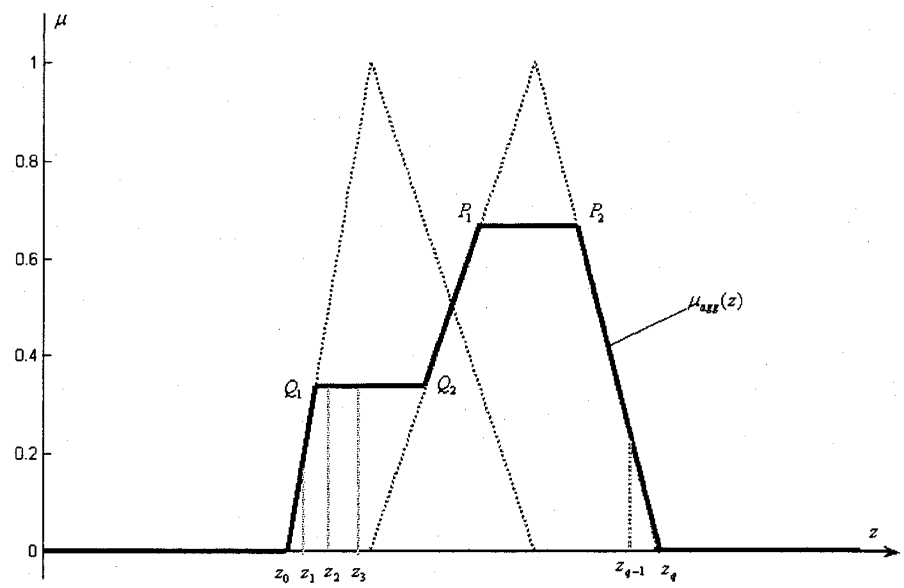


Figure 15: An example of aggregated fuzzy output

Centroid of area method (CAM)

Suppose the result membership function $\mu_{agg}(z)$ is in the interval $[z_0, z_q]$. Subdivide the interval into q equal subintervals by the points: z_1, z_2, z_3, ... , z_{q-1}. The crisp output $\hat{z}_{CAM}$ is computed by Equation (4.10).

$$\hat{z}_{CAM} = \frac{\sum_{k=1}^{q-1} z_k \mu_{agg}(z_k)}{\sum_{k=1}^{q-1} \mu_{agg}(z_k)} \quad (4.10)$$

$\hat{z}_{CAM}$ is the centroid of the area under curve $\mu_{agg}(z)$ in geometry.

Mean of maximum method (MMM)

There are two flat segments P_1P_2 and Q_1Q_2. Suppose the projection of segment P_1P_2 with maximum height is z_{P_1} z_{P_2} . The crisp output $\hat{z}_{MMM}$ is computed by Equation (4.11).

$$\hat{z}_{MMM} = \frac{z_{P_1} + z_{P_2}}{2} \quad (4.11)$$

This method is the simplest one, but is not very accurate.

Height defuzzification method (HDM)

Height defuzzification method is a generalization of mean of maximum method. It uses all clipped flat segments of $\mu_{agg}(z)$. The crisp output $\hat{z}_{HDM}$ is produced by the following equation.

$$\hat{z}_{HDM} = w_1 \frac{z_{P_1} + z_{P_2}}{2} + w_2 \frac{z_{Q_1} + z_{Q_2}}{2} \quad (4.12)$$

Where

$$w_1 = \frac{\mu_{agg}(z_{P_1})}{\mu_{agg}(z_{Q_1}) + \mu_{agg}(z_{P_1})}$$
$$w_2 = \frac{\mu_{agg}(z_{Q_1})}{\mu_{agg}(z_{Q_1}) + \mu_{agg}(z_{P_1})} \quad (4.13)$$

If there are more than two segments, Equations (4.12) and (4.13) can be extended accordingly. HDM method can be seen as a simplified version of CAM or a generalized version of MMM. In some places the mean of maximum method refers to HDM and the largest of maximum method (LOM) refers to MMM. Trade-off is made between computational complexity and accuracy, when we choose a defuzzification method. CAM is used in the implementation discussed in Chapter 5 because it can achieve most accurate results.

4.2 Proposed Model of Fuzzy Logic Based Intelligent Negotiation Agent

4.2.1 Introduction

Software agent technology is widely used in agent-based eCommerce. These software agents have a certain degree of intelligence, i.e. they can make their own decisions. These software agents interact with other agents to achieve certain goals. However, software agents can not directly control other agents because every agent is an independent decision maker. Negotiation becomes the necessary method to achieve mutual agreement between agents.

This thesis focuses on modelling multi-issue one-to-one negotiation agent for a third party driven virtual marketplace. We consider one-to-one negotiation because it is the characteristic of individual negotiation and because it allows cooperative negotiation which is not suitable for many-to-many auction based negotiations. The above defines the scope of our proposed fuzzy logic based intelligent negotiation agent model..

When building autonomous negotiation agents which are capable of flexible and sophisticated negotiation, three broad areas need to be considered:

- ***Negotiation protocols*** - the set of rules which govern the interaction,
- ***Negotiation issues*** - the range of issues over which agreement must be reached,
- ***Agent reasoning models*** – the agents employ to act in line with the negotiation protocol in order to achieve their negotiation objectives.

4.2.2 Proposed bilateral negotiation protocol

The following diagram shows a general example of bilateral negotiation protocol.

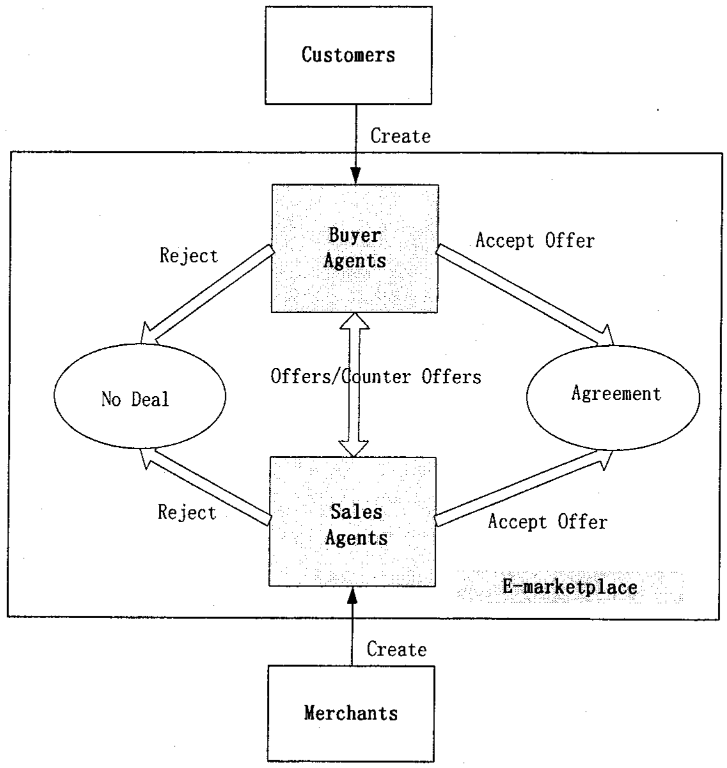


Figure 16: A general example of bilateral negotiation protocol

The above protocol is very straightforward. For example, a customer wants to buy a TV set. He/she logs into the e-marketplace and inputs his/her requirements for the TV. This buyer input creates a software buyer agent in the above e-marketplace system. The buyer agent will help the customer to find the possible vendors (by interacting with the sales agents created by the merchants in the above e-marketplace system) and bargain with the sales agent by exchanging the offer and counter offers. The buyer agent may receive offers/proposals from sales agents. Each proposal defines, for example, a complete product offering including the size of the TV set, brand, price, the quality of the TV and return policy etc. The buyer agent evaluates these proposals based on its buyer's criteria and makes a suggestion (counter offer). The buyer's buyer agent continues negotiation until an agreement can be reached. On the other hand if the buyer and seller agents reject the current incoming offer criteria, conditions the negotiation terminates at the 'no deal' state.

The above general negotiation protocol can be separated into three sub-processes: pre-negotiation, negotiation, post-negotiation. The detailed procedures of the three parts are shown in the following flow charts. There exists overlaps between each two processes which make the whole system can be merged into one system.

Pre-negotiation:

The process of the pre-negotiation is complex. Both sides need to be well prepared for the negotiation. They are getting to know each other and know the objective of the other party. Their final goal is to achieve not a win-lose result but a win-win result.

Here Party A represents the buyer and Party B represents the vendor.

The requirements for the vendors are for example verify buyers' credit, appropriate buyer incentives to quickly conclude the deal, etc. If the negotiation can not be concluded, because, for example, the cost of the product can not be agreed upon, the negotiation process will jump to the "No deal" state which is defined in the post-negotiation part. The detailed process is shown in Figure 17.

Negotiation:

If both sides agree to negotiation the negotiation process starts. The seller agent gives the initial offer to which the vendor has 100% acceptance. The buyer agent evaluates the initial offer and has three choices reject, agree or counter. Anytime the agent chooses reject the process will jump to the "No deal" state. On the other words each side can terminate the negotiation in the middle of the negotiation. If buyer/seller agent accepts the offer, the process will jump to the start of the "post-negotiation". Otherwise, the buyer and the seller agents exchange the counter offers they can accept until one side makes a decision to accept or reject. The flowchart of negotiation process is shown in Figure 18.

Post-negotiation:

The seller agent and buyer agent stop negotiation here. Agreement or "no deal" state will be achieved. The seller agent gives the final agreement where all final results of negotiation issues are listed. The human buyer has a chance to reject the final agreement. The flowchart of post-negotiation process is shown in Figure 19.

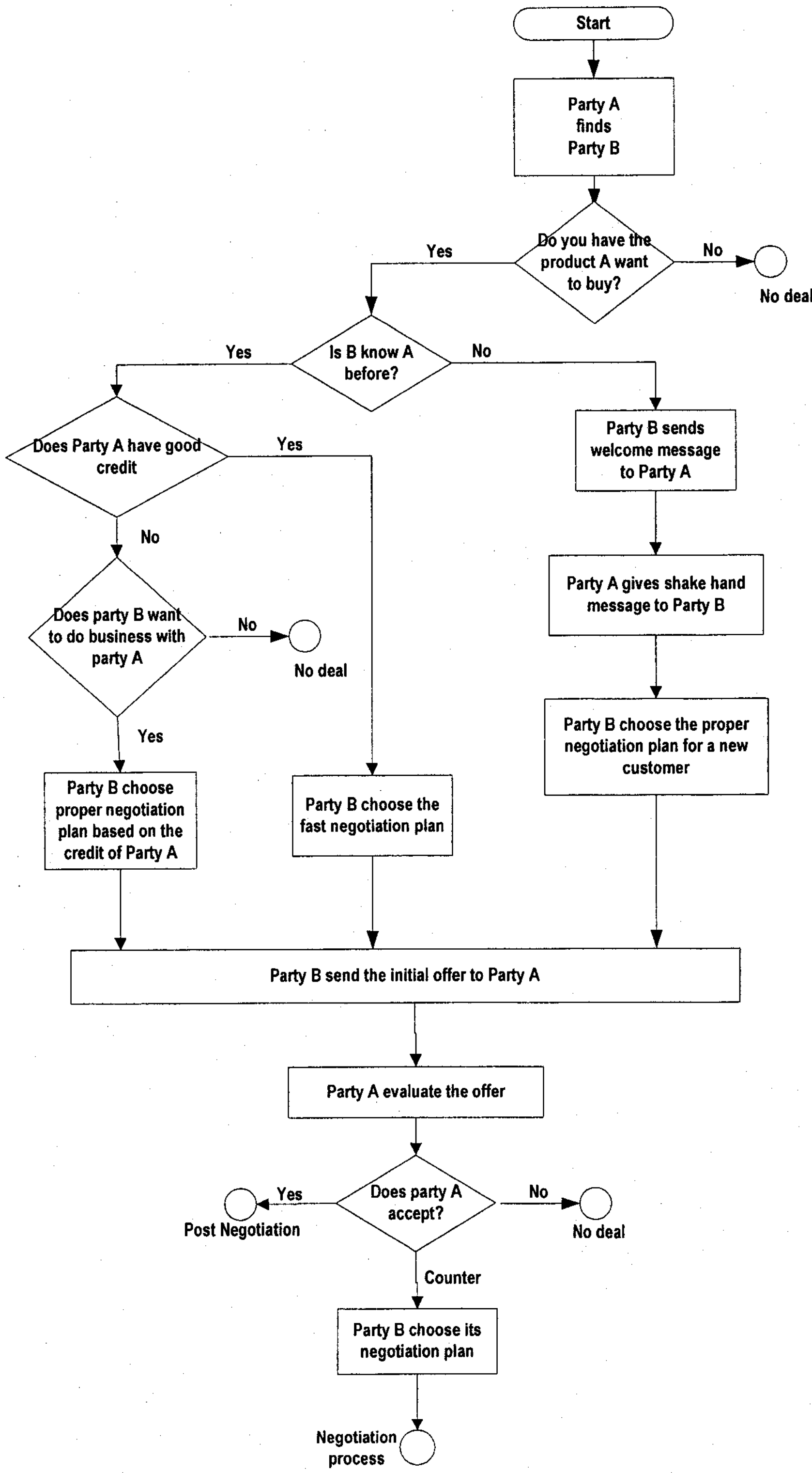


Figure 17: The flowchart of pre-negotiation process

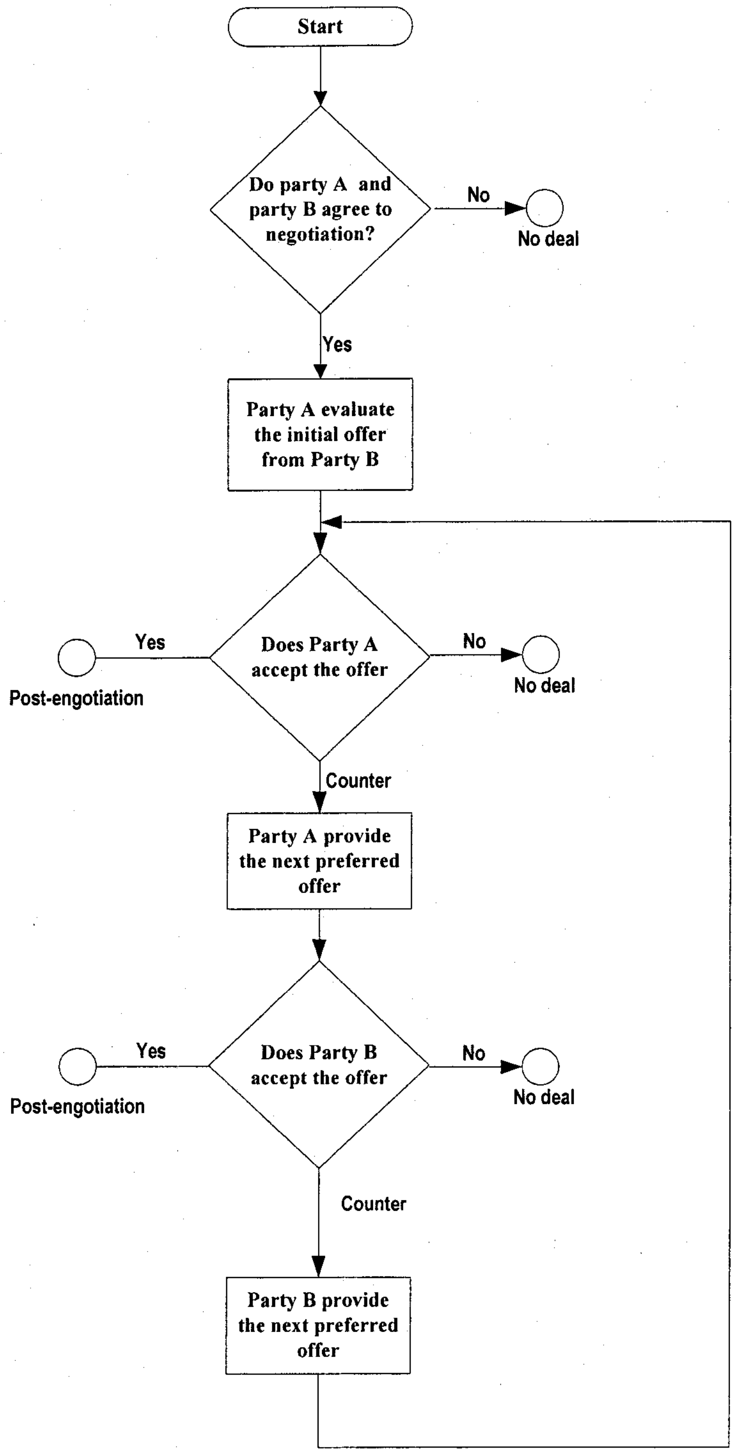


Figure 18: The flowchart of negotiation process

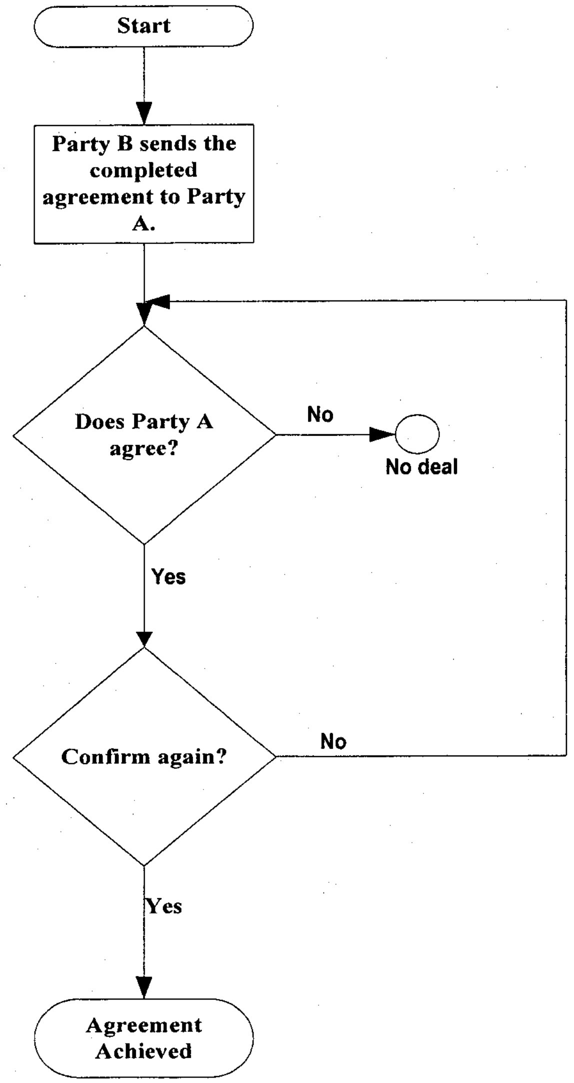


Figure 19: The flowchart of post-negotiation process

In order to make the proposed bilateral negotiation process suitable for a practical marketplace, for example, reliability, security, customization, etc., issues should be considered. For example, how to generate the buyer agent that fully represents, the buyers requirements, identify the seller agents, handle a run-away scenario. The possible solutions for the security requirements will be discussed in Chapter 5.

4.2.3 Proposed Reasoning Model

The intelligence of the negotiation agent is realized by the reasoning model. The reasoning component, in particular, concentrates on the processes of generating initial offer/counter offers, of evaluating the incoming offers, and of making decisions to interact with the negotiation opponent. Figure 20 shows the proposed reasoning model for a negotiation agent.

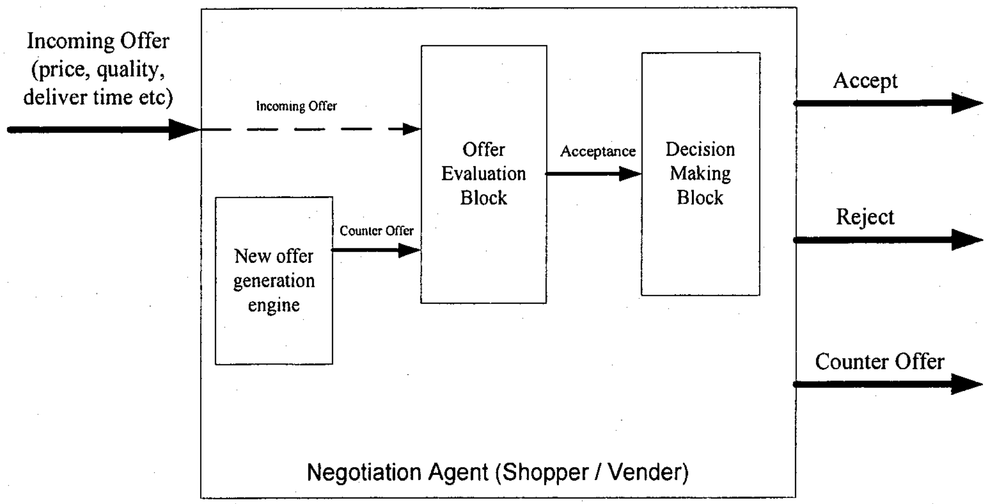


Figure 20: The reasoning model of negotiation agent

This reasoning model aims at the negotiation process. The process of matching and hand shaking in a pre-negotiation process has been solved in several papers [29]. We assume that the buyer agent and vendor agent have roughly matched their similarity and start a negotiation on issues which they have not reached agreement. In a certain round of negotiation, the negotiation agent can pre-prepare a counter offer for the next round of negotiation. The counter offer is generated by the ***new offer generation engine***. And both the incoming offer from the opponent negotiation agent and the counter offer sent to the

offer evaluation block. It does the analysis of the offer and calculates the degree of satisfaction (acceptance of the agent) of incoming offer and the counter offer. The result is scaled over the range from 0 to 100. Finally, the ***decision making block*** makes the decision. It would be acceptation of the current incoming offer, rejection of the current incoming offer, or counter offer.

<u>*New offer generation engine*</u>

The modeling of the counter offer generation engine can be seen as Distributed Fuzzy Constraint Satisfaction Problem (DFCSP). The functions that generated new offers are called *tactics*. The ways to tune the weights of different tactics are called *strategies*. There are lots of works that have been done in tactics and strategies [30].

The main object of designing tactics is how to find ways to make concession. Each agent has its self-interested *utility functions* (U^i_n, where *i* denotes the issue considered in the utility function and *n* denotes the current negotiation round number) of different issues under negotiation. There are two types of utility functions: continuous and discrete. The continuous utility function describes the agent preference of certain issue whose value can be expressed in a continuous real number interval. The discrete utility function is used to present the agent preference of negotiation issue which has finite sets of values such as the delivery destination.

The overall utility equation is shown below.

$$U = \frac{\sum_i w_i U_i(x_i)}{\sum_i w_i} \qquad (4.14)$$

where, the x_i denotes the value of issue i and the U denotes overall utility and $U_i(x_i)$ denotes utility of issue i, $U_i \in [0,1]$. $w_i \in [0,9]$ is the preference degree of agent on issue i. Large value of w_i means the issue is very important to the negotiation agent.

The continuous utility function can be broken down into benefit-oriented and cost-oriented.

For benefit-oriented, the $U_i(x_i)$ increases when x_i increases. Assume the maximum negotiation space for issue i is $[X^i_{\min}, X^i_{man}]$.

$$U_i(x_i) = \frac{x_i - X^i_{\min}}{X^i_{man} - X^i_{\min}} \quad (4.15)$$

For cost–oriented, the $U_i(x_i)$ decreases when x_i increases. Assume the maximum negotiation space for issue i is $[X^i_{\min}, X^i_{man}]$.

$$U_i(x_i) = 1 - \frac{x_i - X^i_{\min}}{X^i_{man} - X^i_{\min}} \quad (4.16)$$

The discrete utility function is used for qualitative issues. It is defined by a set of user preferences. For instance, the utility function of possible delivery destinations is defined as

U {Orleans, Ottawa, Nepeans, Kanata} = {0.4, 1, 0.6, 0.2}

The major objective of a new offer generating engine is to calculate the concession value (Δx_i) for the next round of negotiation.

First step is to find the concession space (CS).

$$CS^A{}_{i,n} = \frac{10 - w^A{}_i}{(10 - w^A{}_i) + (10 - w^B{}_i)} \times \left| x^A{}_{i,n} - x^B{}_{i,n} \right| \quad (4.17)$$

Where, A and B are two agents under negotiation. $\left| x^A{}_{1,n} - x^B{}_{1,n} \right|$ denotes the distance between A and B's offers on issue *i*. *n* is the number the current negotiation round. Note the $x^A{}_{1,n}$ and $x^B{}_{1,n}$ for discrete issues refer to the level of the preference of certain option. $w^B{}_i$ of agent A is unknown. Therefore, a guess is required based on the previous behavior of negotiation partner.

$$w^B{}_{i,n} = \begin{cases} \min(w^B{}_{i,n-1} + 0.5, 9), r < 1 \\ \min(w^B{}_{i,n-1} - 0.5, 0), r > 1 \\ w^B{}_{i,n-1} \qquad , r = 1 \end{cases} \quad (4.18)$$

Where

$$r = \frac{\left| x^B{}_{i,n} - x^B{}_{i,n-1} \right|}{\left| x^A{}_{i,n} - x^A{}_{i,n-1} \right|} \quad (4.19)$$

Second step a concession is made by randomly choosing a valid value in the interval of CS got from step one. The new offer value of agent A on issue *i* is generated by

$$x^A{}_{i,n+1} = x^A{}_{i,n} + \Delta x^A{}_{i,n} \quad (4.20)$$

The tactics discussed above is behavior-dependent tactics [30]. The behavior-dependent tactics have proved important in cooperative problem solve negotiation [31]. Therefore they are good for finding a win-win solution.

Offer evaluation block

The model of offer evaluation block can be seen as a fuzzy expert system. The main reason for using a fuzzy logic based expert system is the human preferences are vague and uncertain. To evaluate the acceptations of the incoming offer is too complicated by using conventional mathematical methodology. When the number of negotiation issues grows, the complicity of conventional model increases dramatically. A detailed example is going to be given in the next chapter.

Decision block

To simplify the function of the decision block, it can be concluded that

$$\begin{cases} ACCEPTANCE_{incom} < ACCEPTANCE_{\min} & \text{REJECT} \\ ACCEPTANCE_{incom} \geq ACCEPTANCE_{counter} & \text{ACCEPT} \\ ACCEPTANCE_{incom} < ACCEPTANCE_{counter} & \text{COUNTER OFFER} \end{cases}$$

If the acceptance of the incoming offer is less than the minimum acceptance of the agent, then reject and terminate the negotiation. If the acceptance of the incoming offer is greater than the acceptance of the counter offer prepared by the agent, then accept the offer to achieve agreement with the opponent agent. Otherwise, if the acceptance of the incoming offer is less than the counter offer then send the counter offer to the opponent negotiation agent.

However, this decision making block is too simple to avoid an impasse in a negotiation process. We assume that the negotiation process is controlled by two external constraints: the maximum duration of negotiation and the maximum number of negotiation rounds. These two constraints are pre-defined by human users before the negotiation agent is built. Applying these two constraints to the negotiation process will avoid the endless bargaining between negotiation participants.

Chapter 5

Implementation and Performance Testing

In this chapter, we describe an implement example of the offer evaluation model based reasoning model with certain assumptions in Section 5.1. . The test results obtained from two implementations of the same offer evaluation model: one is Microsoft .NET based implementation and the other is Matlab implementation is described in Section 5.2. Finally, safety and security issues of future implementation are discussed in Section 5.3.

5.1 Implementation of the offer evaluation model

Figure 21 illustrates the overview of the buyer offer evaluation block we implemented. It takes two inputs as the controlling variables. One is the incoming offer *price* and the other is the *quality level* of the offering product. The quality level is scaled over the range from 0 to 100. It is supposed to be defined by the commercial experts.

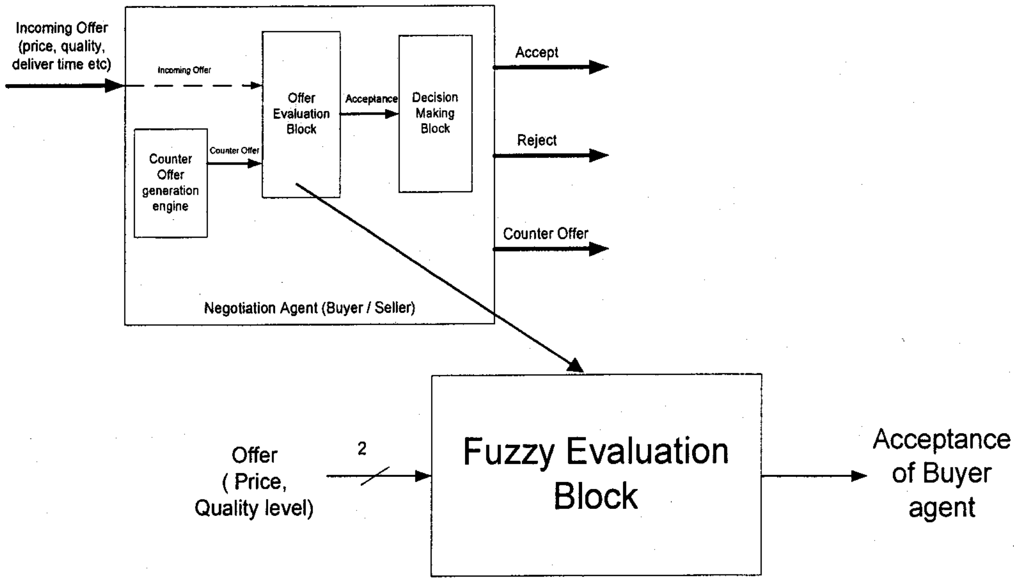


Figure 21: The implemented model of offer evaluation block

5.1.1 Fuzzification

The inputs and output are modeled by fuzzy sets with three terms listing in below.

Input 1: *Offer price* $=A=$ {**Low, Medium, High**};

Input 2: *Quality Level* $=B=$ {**Low, Medium, High**};

Output 1: *Acceptance* $=C=$ {**Low, Moderate, High**};

Let H express the value the buyer think is absolute high, L express the value the buyer think is absolute low, and M be the middle point of L and H. The membership functions of inputs and outputs are similarly described by the triangular and trapezoidal functions presenting in the following equations.

$$\mu_{LOW}(x) = \begin{cases} 1 & 0 \le x \le L \\ \dfrac{M-x}{M-L} & L \le x \le M \end{cases} \quad (5.1)$$

$$\mu_{MEDIUM}(x) = \begin{cases} \dfrac{x-L}{M-L} & L \le x \le M \\ \dfrac{H-x}{H-M} & M \le x \le H \end{cases} \quad (5.2)$$

$$\mu_{MEDIUM}(x) = \begin{cases} \dfrac{x-M}{H-M} & M \le x \le H \\ 1 & H \le x \end{cases} \quad (5.3)$$

Based on the functions (5.1), (5.2) and (5.3), the membership functions of each component are shown in the following Figures 22, 23.

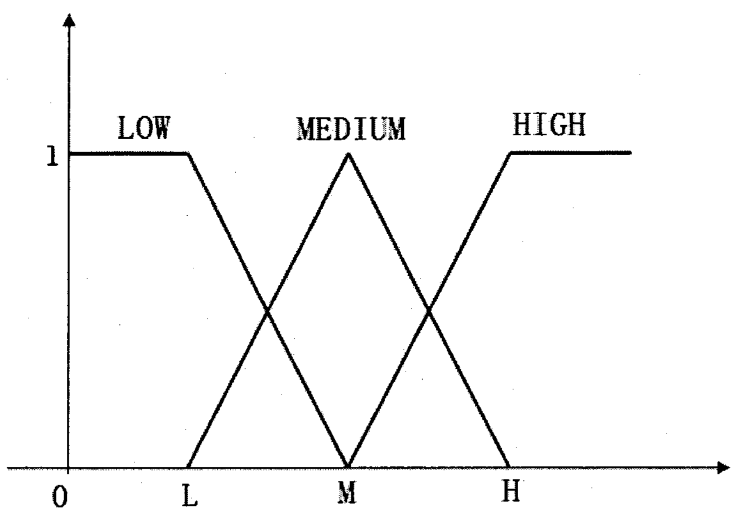


Figure 22: The membership functions of the inputs

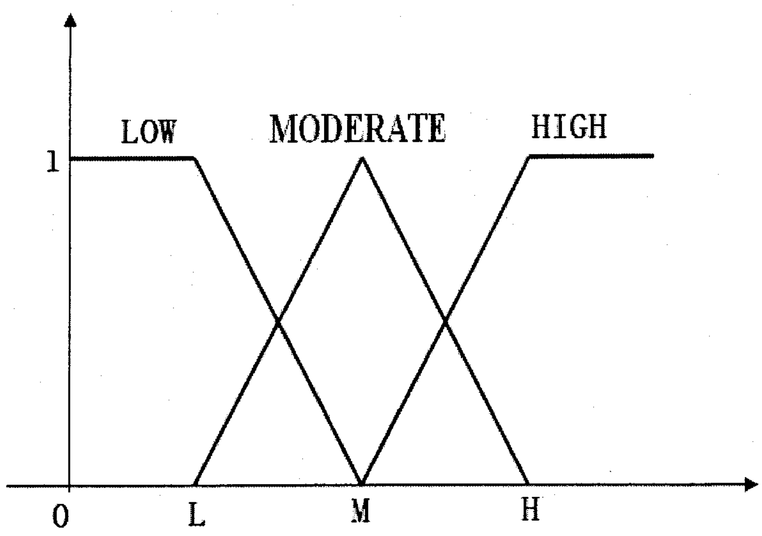


Figure 23: The membership functions of the output- acceptance

5.1.2 Inference rules

The complete set of inference rules that map the inputs to the output are setting as following.

Rule 1: If *offer price* (**OF**) is *low* (**L**) and quality level (**QL**) is *low* (**L**), then the *buyer acceptance* (**BA**) is *moderate* (**MO**).

Rule 2: If **OF** is **L** and **QL** is **M**, then **BA** is **H**

Rule 3: If **OF** is **L** and **QL** is **H**, then **BA** is **H**

Rule 4: If **OF** is **M** and **QL** is **L**, then **BA** is **L**

Rule 5: If **OF** is **M** and **QL** is **M**, then **BA** is **MO**

Rule 6: If **OF** is **M** and **QL** is **H**, then **BA** is **H**

Rule 7: If **OF** is **L** and **QL** is **L**, then **BA** is **MO**

Rule 8: If **OF** is **L** and **QL** is **M**, then **BA** is **L**

Rule 9: If **OF** is **L** and **QL** is **H**, then **BA** is **L**

These rules are defined based on common sense. It is quite natural for a buyer who wants to buy a product with high quality and low price. These rules are easy to read and understand.

5.2 Performance test

The previous offer evaluation block for a buyer agent is implemented by c# based on Microsoft .NET techniques.

5.2.1 Microsoft .NET implementation

Introduction of .NET framework and c#

When Microsoft announced C# in July 2000, the unveiling was a part of the much larger event: the announcement of Microsoft .NET platform. .NET platform provides a fresh application programming interface (API) for application development in Windows operating system, while bringing together a number of legacy technologies that emerged from Microsoft, such as COM/ActiveX component services, ASP web development framework, XML and object-oriented design.

Microsoft claimed it has devoted eighty percent of its research and development resources to .NET and associated technologies. The results of this commitment are really impressive, this is main reason we choose .NET platform. The .NET platform consists of four separate product group: [32]

- A set of languages, including C#, VB.NET; a set of development tools, including Visual Studio .NET; a comprehensive class library for building web and Windows applications, as well as web services; the Common Language Runtime (CLR) to execute objects built within the framework.
- A set of .NET Servers, including SQL server 2000, Exchange 2000, BizTalk 2000, provide functionalities of data storage.
- A repository of commercial web services offer developers for a fee to build applications.
- .NET- enabled non-PC smart devices, from cell phone to PDA.

.NET framework contains many features that are common to the Java Virtual Machine (JVM), the code written in .NET languages, no matter which one it is, is first compiled into a intermediate language called Microsoft Intermediate Language (MSIL); since the MSIL is targeted to the CLR, it means that it is processor and platform independent, just like the byte-code in Java; then the MSIL code is compiled to machine code using Just-In-Time (JIT) compiler and executed by CLR.

The Figure 24 illustrates how C# code is executed on a Windows platform.

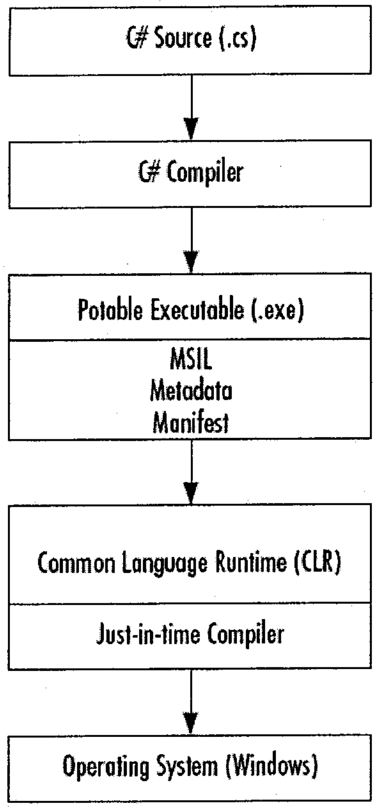


Figure 24: The process of c# program running on windows system [33]

Testing interface

Figure 25 is the screenshots of testing interface implemented in windows application.

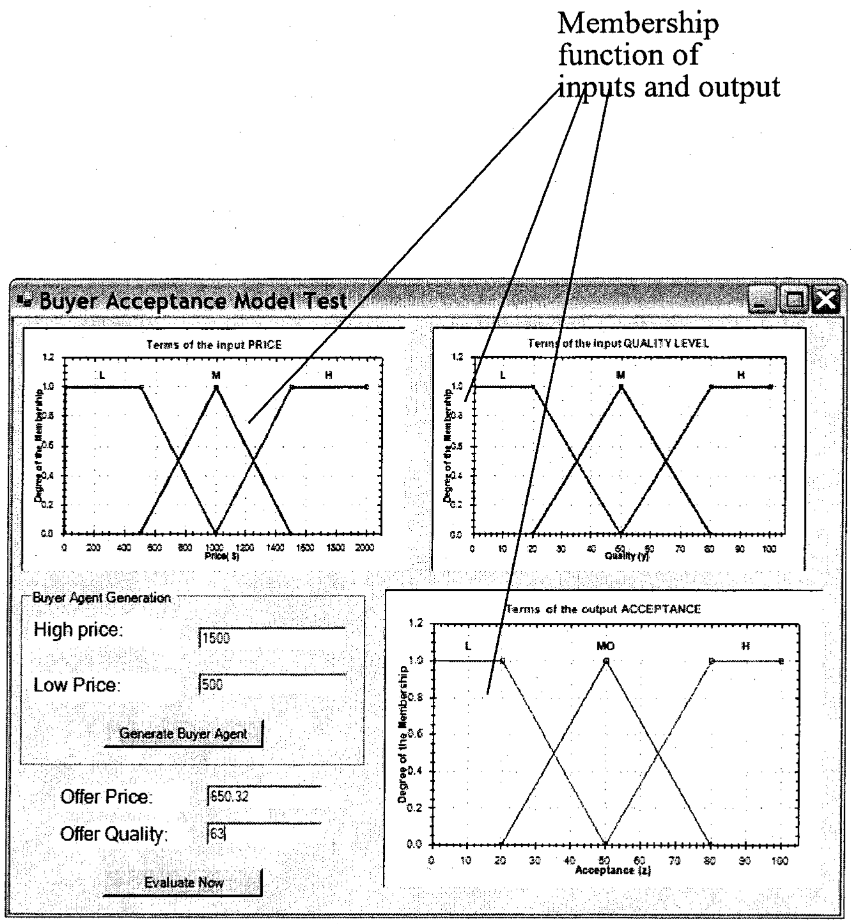


Figure 25: The screenshot of the test interface of offer evaluation model

The three graphs are showing the membership functions of the two inputs (the PRICE and the QUALITY LEVEL) and the output (the ACCPETANCE). The user first sets the price he/she thinks is absolutely high and the price he/she thinks is absolutely low in the 'buyer agent generation' region. Then, he/she clicks the 'Generate Buyer Agent' button to generate a buyer agent which represents the user. We can see the current high price is $1,500 and low price is $500. Later, the user fills in the current offer price and the offer quality level into the text box at the bottom. We can see the current offer price is $650.32

and offer quality level is 63. Finally, the user clicks the 'Evaluate Now' button. The output is shown in Figure 26.

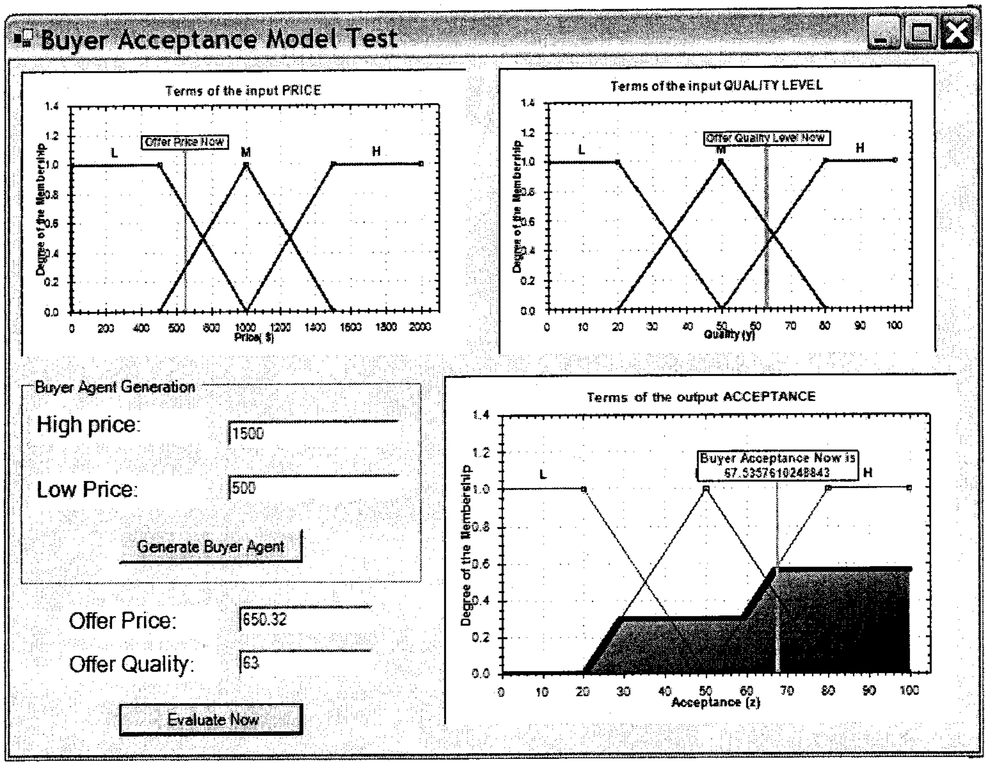


Figure 26: The screenshot of the test interface of offer evaluation model (2)

The final crisp acceptance can be read from the graph. It is 67.536 for the above case. We tested different combinations of the inputs using this test application.

The results are shown in Tables 5-6. The buyer highest acceptable price and lowest desired price are $1500 and $500, respectively.

The offered price	The quality level	The buyer acceptance
$0.00	50	81.5
$200.00	50	81
$400.00	50	81
$600.00	50	73
$800.00	50	60.5
$1,000.00	50	50
$1,200.00	50	39
$1,400.00	50	26
$1,600.00	50	18

Table 5: The buyer acceptance vs. fixed quality level and various offer prices

The offer price	The quality level	The buyer acceptance
$1,000.00	0	18
$1,000.00	10	18.5
$1,000.00	30	31
$1,000.00	50	50
$1,000.00	70	68
$1,000.00	90	81
$1,000.00	100	81.5

Table 6: The buyer acceptance vs. fixed offered prices and various quality levels

The results show that the performance of the system is pretty good and reasonable. When the price is high and quality level is low, the acceptance is low and vice versa.

5.2.2 Performance Test using MATLAB fuzzy logic toolbox

A more comprehensive performance test of the offer evaluation block than .NET testing is done by using the MATLAB fuzzy logic toolbox.

Introduction of MATLAB fuzzy logic toolbox

The Fuzzy Logic Toolbox is a set of functions which are built on the MATLAB® numeric computing environment. It provides lots of tools to create and edit fuzzy inference systems (i.e. fuzzy expert system or fuzzy control system) within the framework of MATLAB. The customized fuzzy inference systems can be simulated with Simulink®. The stand-alone C programs which call on fuzzy system can be built with MATLAB also. A user-friendly graphical user interface (GUI) helps to design of fuzzy inference system easily. Figure 27 shows the relation between fuzzy logic toolbox and other components of MATLAB framework.

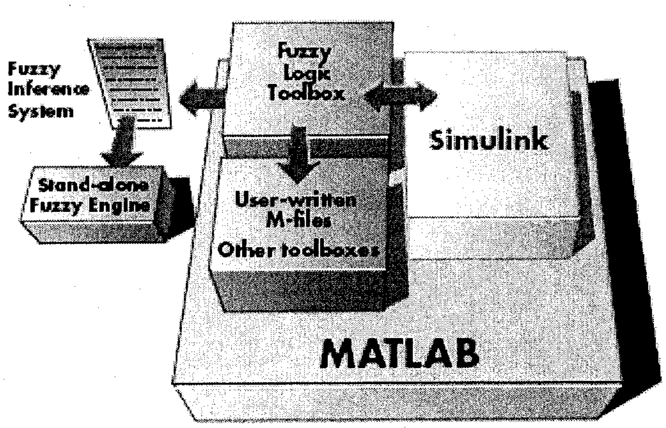


Figure 27: The relation between fuzzy logic toolbox and other components of MATLAB [34]

The toolbox provides three categories of tools:

- Command line functions
- Graphical, interactive tools
- Simulink blocks and examples

The first category of tools consists of functions which can be called in the command line. Many of these functions are written in MATLAB M-files, series of MATLAB statements that implement specialized fuzzy logic algorithms.

Secondly, the toolbox provides a number of interactive tools that let users access many of the functions through a GUI. Together, the GUI- based tools provide an environment for fuzzy inference system design, analysis, and implementation.

The third category of tools is a set of blocks for use with the Simulink simulation software. These are specifically designed for high speed fuzzy logic inference in the Simulink environment.

The performance test of offer evaluation block built in MATLAB

Using the MATLAB fuzzy logic toolbox to test the performance of the offer evaluation model is quite simple, compared with using c#. The MATLAB should be a good tool for designing the model of negotiation agents and testing the performance of the model. However, the actual software negotiation agent will be implemented by using Microsoft .NET and put into the e-marketplace.

The following figures are the results obtained from the MATLAB code.

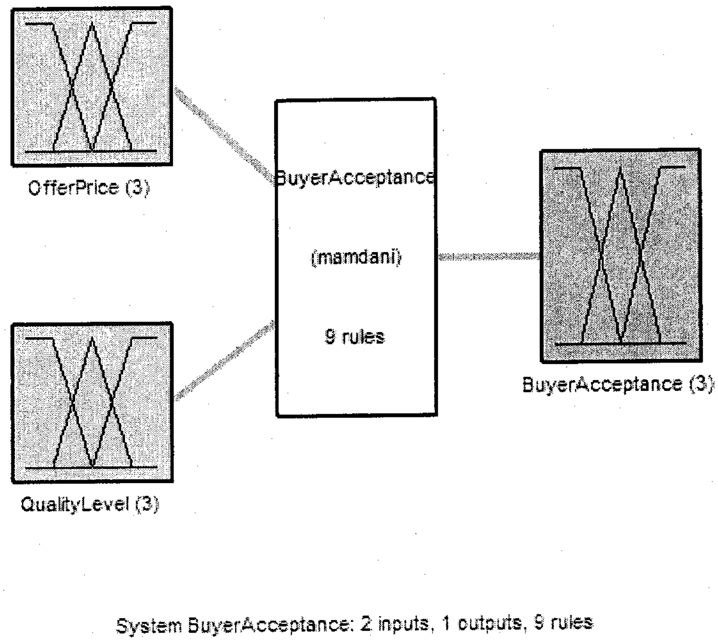


Figure 28: The overview of the offer evaluation block model drawn by MATLAB

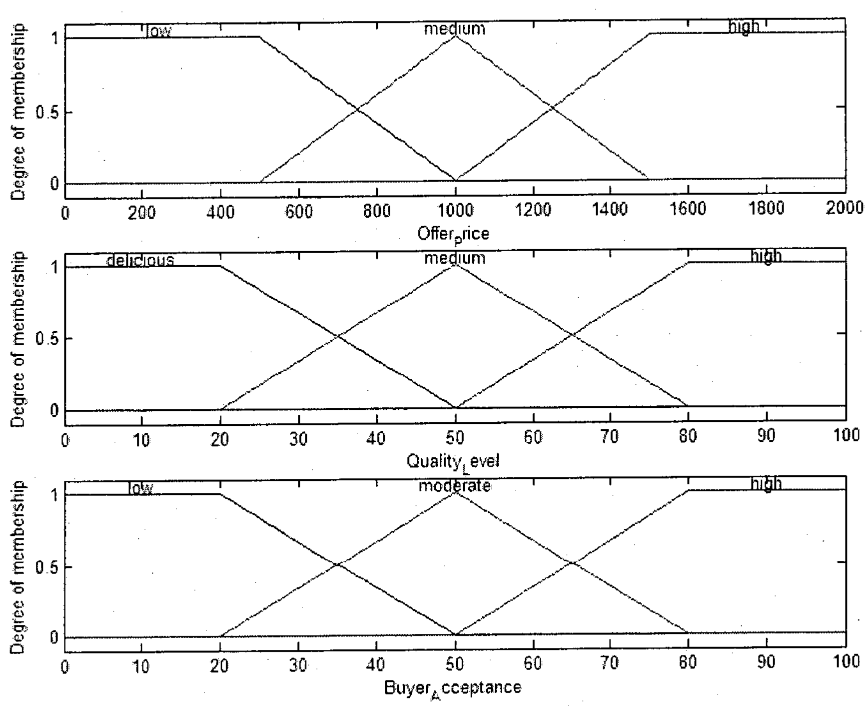


Figure 29: The functions of degree of membership for inputs and output

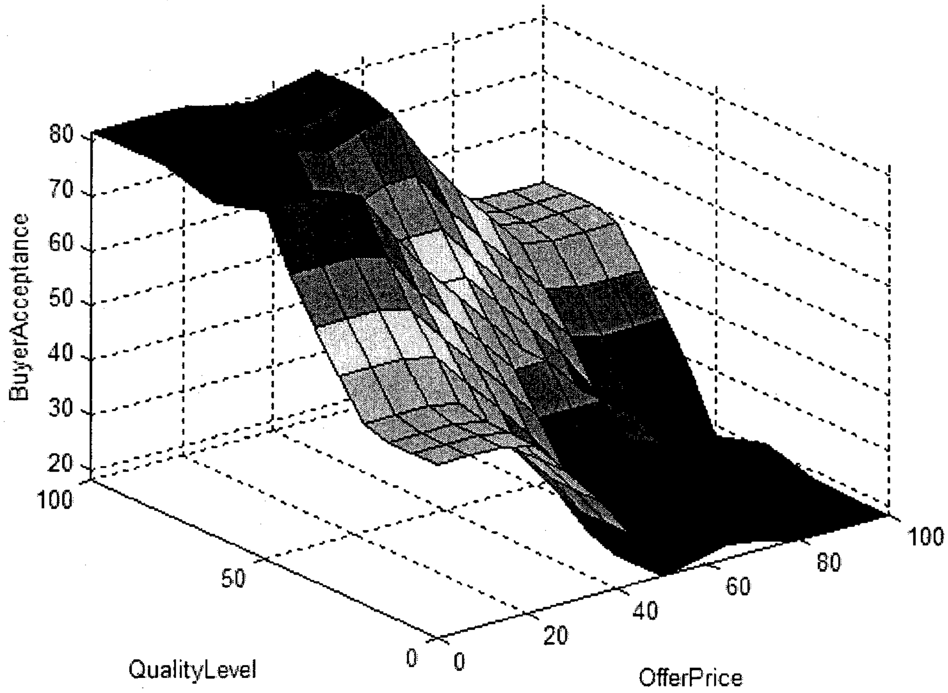


Figure 30: The 3D acceptance surface vs the offer price and quality level

We can see from the Figure 30 that when the offered price is low and the quality level is high the acceptance is high and vice versa. We can conclude that the performance of system is correctly expressing the actual view from the angle of a buyer.

To prove the flexibility of the model, we added a new input 'return policy' into it. We used the fuzzy interference system (FIS) editor. The fuzzy set of 'return policy' includes three vague components poor, average and good. The model is shown in Figure 31.

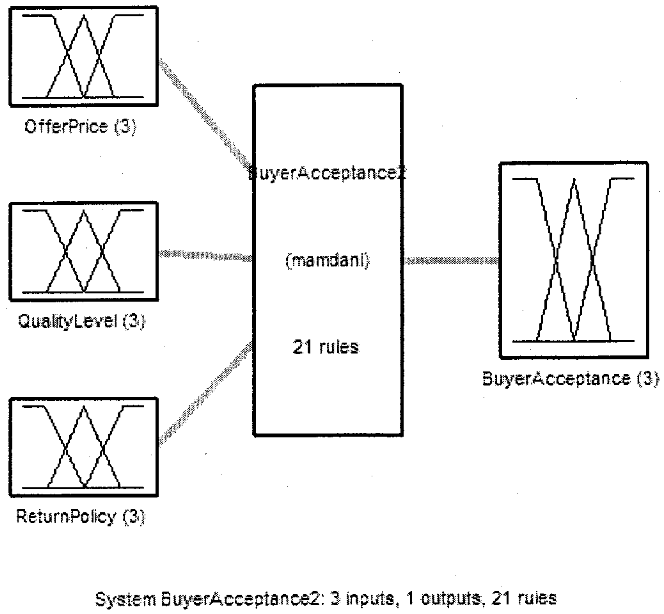


Figure 31: The overview of the extended offer evaluation block model drawn by MATLAB

Twelve extra rules are added onto the previous model, because the new issue "return policy" is taken into consideration. The new set of inference rules are listed in below:

Rule 1: If *offer price* (**OF**) is *low* (**L**) and quality level (**QL**) is *low* (**L**), then the *buyer acceptance* (**BA**) is *moderate* (**MO**).

Rule 2: If **OF** is **L** and **QL** is **M**, then **BA** is **H**

Rule 3: If **OF** is **L** and **QL** is **H**, then **BA** is **H**

Rule 4: If **OF** is **M** and **QL** is **L**, then **BA** is **L**

Rule 5: If **OF** is **M** and **QL** is **M**, then **BA** is **MO**

Rule 6: If **OF** is **M** and **QL** is **H**, then **BA** is **H**

Rule 7: If **OF** is **L** and **QL** is **L**, then **BA** is **MO**

Rule 8: If **OF** is **L** and **QL** is **M**, then **BA** is **L**

Rule 9: If **OF** is **L** and **QL** is **H**, then **BA** is **L**

Rule 10: If **OF** is **L**, **QL** is *not* **H** and *return policy* **(RP)** is *good* (**G**), then **BA** is **H**

Rule 11: If **OF** is **L**, **QL** is *not* **L** and **RP** is *not* **G**, then **BA** is **MO**

Rule 12: If **OF** is **H**, **QL** is **H** and **RP** is *not poor* (**PR**), then **BA** is **MO**

Rule 13: If **OF** is *not* **H**, **QL** is *not* **L, and RP** is *not* **PR**, then **BA** is **MO**

Rule 14: If **OF** is **H**, **QL** is **L** and **RP** is **PR**, then **BA** is **L**

Rule 15: If **OF** is **H**, **QL** is not **M** and **RP** is **PR**, then **BA** is **L**

Rule 16: If **OF** is **H**, **QL** is **H** and **RP** is **PR**, then **BA** is **L**

Rule 17: If **OF** is **H**, **QL** is **H** and **RP** is *average* (**AVG**), then **BA** is **MO**

Rule 18: If **OF** is **H**, **QL** is not **H** and **RP** is **G**, then **BA** is **MO**

Rule 19: If **OF** is **L**, **QL** is not **M** and **RP** is **G**, then **BA** is **H**

Rule 20: If **OF** is **L**, **QL** is not **H** and **RP** is **G**, then **BA** is **H**

Rule 21: If **OF** is **M**, **QL** is not **H** and **RP** is **G**, then **BA** is **H**

The first nine rules are inherited from the previous model. The new inference rules, as well as the old rules, are used for reasoning in the extended model; however, these rules are not complete for each possible case of inputs. Presently, we have three inputs and each input has three elements. We need at least $3 \times 3 \times 3 = 27$ rules to cover every combination of inputs. With the increase of the number of inputs, the size of the complete set of inference rules increases geometrically. Fortunately, few rules in fuzzy logic are needed to get reasonable results. Therefore, the fuzzy logic based model can be extended without increasing too much complexity for more complicated cases.

Figures 32-34 are the results we got from the MATLAB model.

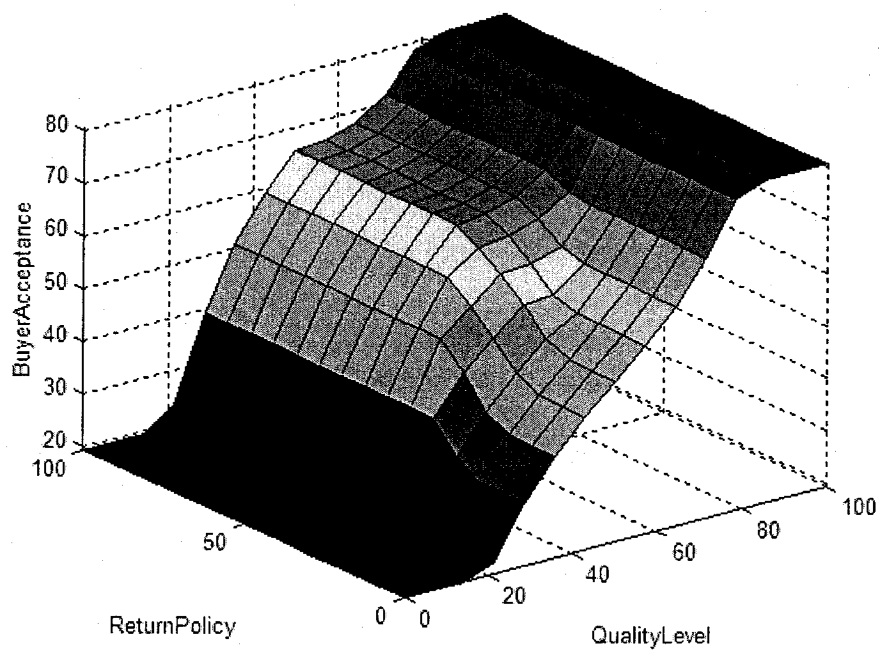


Figure 32: The 3D acceptance surface vs the return policy and quality level

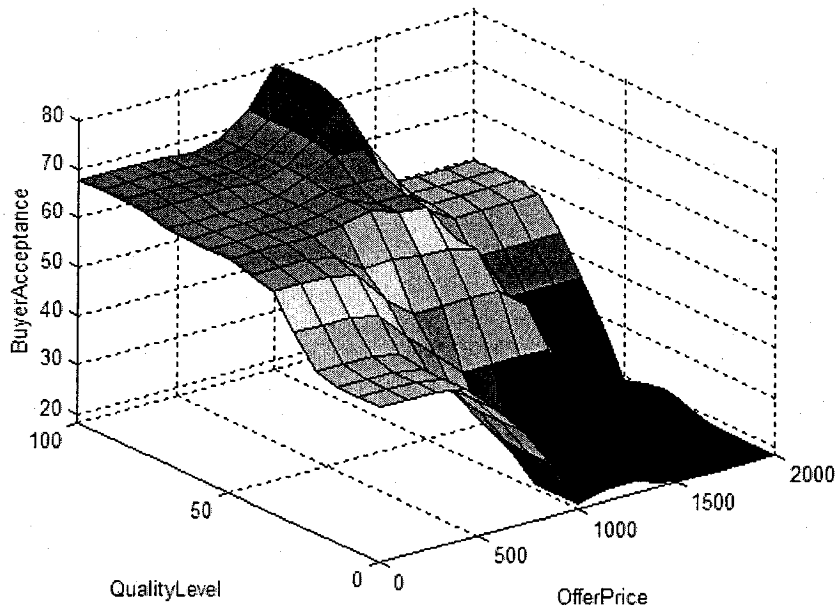


Figure 33: The 3D acceptance surface vs the offer price and quality level

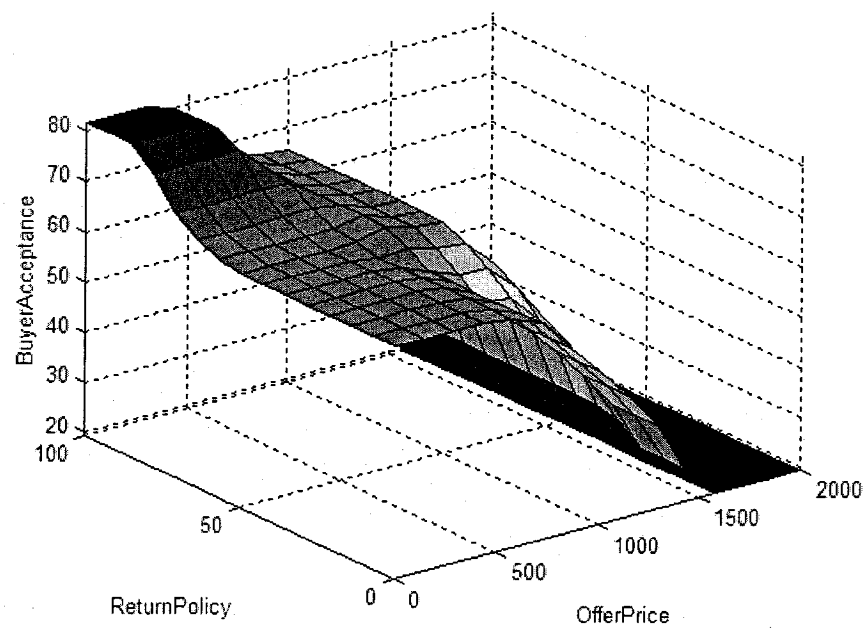


Figure 34: The 3D acceptance surface vs the offer price and return policy

The performance of the extended system is good. For example, when return policy is good, price is low and quality level is high, the acceptance is high. This demonstrates the ease to extend the fuzzy system to achieve more complicity. There is no requirement to rewrite the original rules. After adding an input, we only add few new rules to describe the relation between new input and original inputs and output and then achieve desired performance. For a conventional expert system, all possible inference rules are necessary to achieve good performance. Compared to the conventional expert system the fuzzy expert system is simpler and need less memory for storing inference rules.

5.3 Safety and security concerns

"*Growth of eCommerce has made the Internet an enticing playground for hackers and crackers*" [44]

In this thesis, the security, authentication, authorization, confidentiality, integrity, nonrepudiation, etc., requirements for eCommerce systems are not included in the proposed model.

Security is required to protect eCommerce system from hacking, port sniffing, viruses, etc..

A face-to-face, paper-based negotiation does not have several challenges that are inherently involved inside an automated agent based e-negotiation. Properties of ink, the organization letterhead, watermarks, signatures, timestamps and the ability to detect modification are absent in e-transactions. Thus several security concerns should be taken into consideration to protect our automated negotiation model.

Authentication to validate the buyer/seller involves, for example, a login system, digital signature, biometric characteristic, such as fingerprint, etc. Authentication technology may also involve a combination of different authentication methods.

Authorisation includes the control of access to particular information, once identity has been verified. Authorisation involves a set of policies that limit the actions or operations of authenticated users, for example, the different levels of permission that are given to users. Table 7 is a simple example of different levels of permission.

Level of leniency	Permissions
Level 3 (Most strict)	Reading, browsing general information Creating and modifying documents Setting up seller/buyer agents Executing transaction
Level 2	Reading, browsing general information Creating description documents
Level 1 (Most lenient)	Reading, browsing general information

Table 7: An example of levels of permission considered by Authorisation

Confidentiality considerations include, for example, the privacy of data, etc. The goal of confidentiality is to protect confidential information from unauthorised access. For automated commercial negotiation, the negotiation strategy, the preference for each negotiated issue, and the detailed information of the initial offer are confidential for each agent. Confidentiality policy should ensure such information couldn't be read, copied, modified, or disclosed without proper authorisation. Furthermore, no interception of information is allowed during a negotiation.

Integrity considerations include the protection of data from modification in transit or in storage. A practical eCommerce system must be able to ensure that data transmissions over a network arrive at their destinations in exactly the same form as they were sent. In our proposed negotiation model, the messages, which are exchanged during a negotiation process, should be protected from modifications, additions, deletions, and reordering parts of data.

Nonrepudiation involves protection against a party involved in a commercial activity that later denies that activity occurred. The above nonrepudiation may involve third party verifications.

Chapter 6

Conclusion

This thesis proposed a model for fuzzy logic based intelligent negotiation agent. The proposed negotiation agents are suitable for an agent based e-marketplace. An offer evaluation block has been implemented by using Microsoft .NET and fuzzy approaches. The advantages of our proposed fuzzy approach:

- The fuzzy logic based model is based on natural language and easy to interpret.
- The fuzzy logic based model can handle imprecise, vague or incomplete information. Real-world information is imprecise, vague, or incomplete, e.g. low price, budget about $50, high quality, etc. The users cannot define a clear boundary for making decisions.
- The fuzzy logic based model can be built on human experience. In direct contrast to neural networks, which take training data and generate opaque, impenetrable models, fuzzy logic lets you rely on the experience of people who have understanding of the system. The negotiation agents are experts on negotiating and built by users.
- The fuzzy logic based model is flexible. With any given system, it's easy to manage or put more functionality layer on top of it without starting again from scratch. Thus, the system can be easily extended to handle more sophisticated future requirements. The model can be extended to a more advanced model by adding inputs, outputs and new inference rules without modifying the original model.

By analyzing the results from the Windows .NET test application and the MATLAB solutions, we can conclude that our proposed offer evaluation blocks are acceptable, implementable, extendable and scalable.

Although the current work has achieved the goal of finding an appropriate methodology to model an automated negotiation agent in order to alleviate the complexity of negotiations, more need to be accomplished to provide a truly automated negotiation system. For example, our proposed model uses one symmetrical triangular fuzzy number, one right trapezoidal fuzzy number and one left trapezoidal fuzzy number to describe the three vague terms of inputs and outputs. It would be interesting to test the performance of a model under various shapes of membership functions. For example, a more profitable solution can be achieved by choosing a flatter triangular fuzzy controlling variable instead of current one. .

Our proposed model provides for the one-to-one negotiation in which two automated agents are involved; however, a many-to-many negotiation might be more efficient, practicable.

There are several researches on how to extend one-to-one negotiation to many-to-many negotiation. For example, in Kurbel and Loutchko[39], provides for breaking the global time into several intervals. Within certain interval an agent negotiates with one participant whose proposal is the most preferable to the agent among other participants.

Another obvious area of future work is to develop a solution for ensuring that the negotiation process takes place in a secure trustable environment. Security issues, such as authentication, authorization, confidentiality, integrity, nonrepudiation, etc., should be considered in the future model in order to make it more practical.

References

[1] M. He, N. R. Jennings and H. F. Leung, "On Agent-Mediated Electronic Commerce", *IEEE Transactions on Knowledge and Data Engineering*. Vol.15, No.4. pp 985-1003, July/August 2003.

[2] R. Fisher, W. Ury, and B. Patton ed., *Getting to Yes: Negotiating Agreement Without Giving In*, 2nd ed. Boston, Houghton Mifflin, 1991.

[3] S. Murugesan, "Intelligent Agents on the Internet and Web", *Proceedings of the IEEE Region IO Conference*, Tencon'98, New Delhi, 1998.

[4] G. P. Schneider, "Electronic Commerce, Fourth Annual Edition", pp. 5, Thomson Course Technology, July, 2003

[5] K. Nagi, "Transactional Agents towards a Robust Multi-Agent System", Doctoral thesis, University of Karlsruhe, Germany

[6] F. Dignum, *Agent mediated electronic commerce*, Springer, 2001

[7] M. Weiss, "A gentle introduction to agents and their applications", http://www.magma.ca/~mrw/agents/toc.html , 1999

[8] S. Bussmann and J. Muller, "A Negotiation Framework for Co-operating Agents", in Dean S M (Ed): 'Proc CKBS-SIG', Dake Centre, University of Keele, pages 1-17, 1992.

[9] M. M. Delaney, A. Foroughi and W. C. Perkins, "An Empirical Study of the Efficacy of a Computerized Negotiation Support System (NSS)," *Decision Support Systems*, vol. 20, no. 3, pages 185-197, 1997.

[10] T. X. Bui, and M. F. Shakun, "Negotiation Processes, Evolutionary Systems Design, and Negotiator," *Group Decision and Negotiation*, vol. 5, pages 339-353, 1996.

[11] G. E. Kersten and S. J. Noronha, "WWW-based Negotiation Support: Design, Implementation, and Use," *Decision Support Systems*, vol. 25, no. 2, pages 135-154, 1999

[12] S. Matwin, S. Szpakowicz and Z. Koperczak, "Negoplan: An Expert System Shell for Negotiation Support," *IEEE Expert*, vol. 4, no. 4, pages 50-62, 1996.

[13] G. E. Kersten and G. Lo, "Negotiation Support Systems and Software Agents in Business Negoitaions," The First International Conference on Electronic Business, Hong Kong, 2001.

[14] A. Chavez and P. Maes, "Kasbah: An agent marketplace for buying and selling goods", *Proc. of 1st Int'l Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology*, pp 75--90, 1996.

[15] MIT Media Lab. "Tete-a-Tete," 2000. Online: http://ecommerce.media.mit.edu

[16] G. E. Kersten and S. J. Noronha, "WWW-based Negotiation Support: Design, Implementation, and Use," *Decision Support Systems*, vol. 25, no. 2, pages 135-154, 1999

[17] A. Roger, "McCain: Game Theory: A Non-Technical Introduction to the Analysis of Strategy", available at http://william-king.www.drexel.edu/top/eco/game/nash.html.

[18] M. Shor, "2001-2005: Repeated Prisoner's Dilemma Applet", available at http://www.gametheory.net/Web/PDilemma/ .

[19] E. H. Gerding, D. D. B. van Bragt and J. A. La Poutre, "Multi-issue negotiation processes by evolutionary simulation". In Workshop on Complex Behaviour in Economics, Aix-en-Provence, France, May 2000.

[20] C. Beam and A. Segev, "Automated negotiations: A survey of the state of the art", *Wirtschafts- informatik* , pp. 263-267. 1997

[21] J. A. Rodriquez, P. Noriega, C. Sierra and J. Padget, "FM96.5-A Java-based electronic Auction House", *Proceedings of the Second International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM97)*, London, UK, pp.207-224, 1997.

[22] H. Raiffa, *The art and Science of Negotiation*, Cambridge: Harvard University Press, 1982.

[23] B. Russel, "Vagueness", *Australian Journal of Psychology and Philosophy*, 1, pp84-92, 1923.

[24] G. Bojadziev and M. Bojadziev, *Fuzzy Logic for Business*, Finance, and Management, World Scientific, 1997

[25] J. Lukasiewicz, *On 3-valued logic*, Ruch Filozoficzny, 5, pp169-171, 1920.

[26] PCAI - Glossary of Terms, http://www.pcai.com/web/glossary/pcai_d_f_glossary.html

[27] A. Kandel ed., *Fuzzy Expert Systems*, CRC Press, Nov. 1991

[28] H. Hellendoorn and C. Thomas, "Defuzzification fuzzy controllers", *Journal of Intelligence and Fuzzy Systems*, 1, pp109-123, John Wiley and Sons, Inc., 1993.

[29] S. Klaue, K. Kurbel and I. Loutchko, "Automated Negotiation on Agent-Based E-Marketplaces, An Overview", Proceedings of 14th Bled Electronic Commerce Conference, Bled, Slovenia, June 25-26, 2001, 508-519.

[30] P. Faratin, C. Sierra and N. R. Jennings, " Negotiation decision functions for autonomous agents", *Robotics and Autonomous Systems* 24 (1998), 159-182.

[31] R. Axelrod, *The evolution of cooperation*, Basic Books, Inc. Publishers, New York, USA., 1984

[32] J. Liberty, *Programming C#*, 2nd edition, O'Reilly, 2002

[33] B. Bagnall, P. Chen, S. Goldberg, J. Faircloth and H. Cabrera, *C# for Java Programmers* , Syngress, 2002

[34] Fuzzy Logic Toolbox 2.2.2 http://www.mathworks.com/products/fuzzylogic/

[35] S. S. Fatima, M. J. Wooldridge and N.R. Jennings, "Multi-issue negotiation under time constraints", In: Proc. 1st Int. Joint Conf. On Autonomous Agents and Multi-Agent Systems, Bologna, Italy (to appear), 2002.

[36] S. Bussmann, N. R. Jennings and M. J. Wooldridge, "On the identification of agents in the design of production control systems", *Agent-Oriented Software Engineering* (eds. P. Ciancarini and M. Wooldridge), Springer Verlag, 141-162. 2001

[37] R. Kowalczyk and V. Bui, "On fuzzy e-negotiation agents: Autonomous negotiation with incomplete and imprecise information", *Proc. DEXA e-Negotiation Workshop*, 2000

[38] R. Kowalczyk and V. Bui, "FeNAs: A Fuzzy e-Negotiation Agents System", the Conference on Computational Intelligence for Financial Engineering (CIFEr 2000), NY

[39] K. Kurbel and I. Loutchko, "Multi-agent Negotiation under Time Constraints on an Agent-based Marketplace for Personnel Acquisition", *Proceedings of the 3rd International Symposium on Multi-Agent Systems, Large Complex Systems, and E-Business (MALCEB2002)*, Erfurt, 8-10 October 2002, S. 566-579.

[40] V. Loia, M. Nikravesh and L. A. Zadeh (Eds.), *Fuzzy Logic and the Internet*, Springrt-Verlag Berlin Heidelberg, Germany, 2004

[41] R. G. Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver." *IEEE Transactions on Computers* C-29, 12: pp1104-1113, December 1980

[42] N. J. Daviesand R. Weeks, "Jasper: Communicating Information Agents", *Proceedings of the 4th International Conference on the World Wide Web*, Boston, USA, December 1995

[43] N. Shi and V.K. Murthy ed, *Architectural Issues of Web-Based Electronic Business*, Idea Group Inc., 2003

[44] E. Turban,J. Lee, D. King,and H. M. Chung, *Electronic Commerce A Managerial Perspective*, Prentice Hall International Inc., New Jersey, 2002