Université de Montréal

Nego: A Virtual Negotiation Market

Par

Dongfang Li

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en informatique

Avril, 2004

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé

Nego: A Virtual Negotiation Market

Présenté par :

Dongfang Li

A été évalué par un jury composé des personnes suivantes :

Président-rapporteur :	Esma Aïmeur
Directeur de recherche :	Claude Frasson
Membre du jury :	Yann-Gaël Guéhéneuc

Mémoire accepté le: 12 octobre 2004

Résumé

Les négociations font partie de toute transaction commerciale. De nos jours, nous assistons à une prolifération importante du commerce électronique d'où le besoin grandissant de passer des négociations traditionnelles aux négociations électroniques. Ce passage est motivé par l'adhésion d'un nombre croissant d'entités commerciales à des réseaux informatiques qui fournissent un volume d'information important avec des moyens de communication rapides, sophistiqués et de plus en plus sécurisés. Ce progrès ne peut être atteint sans résoudre de nouveaux défis d'ordre technique et organisationnel. En réponse à ces défis, nous proposons une plateforme de négociations, appelée 'Nego', permettant aux adeptes du commerce électronique de négocier des prix de produits d'une manière efficace et rentable.

Le système Nego est un marché virtuel dans lequel on a adopte la technologie des agents mobiles. Les agents mobiles négocient aux noms des acteurs de la transaction commerciale. Les principaux avantages du système Nego consistent à, premièrement, libérer l'acheteur des détails de la négociation réduisant ainsi le trafic des informations de négociation dans le réseau; deuxièmement, rendre l'acheteur dépendant de ses expériences d'achat précédentes ainsi que de ses préférences par l'application de la théorie de l'utilité subjective; ceci conduit à la production de la solution la plus pratique et la plus personnalisée qui puisse exister. Finalement, Nego accélère la négociation en traitant tous les vendeurs d'une manière équitable en terme de temps. En somme, notre système peut être considéré comme un outil de traitement de la quatrième étape du comportement d'achats selon le model CBB (Consumer Buying Behaviour), à savoir, la négociation où les agents intelligents jouent un rôle primordial.

Mots clés : agents intelligents, agents mobiles, négociation, offre, prix d'équilibre.

Abstract

Negotiation plays a very important role in commercial transaction. Today, e-commerce develops rapidly, and conventional negotiation is transforming to e-commerce negotiation. The change of conventional negotiation is significant because of the internet provides wide-ranging information, convenient and rapid communication, and a great quantity of commercial entities. In this thesis, we describe a negotiation system, called 'Nego', to provide the e-commerce consumer with an efficient way to negotiate for the price of products in e-commerce market.

Nego is a marketplace, in which mobile agent technology is adopted in the e-commerce negotiation on behalf of the buyer. It mainly solves three problems: First, it frees the buyer from negotiation detail and reduces the amount of negotiation information in the network; Second, it leads that the buyer must base on his previous experience and preference to produce the solution more practical and personalized, it follows the theory of subjective expected utility (SEU); Third, it speeds up the negotiation by handling all sellers in a fair time manner. The proposed system can be considered as a tool to deal with the fourth stage of the Consumer Buying Behaviour model (CBB): negotiation, in which intelligent agent plays important role. Our purpose is to meet the challenges coming from more and more popular e-commerce activities.

Key words: Intelligent Agent, Mobile Agent, Negotiation, Bid, Offer, Equilibrium Price.

Table of Contents

List of Figures

List of Tables

List of Equations

Acknowledgement

Doing a thesis is a challenge work, in which there are a lot of enthusiasm and hard work. I would like to thank many people for their assistance and encouragement during I work for my thesis. First of all, I want to thank my supervisor, Professor Claude Frasson for giving me this opportunity and giving me advice in every period of my research work. He always arranges time discussing with me and giving me suggestion even though he is very busy. I have learned a lot and improved a lot in this period. I would not have finished my thesis without his advice.

I sincerely thank the other members of dissertation committee, Professor Esma Aïmeur and Professor Yann-Gaël Guéhéneuc. I would like to thank them for their reviewing of this thesis and provide valuable response and comments.

I want to thank Professor Jian-Yun Nie. He discusses with me and gives me his suggestion.

I want to thank my colleagues, Mohammed, Arnoldo, Maryam, and others in the laboratory HERON for their help, encouraging, discussing, and suggestion.

I also want to thank my friends, Amine and the others, for their help and encourage.

Finally and very important, I express my gratitude to my parents. They are always standing by me. Without their encouragement, I will have more difficulty to finish my thesis.

Chapter 1 Introduction

1.1 Overview

E-commerce is a general term of any business, or commercial transaction that involves the transfer of information across the Internet [URL_01]. As Internet provides an available access to global information in a real time manner, e-commerce has the advantage that the merchant is able to broadcast the commercial advertise in Internet easily, and the consumer is able to acquire the commercial information in Internet easily as well. When more and more traditional commercial operations are transformed into electronic commercial operations in Internet, e-commerce expanded rapidly recently and it is predicted to continue to expend in the future.

An e-commerce consumer usually searches in Internet and concerns how to find the best product and merchant that is the closest to his criteria, and how to fix the price which is acceptable to him. These concerned activities are defined as "product brokering", "merchant brokering" and "negotiation" by Consumer Buying Behaviour [Guttman *et al.*, 1998].

Intelligent agent has been widely used in e-commerce to help the buyers in their buying behaviours. The intelligent agent is a software entity, which works on behave of its user, and it is autonomous, proactive, reactive, communicative, cooperative, learning and mobile. For the "product brokering" and the "merchant brokering", there have been some efficient methodologies adopted. The technologies of information retrieval and information filtering succeed in collecting information and filtering for the useful information. And Artificial Intelligent agent techniques have helped to improve the product and merchant brokering result efficiently. In practical, there are some useful agent applications that are functional in the brokering stages to guide the user [Guttman *et al.*, 1998] [Pivk and Gams, 2000] [Turban *et al.*, 2002] [Ma, 2001]. When the buyer finishes searching, negotiation is a necessary step for the transaction.

Negotiation definition Negotiation is the process by which a group of agents communicate with one another to try to reach agreement on some matter of common items [Lomuscio *et al.*, 2003].

The purpose of negotiation is to reach an agreement between the buyer and the seller about the transaction terms, in which the buyer or seller are called "negotiator". Although there have been some negotiation applications, the negotiation in e-commerce is in its primitive state [Lomuscio *et al.*, 2003]. This is because that current e-commerce negotiation implements simple negotiation functions, one of them is the problems in price negotiation. In the following sections, we discuss it.

1.2 Problem Statement

The basic problem in "negotiation" is decision making. Price negotiation involves how to decide the equilibrium price. An equilibrium price is the price that the seller and the buyer match [URL_02]. To decide it, the buyer needs to know how to evaluate the real value of the product, and how to evaluate the seller's minimum offer(e.g., each seller has his minimum accepting price) of this product.

In the B2C (business to consumer, B2C implies the commercial transactions, in which companies sell products or service such as online banking, online consultant and so on to consumers for own use) e-commerce, when the buyer finds some products, he usually wants to negotiate for the transaction price to reach an agreement with the seller. During this procedure, the buyer decides his proposal of price (e.g., both proposal-deal and counter proposal-deal can be called as proposal-deal) according to his situation and the situation of the seller to match the seller's proposal-deal. In the view point of a buyer, he can only accept the price less than his maximum price. In view of a seller, he can only accept the price more than his minimum price. For both buyer and seller, how to find the equilibrium price consistent with their specific expectation is the major challenge. Take the individual buyer as an example. The buyer normally wants to find out a most suitable price, which is the lowest for him to accept, and high enough for the seller to accept. If the buyer's proposal-deal is not the possible lowest price, he will lose money; if the buyer's proposal-deal is not a high enough price, the seller might turn to other possible buyers. Therefore, the main problem of the buyer is to decide exactly the suitable proposal-deal. Failing in proposing a suitable proposal-deal, the buyer may lose his business chance.

Compared with the "product brokering" and "merchant brokering", the "negotiation" in e-commerce is not so popular. The current e-commerce negotiation just implements the simple price negotiation, in which a buyer just gives his minimum price, increment rate, and maximum price. During the negotiation, the buyer just increases an instant increment rate in an instant increment time toward his maximum price. For example, the world's biggest online marketplace eBay's "Reserve Price Auction" [URL_03] and Yahoo's "Automatic Bidding" [URL_04] provide such negotiation function for the buyers and sellers. But the negotiation functions in both sites implement simple negotiation skill, in which a buyer just specifies the starting and maximum price that he can provide and an instant increment rate. The negotiation continues in the buyer's increasing the bid by the instant increment rate toward the maximum price. There is a problem in this way that the user always wants to learn during the negotiation, in which he can adjust the increment with a dynamic rate according to the situation on that time instead of an instant increment rate.

To solve this kind of problem of not being able to adjust the increment dynamically mentioned in the above, we must consider many problems that affect the decision making in the e-commerce negotiation. These problems include: the buyer's knowledge of the market and the seller is limited, the buyer and the seller have different personal preference, and the values between the buyer and seller are inconsistent. Therefore, e-commerce negotiation must solve the problem of uncertainty and incomplete information, which belongs to the decision area.

Decision Science is about how to evaluate and choose actions among several alternatives, and the theory of Subjective Expected Utility (SEU) defines the conditions of perfect utility-maximization rationality in a world of certainty or in a world in which the probability of all relevant variables can be provided by the decision makers, and it is the central idea of decision theory that lies in the foundation of contemporary economics [Simon *et al.*, 1986]. In brief, the SEU assigns each alternative a utility (a numeric value) and each outcome a probability, and maximize its expected utility. It is based on the personal judgement of the utility and personal estimate of likelihood [URL_27]. It can specify the uncertainty with its utility according to the subject of the person and focus on certain attributes by ignoring unimportant attributes [URL_28]. In general, SEU has been used in decision methodologies such as: guided by heuristic to search for solution, uses means-ends analyze (e.g., a kind of reasoning that to find the difference between the current state and the goal, then to choose the best operation to reduce the difference [URL_05]). One of its examples is the expert system in making medical diagnoses [Simon *et al.*, 1986]. Traditional commerce and e-commerce is different that e-commerce is based on Internet; adopting SEU in e-commerce negotiation will solve the problem of the limitation of computation in Internet.

To implement decision making, we need the user profile. The user profile keeps the buyer's personal information, his interest, and his previous experience. Normally, different buyers will have different way to make decision. The buyer's profile is helpful to make personal decision differently from others. Especially, the user's previous experience can tell the history of the use's value. A person has his behaviour-rule and he normally behaves in a consistent way. When a buyer finds something interesting and wants to buy it, he has the following problems: he does not know the general acceptable price of it (although he can check it in other resources, he would better have his own value), and the seller's reserved price.

Normally speaking, a buyer's preference, such as buyer A prefers seller X, while user B prefers seller Y, plays a very important role when a buyer make decision in negotiation. A buyer's preference can change the decision that is based on his experience. Different buyers have different preference, thus in the same situation the different buyer make different decision. To make a personal decision, the decision maker should consider the personal preference. Most of the current negotiation applications ignore the user preference. Examples include eBay's "Reserve Price Auction" [URL_03] and Yahoo's "Automatic Bidding" [URL_04].

In the view point of technology, e-commerce negotiation involves information passed between the client and server forward and backward frequently, thus make the access overwork to the e-commerce. Moreover, the process of negotiation involves a complicate computation, and the negotiation requires a rapidly response ability. The buyer negotiation agent normally can not afford to so great amount of computation with quick feedback.

Making a decision involves a complicate computation. Furthermore, negotiation is sensitive with time limit. If a buyer needs to negotiate with more than one seller to find the best seller, he must find an efficient way to negotiate with all of the sellers in the same time, otherwise some sellers must wait. It is a trouble for the buyer to let sellers wait because a seller normally contacts more than one buyer and the seller can not only wait for a specific buyer because of other opportunities existing.

In this thesis, we present "Nego", a price negotiation system based on intelligent agents. The basic idea is to provide the e-commerce buyer a useful way to bid, e.g., to negotiate for the product's price with a set of sellers according to the buyer's budget and preference, using mobile intelligent agent and sequence learning base on previous experience. To illustrate our proposal, Nego's output will show the negotiation result, e.g., a list of sellers' offers and buyer's bids in a buyer's negotiation. Table 1.1 shows the specific output sample of Nego. From it, we see how the buyer agent negotiates with all the sellers and how to choose the best one. In the example in Table 1.1, the buyer agent can not reach a deal with company A nor company C, but reach a deal with company B; therefore, the buyer agent will choose company B to buy.

Table 1.1: Output Sample of Nego

Merchant Name	Seller Offer	Buyer Bid
Company A	680	600
……	672.3	665.679
Final of A	672.3	665.679
Company B	700	600
……	677.462	665.679
……	668.948	668.948
Final of B	668.948	668.948
Company C	710	600
……	697.081	665.679
Final of C	697.081	665.679

1.3 Scope of Thesis

This thesis discusses how to use the intelligent agent technology to implement the price negotiation function in a virtual market with SEU theory, making use of the experience and user's preference. It focuses on the following problems: intelligent agent research, e-commerce negotiation tactics, Nego's Theory and Methodology, and mobile agent in Nego.

- **Intelligent Agent research**

Intelligent agents are a new paradigm for developing software applications [Jennings and Wooldridge]. They have certain properties. Here we answer the questions as: what properties the agent must have? What each property means? What the agent can do with certain properties such as autonomous, learning, communicative, and mobile?

There are several agent classifications. The first of them classifies the agents by their intelligence [Russell and Norvig, 1995], the second classifies the agents by their properties [Nwana, 1996], and the last classifies the agents by their application [URL_26]. Each kind of classification is meaningful to different situation.

Intelligent agents are widely used in e-commerce. Especially, there are many agent providing efficient functions in the product brokering and merchant brokering. There are several agents functional in negotiation, but there is still large space for the negotiation agents to improve. Through the discussion of different agent classification, a summary is given that some kinds of agent technique can be adopted to improve the e-commerce price negotiation.

- **E-commerce Negotiation Tactics**

There are several kinds of traditional negotiations in commerce, such as English Auction, Dutch Auction, First Price Sealed-Bid Auction, and Second Sealed-Bid Auction, which are the most popular types of auction [McAfee and McMillan, 1987]. Since e-commerce is sorted into B2B and B2C models, accordingly, we discuss how the conventional auctions can be adapted in these two models.

Several important applications of e-commerce negotiation also are discussed to find the advantages and problems. By discussing how to give a suitable bid in e-commerce negotiation, we propose our solution.

- **Nego's Theory and Methodology**

The main problem of Nego is to adopt an efficient methodology for the user to give proposal-deal when his agent negotiates for the product's price. We propose a methodology for Nego to solve the problem of giving a proposal-deal.

The useful way to value a buyer proposal-deal is to use the history-based algorithm, and the previous experience is a useful and available way to help him to make decision. The previous experience can guide him to have an exact value, and helps him to avoid mistake.

A buyer's preference normally includes two conflict aspects: favourite aspect and harmful aspect. The favourite aspect relates to the "gain" value of the buyer, and the harmful aspect relates to the "loss" value of the buyer. The "gain" and "loss" are the buyer's personal values to specify his favourite or harmful rate to a specific price.

We use SEU theory in Nego's decision making methodology. Based on the previous experience and user preference, a specific way produces the user expectation values, therefore deduces the proposal-deal.

- **Mobile Agent in Nego**

When the e-commerce continues expending, the accessing rate of the Internet increases rapidly. To reduce the access workload of Internet, it is necessary to reduce the repeated access. The negotiation between the buyer and the sellers will have a lot of data passed forward and backward, and make the situation worse. To solve this problem, we will take advantage of Nego.

There exists another problem that Nego's methodology involves a complicate computation. Working independently in the virtual market, the mobile agent has a side effect that it can not handle large amount of computation. This limitation will lead to slow down the negotiation. Nego solves this problem by an efficient way.

The buyer negotiation agent works on behalf of the buyer, and to avoid the problem of letting a part of potential sellers wait because the buyer negotiation agent is busy in negotiating with other potential sellers. A way to let all the negotiations between the buyer negotiation agent and the potential sellers process in a parallel time is implemented in Nego.

1.4 Thesis Organization

This thesis is divided into six chapters as the following indicated:

Chapter 2 is a literature review of the intelligent agent. It focuses on the agent classification and the agent's functions in the e-commerce application. It introduces the Consumer Buying Behaviour model as an underlying model to explain some important e-commerce agent applications and their supporting techniques to see the autonomous level of the e-commerce agent.

Chapter 3 introduces the negotiation theory. It outlines the properties of traditional negotiation and negotiation in e-commerce. It focuses on B2B and B2C negotiations and their strategies. It introduces the e-commerce negotiation development and introduces several important e-commerce negotiation applications, and discusses the current problems and the future development.

Chapter 4 explains the requirements of Nego's negotiation model. The Nego's negotiation model is presented to explain negotiation components, negotiation process flow and negotiation strategy. It focuses on the negotiation strategy which has detailed methodologies and algorithm-base on the SEU theory. Both the way to decide the utilities and experience values and the way to approximate the user's utility are explained. It implements the least-squares parabola, a kind of approximation of the given set of data ($utility_1$, $price_1$), ($utility_2$, $price_2$), ..., ($utility_n$, $price_n$) [URL_24], to find the price with the maximum utility. Finally, it compares Nego's negotiation strategy with the other current negotiation strategies and analyses its advantages and disadvantages.

Chapter 5 describes the Nego's distributing server-client architecture in which agents take an important role. It describes responsibilities of each agent during the procedure of negotiation. In the client side, it mainly describes the CNA (client negotiation agent). In the server side, it mainly describes the seller agent, and negotiation monitor agent. We explain the way the mobile agent is sent to the other side, the way the result is received, and the way the mobile agent negotiates with several sellers in parallel. The results of the Nego are presented as the support of our negotiation methodology's evaluation. We also explain the user interface, the user negotiation parameters.

Chapter 6 makes conclusion of Nego as an e-commerce negotiation agent supporting system. It evaluates the contribution of our work and discusses the disadvantage and future work.

Chapter 2 Agent Application in E-commerce

2.1 Introduction

Electronic commerce(also called "e-commerce") is all the commerce activities on network, which involves information processing. The basic objective of e-commerce is to improve the efficiency of business process and customer service. Absolutely, e-commerce can streamline the procurement process, cut the cost by efficient process, enable more and distant client, and let small enterprise or individual to participate.

One of technologies widely applied in e-commerce is agent technology. The successful and popular e-commerce agents include many search agents such as Copernic Agent [URL_08] and WebRankingAgent [URL_06]. But there are some agents such as decision making agent did not have popular application in e-commerce. Is this because the e-commerce does not involve decision making? To understand if decision making agents can be useful to the e-commerce, let us take a brief view of agent theory and the agent applications in the current e-commerce in the following sections.

2.2 What is Intelligent Agent?

An intelligent agent is a software entity that acts on behalf of itself to achieve the particular object. An intelligent agent might possess one or more of the following properties [Pivk et Gams, 2000] [Franklin and Graesser, 1996] [He *et al.*, 2003]:

- ***Autonomous***: can decide what action to take to achieve its goal by itself instead of referring to the user.
- ***Reactive (sensing and acting)***: interacts with its environment, responds in a timely fashion to changes in the environment.
- ***Proactive (Goal-oriented)***: does not simply act in responds to the environment, it acts to achieve its goal.

- ***Communicative***: communicates with other agents and people.
- ***Learning (Adaptive)***: can learn and change its behavior base on its previous experience.
- ***Mobile***: can travel from machine to machine.
- ***Collaborative***: can work in collaboration with other agent to achieve a common goal.
- ***Knowledge-level communicative***: the ability to communicate with human and other agents with language more resembling human-like speech than symbol-level protocols.
- ***Inferential***: can act on abstract task specification using prior knowledge of general goals and preferred methods to achieve flexibility.

In general, an agent has essential properties: autonomous, proactive and reactive. As for the other properties: communicative, learning, mobile, collaborative, knowledge-level communicative, inferential, and temporal continued, they are optional properties. In the following, let's see the agent definitions in the agent literature.

2.2.1 The AIMA Agent

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through effectors [Russell and Norvig, 1995].

AIMA points out that, a system is autonomous to the extent that its behavior is determined by its own experience. AIMA emphases that an agent should be rational doing right thing. "Rational" means "possible in reality". To measure rational at any given time there are four necessary parts:

- The performance measure that defines degree of success,
- Percept sequence that everything the agent has perceived so far,
- What the agent knows about the environment,
- A set of actions that the agent can perform.

2.2.2 The Maes Autonomous Agent

Autonomous agents are computational system that inhabits some complex dynamic environment, sense and act autonomously in this environment, and do so to realize a set of goal or tasks for which they are designed [Maes, 1995].

The autonomous agents are built in the approaches focus on fast, reactive behavior, rather than knowledge and reasoning, as well as adaptation and learning. Its approach is largely inspired by Biology, and more specifically the field of Ethnology, which attempt to understand the mechanism which animals use to demonstrate adaptive and successful behavior[Maes, 1995].

2.2.3 The Wooldridge-Jennings Agent

A computer system that has the following properties[Wooldridge et Jennings, 1994]:

- Autonomous: agents operate without the direct intervention of humans or others, and have some kind of control over their actions and internal state;
- Social ability: agents interact with other agents (and possibly humans) via some kind of agent-communication language;
- Reactivity: agents perceive their environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the INTERNET, or perhaps all of these combined), and respond in a timely fashion to changed that occur in it;
- Pro-activeness: agents do not only act in response to their environment, they are also able to exhibit goal-directed behavior by taking the initiative.

2.2.4 The Nwana's agent

[Nwana, 1996] defines an agent as referring to a component of software and hardware which is capable of acting exactly in order to accomplish tasks on behalf of its user.

Nwana points out that it is hard to define agent exactly. Because agents come from many different areas and agents can play many roles, such as personal assistants or know bots, which have expert knowledge in some specific domain.

To avoid having synonyms or plethora definition of agent, Nwana defines the agent by classifying the agents into different classes. He classifies the agents by three dimensions.

The first dimension is if the agent is static or mobile. The second dimension is if the agent is deliberative or reactive. Deliberative agents possess internal symbolic, reasoning model and they engage in planning and negotiation in order to achieve coordination with other agents. Reactive agents, on the other hand, do not have any internal, symbolic model of their environment, and they act using a stimuli-response type of behaviour by responding to the present state of the environment in which they are embedded.

The third dimension is if the agent is autonomous, learning or cooperative. In section 2.3.2, we will discuss Nwana's agent classification, which will further illustrate the opinion.

2.3 Agent Classification

The agents in Nego possess different properties, and their functions vary from simple to complex depending on their roles. To understand the functions of different agents in Nego, it is necessary to go through the most popular agent classifications because their agent skills are used to build agents in Nego.

2.3.1 AIMA agent classification

AIMA classifies the agents into four types, in the order of increasing generality; they are as following indicated [Russell and Norvig, 1995] [URL_31]:

- **Simple Reflex (Stimuli – Response) Agent:**

As showed in Figure 2.1, simple reflex agents response to stimuli immediately, but they can not learn. The "condition-action rules" in the diagram are production rules. These agents can help more intelligent agents in the multiple agent systems to implement some tasks.

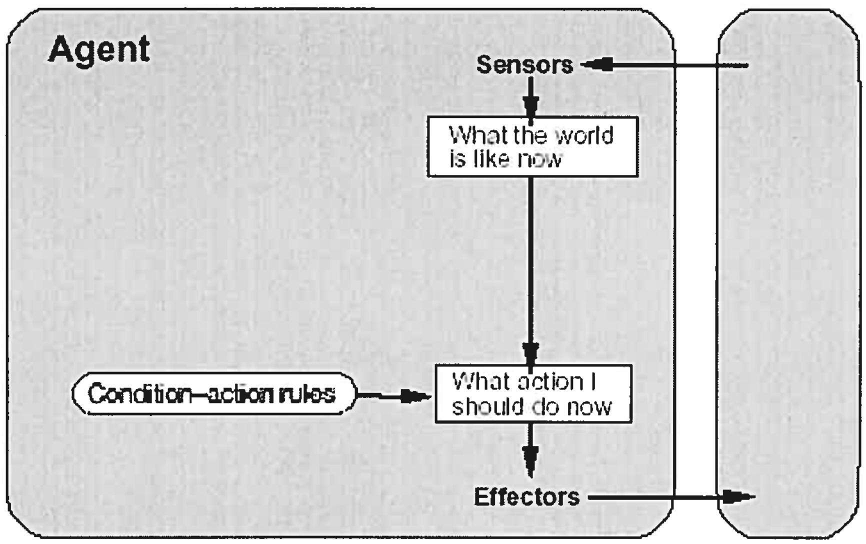


Figure 2.1: Simple reflex agent

- **Reflex Agent with States**

They can learn from earlier experiences. They can adopt the current information with current state and experience. Figure 2.2 shows these agents. These agents are more intelligent than the simple reflex agents. They can be used in consultant system.

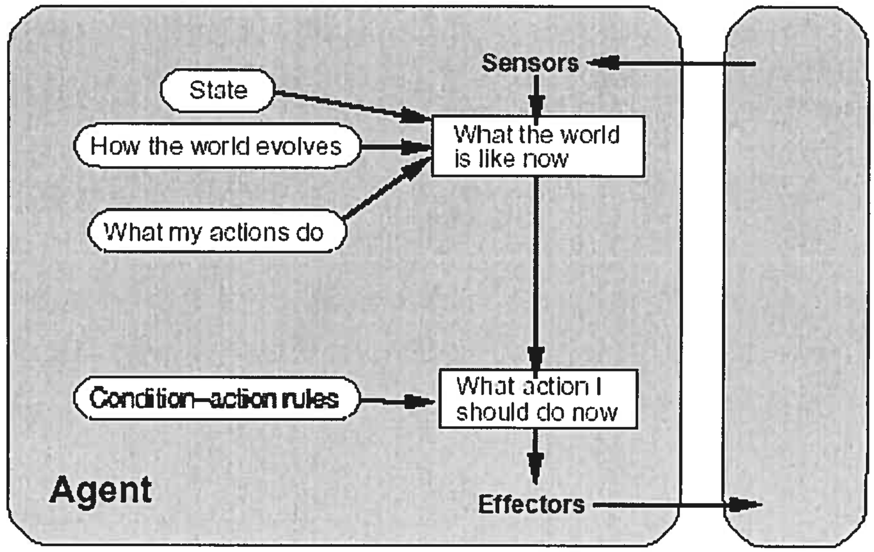


Figure 2.2: Reflex agent with states

- **Goal Based Agent**

Goal based agents can plan before they make decision. They have several search methods to search potential result for each proposal. Each result is evaluated to know which proposal can be the closest to their goal. Figure 2.3 shows these agents. The best example of goal based agent is the agent in chess game.

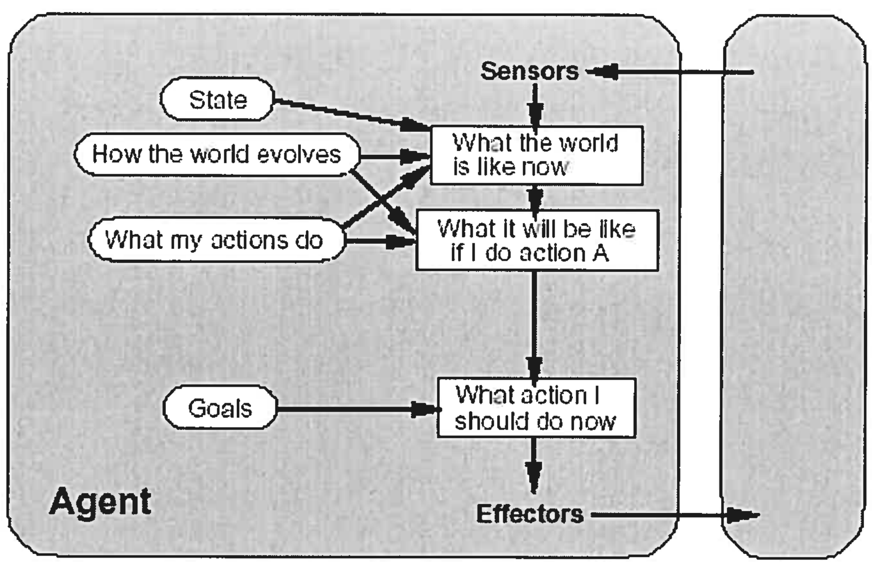


Figure 2.3: Goal based agent

- **Utility Based Agent**

Utility is a mathematics term of the personal evaluation or estimation of a specific object. The person has his utilities for different attributes of an object. These utilities usually are used with their probabilities to have a total utility of an object, and they entirely subject to the person. The utility based agent mainly refers to the utility when they take action. The difference of a Utility Based Agent from the other kind of agents is that the utility affects the agent to decide which action the agent will take. Figure 2.4 shows these agents. Utility based agents are the development trend of agent, our project will implement them

One of the advantage of the Utility Based Agent is that it can have its utilities of an object according to its (e.g., the agent's user's) subject of this object, and it can specify the utility for the important attributes while ignore the unimportant attributes in order to take actions which contribute the most to its profit (e.g., maximize its utility). For example, when a buyer wants to buy something, and before he decides which seller he will buy from, he can give each seller several attributes that relative to his buying decision such as "delivery time (D)", "after sales service (A)", "payment term (P)", and give each attribute a utility such as D1=10, A1=50, P1=0 (e.g., utility=0 means the user doesn't care about it), D2=13, A2=46, P2=0,..., Pn=0, ...Dn=10, An=45, Pn=1; after he sums up the utilities of all attributes for each seller such as U1=D1+D2+...,+Dn, U2=A1+A2+...+An, U3=P1+P2+...+Pn, he can compare the utilities from different sellers (e.g., compare U1, U2, U3) and find the maximum utility of seller, and decide to buy from him.

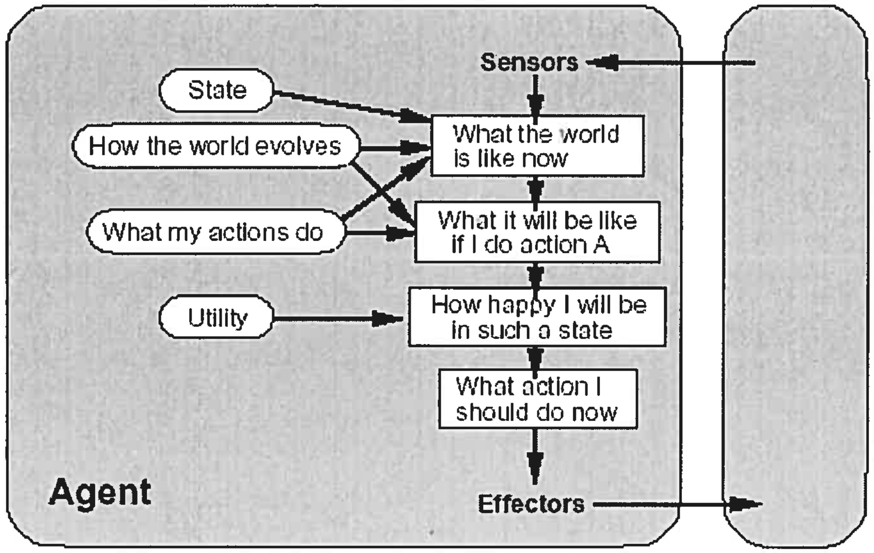


Figure 2.4: Utility based agent

In summary, the AIMA agents are classified on the agent's intelligent degree, from less intelligent to more intelligent. It is obviously that agent involves being more and more autonomous, and the utility agents are the current trend.

2.3.2 Nwana's Agent Classification

There are several dimensions to classify existing software agents [Nwana, 1996].

First dimension is the agent's mobility. Two classes in this point: static or mobile. Secondly, agents may be classified as either deliberative or reactive. Thirdly, agents may be classified along several primary attributes which agents should exhibit: autonomous, learning and cooperation.

Autonomous refers to the principle that agents can operate on their own without the need for human guidance. Learning means the agents would have to learn as they react and interact with their external environment, learning is the key attribute of any intelligent agent. Cooperation is a kind of social ability, it is necessary for multiple agents. In order to cooperate, agents need to possess social ability. Figure2.5 shows this classification.

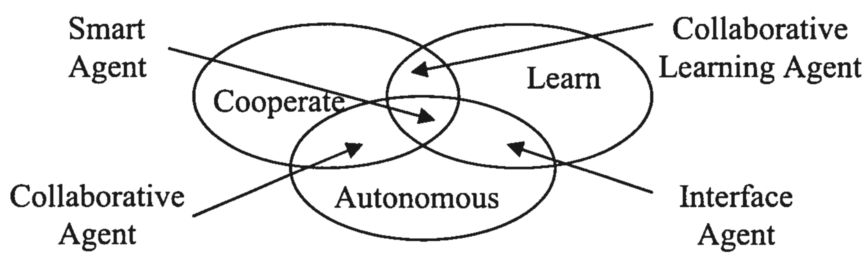


Figure 2.5: Nwana's agent classification by agent's primary attributes

Fourthly, agents can also be classified by the roles they play. For example, internet agents usually exploit internet search engines such as Spiders –WebCrawlers and Lycos [URL_25], to help manage the vast amount of information in wide area networks like internet.

Fifthly, the agent classification includes the category of hybrid agents which combine of two or more agent philosophies in a single agent.

In summary, Nwana identifies seven types of agents:

- Collaborative Agent
- Interface Agent
- Mobile Agent
- Information(Internet) Agent
- Reactive Agent
- Hybrid Agent
- Smart Agent

The Nwana's agent is classified on agent attributes, but collaborative learning agent is ignored because there does not exist an agent which collaborative and learn but not autonomous.

2.3.3 Dick Stenmark's Agent Classification

Dick classifies the agent according to the agent's function. He classifies the agent as the following [URL_26]:

- Interface Agent

Interface agent is used to guide the user using the interface. It helps the user communicate with the system and solve the user's problem such as wrong input.

- System Agent

System agent is the manager of a application, it controls the operating system, the storage, takes care of the security, puts the complicated operation in an order.

- Advisory Agent

Advisory agent is to give help when it is necessary, for example the diagnostics system.

- Filtering Agent

Filtering agent is used to filter information to abstract the necessary information

- Retrieval Agent

Retrieval agent searches and retrieves information to get the necessary information.

- Navigation Agent

Navigation agent works to navigate networks.

- Monitoring Agent

Monitoring agent used to check the appearance of one specific event.

- Recommender Agent

Recommender agent is used to give suggestions base on the profile.

- Profile Agent

Profile Agent is used to record the user's information such as user's personal information, preference, purpose and experience. It is also used to reflect the relation with other users.

We can see that the AIMA's agent classification is based on the agent intelligent, and the Nwana's agent classification is based on the agent properties, and the Dick Stenmark's agent classification is based on the agent's function. I the following, we introduce the agent applications in e-commerce.

2.4 Agents and their Application

[Lomuscio *et al.*, 2003] explains the term "electronic commerce" as: electronic commerce generally denotes an advanced step of modern commerce in which the functions of buyers and sellers are replaced by electronic entities.

[He *et al.*, 2003] estimates that the global market for software agents is 7.2$ million in 1997, is 51.5 million in 1999, will be 873.2$ million in 2004. From 1999 to 2004, the annual increase rate reaches 76.2%.

With the great amount of agents, what is their functionality? The following we discuss the e-commerce agents' type base on its functionality.

2.4.1 Functions of Agent

The agent applications can be divided into the following categories:

- Domain Expert

In the domain expert system, what an agent should do is not an agent developer's area of expertise; rather, it is a domain expert's expertise. The agent should take advantage of the domain expert's knowledge [Scerri *et al.*, 2000].

For example, the intelligent distribute environment for active learning (IDEAL) is a multi-agent active learning system. The system ties web clients (for students) and the underlying information servers (for courseware and student profiles) together with the multi-agent resource management. IDEAL consists of a number of specialized agents with different expertise. The personal agent acts on behave on a student who has profile and wants to take course. The course agent manages course materials and course specific teaching techniques for a course. The teaching agent acts as tutor of a course and gets its idea to teach a specific course from the course agent. In this case, the teaching agent is an expert agent who intends to find the best way to teach a student suitable for the student's background [Shang *et al.*, 2001].

- Decision Making

Based on domain information or retrieval information, the agent makes decision on behalf of the user given the specific strategy.

There are lots of examples for the decision making agents. The Negoplan is a multi-agent decision support application, it represents and stimulates the complex decision environment, enables the sequential, context dependant decision making methodology. Negoplan concerns self-training of a business administration student to test his managerial decision-making skills in an environment in which Negoplan represents two other autonomously behaving participants [Erkol, 1998].

- Search and Retrieval

The search and retrieval agent use the search engine to search and filter the information. For example, the Copernic is an agent which can access maximum 1000 search engine in 120 categories, it can create client profile, summarizing and analyze the search result [URL_08] [URL_09].

- Repetitive Activities

It is a good way for the agent to handle the repetitive routine work. For example, the agent "Compagnon Office" in the Microsoft Word in Windows XP is a help agent to guide the user using Microsoft word, its purpose is to detect mistaken operation and give assistance when the user needs help.

- Personal activity

This agent helps user to achieve personal goals. For example, Amazon helps the user find the useful thing such as CDs, books in the web by user's preference [URL_10].

2.4.2 Consumer Buying Behavior(CBB) Model

With the above mentioned agent functions, we use the CBB model to relate the agent functions with consumer buying behavior. Proposed by [Guttman *et al.*, 1998], the Consumer Buying Behavior (CBB) model is the most popular, formally categorized and accepted model involving in buying and selling behavior, which stemmed from traditional marketing research. CBB has six steps as shown in figure 2.6.

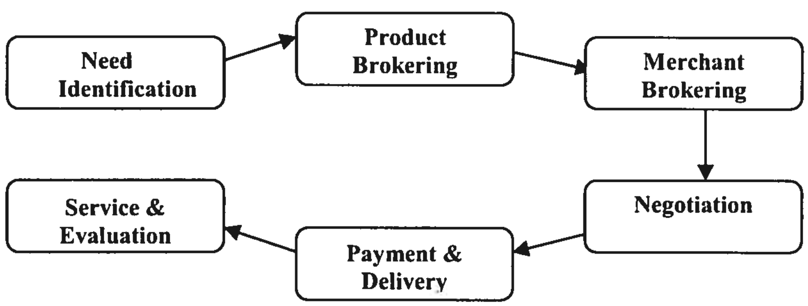


Figure 2.6: Consumer Buying Behavior Model

- **Need Identification**

This stage characterizes the consumer becoming aware of some unmet need. Within this stage, the consumer can be stimulated through product information[Guttman *et al.*, 1998]. The consumer keeps on receiving the update market information such as product advertising by email or notification, and will be stimulated to identify his need.

- **Product Brokering**

After the Need Identification stage, the need has been identified, based on the retrieved product information, the consumer decide what to buy with a critical evaluation. Usually, the consumer specifies his request for the product, the agent will feedback with the expected product list accordingly after an evaluation based on the consumer's criteria set.

- **Merchant Brokering**

Combined with the Product Brokering stage deciding what to buy, this stage decides whom to buy from. The consumer made a critical evaluation based on the retrieved merchant information and consumer's criteria, and chooses a merchant. Sometimes it is possible to change the product decided in the Product Brokering stage because of the merchant brokering [Pivk and Gams, 2000].

- **Negotiation**

This stage is to determine the terms of transaction. Transaction terms include price, product character, delivery, payment term, and warranty, but the most important term is the price. The price is an evaluation of all the factors to a specific product. It can be a condition to change the other negotiation terms. The consumer and the supplier reach an agreement on a single or multiple terms of the transaction to maximum their utility function based on intended gain.

- **Purchase and Delivery**

This stage can either signal the termination of the negotiation or occur sometime afterwards[Guttman *et al.*, 1998]. Sometimes the payment or delivery terms may influence product and merchant brokering. The delivery can be either online if the goods are in electronic format or by additional shipment if the goods are in physical format.

- **Service and Evaluation**

This stage includes after sales service provided by the merchant and an evaluation of overall buying experience and decision.

2.4.3 Agent Examples in CBB

Table2.1 shows the agents in the specific stage of existing e-commerce systems. We focus on the first four stages in CBB, which are Need Identification, Product Brokering, Merchant Brokering and Negotiation. Because these stages are more complicated, most of the existing agents are characteristic functional in these stages. The agents in the other two stages will be ignored in discussion here.

Table 2.1: Agents in CBB

CBB stages / Agents	Need Identification	Product Brokering	Merchant Brokering	Negotiation
Sales Mountain	X			
PersonaLogic		X		
Firefly		X		
Bargain Finder			X	
Jango		X	X	
Kasbah			X	X
AuctionBot				X
Amazon		X		
AllBookstores			X	
eBay				X
Yahoo!Auctions				X

❖ **Agents in Need Identification stage in CBB**

Agent in this stage is to help the user to find what he needs.

Sales Mountain [URL_11]

Sales Mountain is a place for comprehensive and current sales information, both nationally and locally. The agent in Sales Mountain helps user who are looking for certain items put "on sale", it filters the information according to the user's constrains and sending notification to the user. The slogan of the Sales Mountain is "find what's on SALE at your favorite local stores—The Nation's Instant Sales' Guide". The agents in Need Identification stage include discogs.com, netcactus.com, and querybot.com/shopping [Turban *et al.*, 2002].

❖ **Agents in Product Brokering Stage in CBB**

The agents in the brokering stage have the following functions: information retrieval and processing, learning from user, prediction of user requirements, matching merchant, and collaborating with other brokers, etc[He *et al.*, 2003]. The following are some examples.

PersonaLogic

The PersonaLogic guides the consumer through a large feature space in order to narrow down the products best meet the consumer's needs. Based on the consumer's constrain, it can filters unwanted products. It made product recommendations base on the prioritization of attributes such as price and delivery time by consumer [Guttman *et al.*, 1998] [Pivk and Gams, 2000] [Turban *et al.*, 2002].

Firefly

Firefly uses a collaborative filtering process to build profiles of people who visit a web site. Firefly provide users with a "passport" that identifies them when they visit sites participating in the Firefly program and recommend products/services to them. Based on people's preference, Firefly help marketers predict what customers were likely to want next. This allows marketers to reach out to consumers with a customized pitch that was cheaper and more effective than mass advertising [Turban *et al.*, 2002].

Amazon [URL_10]

Amazon provides very powerful product brokering function. It has different kind of search service: key word search, advance search, and power search. Furthermore, every buyer is provided with a "Shopping Card" to help the buyer to organize his potential shopping package. Except that, it provides recommendation service, and the buyer needs only to input his request to get his recommendation.

❖ Agents in Merchant Brokering Stages in CBB

Agents in this stage is to find the best merchant to match the product found in the Product Brokering stage.

BargainFinder [URL_13]
BargainFinder is the pioneering agent in merchant brokering. Just input the CD name, the agent will look in several online vendors and return the price. It had a big problem: the online vendors can "block" the agent's visiting if they don't want to compare the price [Turban *et al.*, 2002].

Jango [URL_14]
Jango is a newer agent who can solve this problem caused by BargainFinder. Because the vendors can identify the agent by identifying the site where the request was originated, Jango creates request from the user's site, therefore the vendors can't identify if the visiting is from the agent or from the user, therefore it can't "block" this visiting. Compared with BargainFinder, Jango has more product categories and it provides product review. Jango also functions in product brokering stage. There are some other agents having the similar function with Jango, they are Inktomi Shopping Agent, My Simon(mysimon.com) and Junglee [Guttman *et al.*,1998] [Turban *et al.*, 2002].

AllBookstores [URL_32]
AllBookstores provides the merchant with the lowest price for the new and used books after compares at least 2 dozen suppliers. First, the book information must be input; then, when the button "Compare" is pressed, it search the cheapest supplier. If the "buy" button is pressed, it will link the user to the supplier's site.

❖ Agents in Negotiation Stage in CBB

The negotiation stage is a good place to employ the agent technology. The agent functions on client profile building, decision making, and so on. The purpose of the negotiation here is to find current available information, and to bid base on its owner's utility and counterpart's utility to maximize its owner's benefit.

Kasbah

Kasbah is a multi-agent system which is a web site where users go to buy and sell things. The users create buying and selling agents with appropriate protocol, send them in the marketplace. These agents negotiate in the marketplace, with one of three negotiation strategies: anxious, cool-headed, and frugal, which corresponding to a linear, quadratic, or exponential function, respectively. The most interesting in Kasbah is its multi-agent aspect [Chavez and Maes, 1996].

AuctionBot [URL_15]

AuctionBot is a general purpose internet auction server in the University of Michigan. The users create new auctions by selecting from a set of negotiation type, and give the parameter such as participation number, discrete goods, bid rules, clearing schedule, closing conditions, quote schedule, allocation policy. The allocation policy dictates which agents transact, and at what price(s) [Wurman *et al.*, 1998].

eBay [URL_33]

The buyer clicks the "Continue Bid" button, and then inputs his user ID and password. He inputs his maximum bid, then click the "Place Bid" button. The agent in the eBay will bid on behalf of the user, and it will increase every time 0.5$ until the user's maximum bid. If somebody outbids you, you will fail in this auction; if you outbid the others, you will win and receive an email from eBay.

Yahoo!Auctions [URL_34]

There are two kinds of bidding. The first kind of bidding is the Automatic bidding [URL_04]. The buyer inputs its maximum bid, user ID, and password, and than the agent in Yahoo!Auctions will bid on behalf of the buyer. The increment of the bid depends on the range of the bid. For example, if the bid is $0.01-$0.99, the increment is $0.05, and if the bid is $100.00-249.99, the increment is $2.50. At the end of the bidding, if the buyer wins, the supplier will inform him by email. The second kind of bidding is straight bidding [URL_35], where the buyer specifies the exact bid, and there is no increment in it.

2.5 Discussion

From the above description, we can see that agents have played many functions, and especially in e-commerce area, agents have already involved in the first four stages of the CBB model. However, with the development of the agent technology, more new agents will involve in the e-commerce. For example, the Utility Based Agent introduced in section 2.3.1 is the more recent kind of agent in classification of [Russell and Norvig, 1995], where it is claimed to be the most intelligent agent.

The Utility Based Agent has some advantages compared with the other types of agent. One of these advantages is that it can give its utilities to the important attributes of an object, and ignore the unimportant attributes; therefore, it can focus on the relative attributes to maximize its profit. In another word, it can focus on the already known probability, and ignore the unknown probability (e.g., utility=0), which is very useful in negotiation because a negotiator usually has uncertainty toward his counter negotiator. The other advantage of Utility Based Agent is that it is very flexible and it can give utility to any attribute; therefore, it can be used in the simple case such as deciding the specific seller to buy from among different sellers by their attributes such as “delivery time (D)”, “after sales service (A)”, “payment term (P)”; furthermore, it can also be used in the complicate case such as in a negotiation which involves the history based data, probability, user’s gain and loss, relation with the sellers, concerns as if it is urgent or not, and so forth.

At the earlier stage of e-commerce, the agents mainly act for the product brokering and the merchant brokering. They collect useful information, and analyze information with the help of many intelligent search engines. These agents adopt intelligent methods such as customization when they process the information. Thus they can satisfy the user with useful information.

Although there are some negotiation agents in current e-commerce, they are not functional enough. This is because that negotiation involves complicated decision making, and the agents in the product brokering and merchant brokering stages can not solve the problem in negotiation stage. If we adopt the Utility Based Agent, as well as the decision making agent and monitor agent mentioned in section 2.4.1 and 2.3.3, into e-commerce negotiation, it can negotiate on behalf on the user and free the user from the complicate and time consuming negotiation. Based on this opinion, we create a utility based client negotiation agent, and we find it work well in implementing its user's mission and freeing its user from negotiation.

Chapter 3 E-commerce Negotiation

In this chapter, we will introduce the negotiation. We will define the negotiation, describe the negotiation types, discuss the online negotiation state, problem and develop trend.

Negotiation appears in every day. E-commerce is the most popular area where negotiation functions. When people make a deal, negotiation is a must. In our discussion, we mainly focus on autonomous e-commerce negotiation which means the negotiation online guided by agent.

In the view point of business, e-commerce can be classified into business-to-business(B2B), and business-to-consumer(B2C). B2B means commercial transactions, in which companies buy from or sell to other companies online. It is more than purchasing that it involves supply chain management. B2C implies the commercial transactions, in which companies sell products or service such as online banking, online consultant and so on to consumers for own use.

In the following, we introduce the negotiation types.

3.1 Negotiation Classification

[Dholakia *et al.*, 2002] [He *et al.*, 2003] classify e-commerce negotiation into two categories: B2C and B2B. We follow their classification to take a view at the negotiation tactics. B2C negotiation can be sub-classified into auction and bilateral negotiation. B2B negotiation can be sub-classified into buyer-controlled negotiation and seller-controlled negotiation. We will discuss these two classifications in the following sections.

3.1.1 B2C Negotiation Classification

B2C in e-commerce is the commerce model of business to consumer. The business here means the companies(e.g., seller) which sell products, the consumer means the individual consumer(e,g, buyer). Given this B2C model, we discuss the B2C negotiations in the following.

3.1.1.1 Auction

Definition of auction An auction is a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants [McAfee and McMillan, 1987].

Auctions are the most popular negotiation in e-commerce today. Compared with the traditional auction, e-commerce auction inherits the way traditional auction goes, but it does not have limitation of the traditional auction (the limitation of the traditional auction is mainly caused by the problem of the location. Because the Internet makes the world like a village, online auction in current e-commerce consumes less time, and the products to be auctioned are more than before).

With the e-commerce developing, online auction becomes more and more popular. They can be grouted into single-side auction and double-side auction. The single-side auction means that only buyer or seller submits the bid or offer at the same time. In double-side auction both buyer and seller can submit the bid and offer at the same time. The most popular single-side auctions are: English auction, Dutch auction, first-price sealed-bid auction and Vickrey (also called second-price sealed-bid) auction. The double-side auction has the Continuous Double Auction as an example. In the following we will discuss each of them.

- **English Auction**

The bidding price increases till there is only one bidder remaining who bids the maximum price, thus the offer will give to this bidder. In this kind of auction, because all bids are open, the strategy for a bidder is to bid a little more than the current bid of other bidders. The bidder continues to bid till reach his maximum acceptable price.

The English auction is the most common auction form. The word “auction” comes from Latin “augere” meaning “increase” which is the English auction’s basic principle. An example of English auction is “Yahoo” auctions “automatic bidding” and “eBay” auction. In “Yahoo” Auction “automatic bidding”, the bidder tells the auctioneer (e.g. somebody who monitors this auction) increment value and maximum value he is willing to pay for that product, his bid will automatically increase by his increment during the auction[URL_04].

- **Dutch Auction**

The price starts with high by auctioneer, and decreases till there is a buyer signals that he will accept the offer. The Dutch auction is the reverse of the English auction. The difference between these two is that the English auction begins with lower price by the bidder, and the Dutch auction begins with higher price by the auctioneer.

The Dutch auction is famous from the flower market near the airport in Amsterdam, Holland, where near 60% cut flower of the world is sold to other place every year [Dholakia *et al.*, 2002]. Dutch auction is better to be used in the easy evaluated product just as in the Amsterdam’s flower market auctions, there a big amount of flowers transact in a quick way.

Dutch auction is also better in the rare product auction. For example the jewellery market adapts Dutch auction. This is because Dutch auction gives only one opportunity to each bidder to bid, when the product is rare, the bidder, afraid of that others may get it, prefers to bid a little higher than his evaluation so that it is more possible to get the term.

- **First-Price Sealed-Bid Auction**

With this kind of auction, every bidder has only one chance, the bids are not open. The highest bidder wins. The basic difference between the First-Price Sealed-Bid auction and the English/Dutch auction is that the bids in First-Price Sealed-Bid auction are not open, and the bidder has no idea about the other bidders' bids. Given this situation, the bidder bids base on his evaluation for the product, the estimation of the risk and opinion on other bidders.

The example of the first-price sealed-bid auction is that, it is used in the auction of mineral rights to U.S. government owned land, artworks, and real estate [McAfee and McMillan, 1987].

- **Vickrey Auction**

Also known under the name of Second-Price Sealed-Bid, Vickrey auction rule is similar to the first-price sealed-bid, but the winner (i.e., the bidder with the highest bid who will get the offer) should pay the second highest bid instead of his bid. Vickrey Auction exists in theory, but there is seldom example in practice.

- **Continuous Double Auction (CDA)**

The buyer and seller can give a bid or an offer at any time during a trade period. Michigan auctionBot uses CDA combining with chronological match policy [Wurman *et al.*, 1998]. SouthamptonTAC also uses it in its negotiation of "entertainment" in a travel [He and Jennings, 2002]. Referring to [He *et al.*, 2003], table3.1 shows the properties of each type of auctions.

Table 3.1: Auction properties (FPSB--First-price Sealed-bid, CDA--Continuous Double Auction)

Auction	Format	Bid change by	Bid Release	Time Limit	Closing condition
English	One-side	Increase	Open	No	No more bidder
Dutch	One-side	Decrease	Open	No	First bid
FPSB	One-side	Once only	Close	Yes	Time out
Vickrey	One-side	Once only	Close	Yes	Time out
CDA	Two-side	Not fix	Open	No	No more bidder

3.1.1.2 Bilateral Negotiation

Bilateral negotiation involves two parties, buyer and seller, to reach a mutual agreement, it usually concerns with multiple attributes contract. There is not a dominant mode for it, but it generally falls into the following three types [He *et al.*, 2003]:

- Decision making by explicitly reasoning about the opponent's behaviour
- Decision making by finding the current best solution
- Argumentation

The first type, decision making by explicitly reasoning about the opponent's behaviour, is to explicitly reason its opponent's objective and behaviour, and find the best way to correspond to the opponent's behaviour. The second type, decision making by finding the current best solution, is to correspond to maximum the user's own profit by given its condition and preference, etc.

The argumentation is on the opinion that, in the multi-agent system, agents often have no inherent control over one another and so the only way they can influence one another's behaviour is by persuasion. In another word, the persuadee may be unwilling to accept the proposal initially and must be persuaded to change its beliefs, goals or preferences so that the proposal, or some variant thereof, is accepted. In this case, the minimum requirement for negotiation is for the agents to be able to make proposals to one another [Sierra *et al.*, 1997].

3.1.2 B2B Negotiation Classification

The B2B e-commerce involves the business between company and company. It is different from the B2C e-commerce that the B2C most involves retail sale, the B2B most involves whole sale. Given the difference between B2C and B2B, we discuss the negotiation in B2B environment in the following sections.

3.1.2.1 Buyer Controlled Negotiation

The negotiation usually involves the particular product or service specified by the buyer, the buyer designs the maximum price of the item to pay and the sellers bid for the lowest price under the condition that they hope to win. Given this, the market power is shifted to buyer's side. It is also called reverse negotiation [Dholakia *et al.,* 2002].

Buyer controlled negotiation usually are for the great amount of special product that the specific buyer needs only. The buyer specifies the standard and looking for the seller. The examples are the General Electric [URL_19] and Boeing Inc. [URL_20].

3.1.2.2 Seller Controlled Negotiation

The seller wants to sell something and some buyers bid to win. The aim of this market is to create seller's market powering the transaction.

The B2C auction strategies can be used in seller controlled negotiation [He *et al.*, 2003]. The example is the FreeMarkets run Dutch auctions several times a day in seller-controlled markets with many buyers [Dholakia *et al.,* 2002].

3.2 The Development of Online Autonomous Negotiation

The features of personification and autonomy enable agents to simulate the process as it happens already between humans, and hence human negotiation strategies and approaches may be easier to translate to it [Wang *et al.*, 2002].

There are lots effort already put into the research and many methods were proposed in the development of the online autonomous negotiation. There are some researches which worthy of being introduced. In the following we discuss them. They are Kasbah from MIT media lab, auctionBot from Michigan University, and the Bazar.

3.2.1 Kasbah

The Kasbah virtual marketplace is an early application in the web where users create autonomous agents to buy and sell goods on the users' behaves. This kind of earlier negotiation agent is not smart, what makes Kasbah fundamentally interesting is its multi-agent aspect—the interaction and competition between the agents in the marketplace [Chavez and Maes, 1996].

The user creates buying or selling agent and sends it to the market, the agent is given by the parameters about the criteria of the user such as the transaction dead line and the price limitation. The agent goes to the market and does negotiation for its user.

The advantage of the Kasbah agent is that it is proactive, interactive, and competitive. As an early and primitive negotiation model, Kasbah explores and implements a basic negotiation protocol.

One of problems of the Kasbah agent is its negotiation strategy. It adopts three bidding or offering strategies: linear, quadratic and cubic. In these three strategies, the agent doesn't learn, it has no idea of its experience and its counter negotiator. Moreover, the agent's strategy does not reflect the agent's preference, in which the relation exists between the user and his counter negotiator.

3.2.2 Michigan AuctionBot

According to [Wurman *et al.*, 1998], AuctionBot is a price based negotiation platform, it supports a wide variety of auction types. For example, it supports the English auction, first-priced sealed-bid auction, second-price sealed-bid auction, and continuous double auction.

The auctionBot's advantage is to use orthogonal parameters to present different aspects of the auction. These parameters are categorized as "Bidding Restrictions" (i.e., what kind of bid is acceptable), "Auction Events" (i.e., what are the clearing schedule, closing conditions and quote schedule), "Information Revelation" (i.e., what is the price quote, if the past transaction information and the schedule can be published or not), "Allocation Policies" (i.g., the policy decides which agent transacts, at what price).

Among the above mentioned parameters, the "Allocation Policy" is creative. Based on it, the auctionBot supports multiple policies: Mth-price policy, (M+1)th-price policy and chronological match policy.

The Mth- and (M+1)th-price policies come from the first- and second-price sealed-bid, M refers to the number of units offered for sale. Bids are sorted by price, and the auction counts down M or (M+1) units. The chronological match policy implements the sequential effect, the portion of the new bid that is satisfied transact first, the remaining is put into the waiting list for incoming bid.

The auctionBot can construct most of the classic auctions from the above mentioned three rules. The English auction and the first-price sealed-bid auction can be implemented by Mth-price policy, the Vickrey auction can be implemented by (M+1)th-price policy, the CDA can be implemented by chronological match policy. This is the most important achievement of auctionBot.

3.2.3 Bazaar

Bazaar in [Zeng and Sycara, 1998] proposes "Sequential Decision Making" paradigm in which it constructs a learning agent capable of reasoning base on experience. The basic idea is that: most negotiation involves multiple rounds of proposal and anti-proposal.

Current decision making is based on the result of the previous stages. After implement this decision making, the agent has to update its knowledge to be used in the next decision making. For example, the buyer agent can update his knowledge of the seller's reservation price after each round of the proposal and anti-proposal to deduce the seller's reservation price. The advantage of the Sequential Decision Making paradigm is to implement the learning in negotiation to develop new solution. It uses the Bayesian belief network to model the experience and proposal.

3.2.4 Others

There is much research focus on the price negotiation strategy. [Anthony and Jennings, 2003] proposes a methodology using in the multiple heterogeneous auctions. The agent can use any of the three auction types: English, Dutch and Vickrey to bid at any time. Which auction type will choose depends on the product of the probability of the agent winning in that auction at the given bid value and the value of the agent's utility function. The utility is based on four aspects of the auction and their weights respectively. These four aspects are: the remaining time tactic, the remaining auctions tactic, the desire for bargain tactic, and the desperateness tactic. The user can adjust these four weights as he likes. The advantage of [Anthony and Jennings, 2003] proposal is that it gives one way to produce bidding protocol dynamic, and it decides the utility by four different aspects (it is a open model, to which additional aspect can be added to), each user can adjust each aspect by the parameter and weight, which reflects the user's personality.

Present negotiation trends to focus on intelligent strategies and multiple attributes. The SouthamptonTac is an agent in the Second International Trading Agent Competition (TAC), in which it ranged the highest mean score and the lowest standard deviation in the competition after 600 games.

SouthamptonTAC generates eight agents on behalf of eight users to negotiate for: flight (e.g., 2X8 flights), hotel (e.g., 8 hotel rooms) and entertainments (e.g., 8X12 entertainment tickets) of a travel. The negotiation is not easy because these terms are independent and the environment is uncertainty. The terms' independent means that the flight must relate with the hotel, if there is no hotel available the flight should be cancelled, and vice verse. The uncertainty means the prices of the flight and hotel change respectively at any time, and the other hotel's price and state (e.g., the open or close of the hotel) will affect the hotel's price.

SouthamptonTAC separates a game into three phrases: "probing", "decisive", and "finalisation" stages. In the "probing" stage, the agent buys flights which has a high probability of need, and buys a part of hotel rooms and entertainment tickets by estimation, experience and preference. In the "decisive" stage, the demand of various auctions is clearer, and the allocation of the hotel is more accurate, the agent can fix more hotel rooms. In the "finalisation" stage, it is the clearest time to buy and also the last chance for the agent to buy, for example the entertainment ticket and the returning flights.

In SouthamptonTAC, the different term's negotiation applies the different strategies. In the *flight negotiation*, suppose the flight price changes from time to time and the earlier flight the cheaper, and in the time every 24 to 32 seconds, the changing of price is divided into four categories depending on price changing range: [-10, 15], [-10, 30], [-10, 60], [-10, 90]. Each category applies different rule of flight negotiation. For example, if the price changing falls in the first category [-10, 15], that means the flight price changes a little, in this case the agent doesn't need to buy the flight much early, but if the price changing falls in the forth category [-10, 90], the agent should buy the flight early.

In the *hotel negotiation*, the agent applies two strategies:

- Reasoning on hotel demand: it is based on the basic laws of microeconomic: the higher demand the higher price.
- The fuzzy reasoning: it is based on 39 prediction rules considering the following factors:
    - The target hotel ask price
    - The competitive hotel ask price
    - The competitive hotel closing time
    - The current time

- The target hotel ask price changing rate

In the above rules, the ask price can be expressed in fuzzy set: high, medium, and low. The price changed can be expressed in fuzzy set: quick, medium, and slow.

In the *entertainment ticket negotiation*, the CDA (e.g., continuous double-site auction, refer to 1.1.1 Auction) is used. And it uses the fuzzy set to extend the reservation price.

The contribution of SouthamptonTAC is to use different strategies in different terms and stage, and gives a useful way to combine these different strategies. Its reasoning strategy and concept can be broadly applicable in the independent multiple attributes auction. [He and Jennings, 2002], [He and Jennings, 2003].

3.3 Discussion

There are multiple kinds of negotiation, and different kind of negotiation is used in different situation. Each negotiator needs to consider how to give his proposal so that he can maximum his profit and be accepted by his counter negotiator. To make decision, it is important to decide which factor will be considered, the weight for each factor, and how to combine all factors together.

The most important thing for price negotiation is the strategy. No matter which negotiation protocol, such as English Auction, or Dutch Auction, or the protocol of AuctionBat, or the other negotiation protocols, the negotiator should have his strategy to give his proposal-deal. That means the negotiation strategy should be reasonable and personal, which relate to the buyer's budget (e.g., how much he can pay), his evaluation of the object (e.g., how much it worth. Normally, the buyer's evaluation of the object bases on the history data, e.g., his experience.), how much does he get (e.g., in his point, there are "gain" and "loss"), how he likes the seller (e.g., he prefers seller A more than seller B or seller C), how urgent he needs the object, and so forth. Unfortunately, most of the current e-commerce negotiation system can not make the negotiation strategy like this.

For example, Kasbah just simply increases the bid with a fix rate, and it does not have an efficient way to give a reasonable and personal proposal-deal; therefore, it does not reflect any user's preference. In this way, Bazaar is better because it gives a way to deduce the proposal-deal in the model of Bayesian network basing on his user's previous decision made, so it is history-based, but it does not reflect the user's current preference such as "gain" and "loss", if he like the seller or it is urgent. After studying the bidding strategies used in Kasbah and Bazaar, we find that it is possible to give a better bid in regarding both the user's history-data and the user's preference.

The Utility Based Agent mention in 2.3.1 in "Utility Based Agent" can help us to realize the reasonable and personal strategy. As we discussed in section 2.5, the Utility Based Agent can handle multiple attributes well. To consider the attributes such as buyer's budget, his evaluation of the object, how much does he get, how he likes the seller, and how urgent he needs the object, Utility Based Agent can give the reasonable utility to them according to his user.

In our project, we create a negotiation system, Nego, to make the negotiation strategy more reasonable and personal by the help of the utility based agent.

Chapter 4 Nego's Theory and Methodology

4.1 Overview

Nego is a platform for e-commerce with multi-agents and negotiation strategies. It provides a tool for the buyer to negotiate for the product's price that has found in the virtual market and is interested in buying. Suppose that the server side is a virtual market, in which multiple merchants provide similar products and seek buyer; at the same time, the buyer can connect to the server side from his individual side, which is called client side, to search for his intended product and negotiate for its acceptable price.

In Nego, mobile negotiation agent is created to be sent to the server side to negotiate on behave of the buyer. It is equipped with parameters assigned by the buyer, and it can check data in the client side at any time, and it can work in the server side. In this situation, what kind of strategy it will adopt to utilize its advantage of check the information in the client side at any time? How it makes decision base on the buyer's experience and preference?

This chapter mainly illustrates Nego's protocol and the buyer negotiation strategy. Nego focuses on giving a proposal deal in a negotiation, and what factor will be considered to give a proposal deal when negotiate. Here, we introduce the buyer's strategies and the protocol of Nego. In section 4.2, we illustrate the requirement for the negotiation model to implement Nego's strategy. In section 4.3, we illustrate the component of Nego's model. In section 4.4, we explain Nego's negotiation protocol. In section 4.5, we indicate who will participate in a single negotiation. In section 4.6, we introduce two kinds of negotiation strategies implemented in Nego, "Simple Negotiation Strategy" and "Learning-Preference Negotiation Strategy", and compare them. In section 4.7, we introduce the close conditions. In section 4.8, we compare Nego with some important systems of e-commerce negotiation. In section 4.9, we summarize Nego's procedure and strategies.

4.2 The requirement for Negotiation Model

An ideal negotiation model should possess the following factors:

- It can present the negotiation in an available way,
- Its required computational resource should be easily obtained,
- It should support the agent's learning capability,
- It should support the dynamic negotiation.

With dynamic negotiation, Nego can implement the following two functions:

- The current decision is based on the previous decision points,
- After each negotiation, the knowledge base is updated to be referred by the decision maker for the next decision making.

4.3 Component of Negotiation Model

The components we consider in the negotiation process are:

- Negotiator,
- Proposal-deal(e.g., bid or offer),
- Counter-proposal-deal(e.g., bid or offer).

The "proposal-deal" is very important, because the "proposal-deal" can be changed by the negotiator to become a final deal. The "negotiator" means either the buyer or the seller. The "counter-proposal-deal" is a response by the buyer or seller to change the proposal deal. "Counter-proposal-deal" is relative to a proposal-deal, in fact, it is also a proposal-deal.

4.4 Negotiation Protocol

Referring to [Kumar and Feldman, 1998], the negotiation protocol can be explained as in figure 4.1. The state of the "proposal-deal" includes "start", "sellerOffer", "buyerBid", "abort", "deal" states. The message sent by the negotiator to change the state of "proposal- deal"(e.g., is a counter proposal-deal when it is a bid or offer) can be "offer", "bid", "sellerAccept", "buyerAccept", "sellerWithdraw", and "buyerWithdraw".

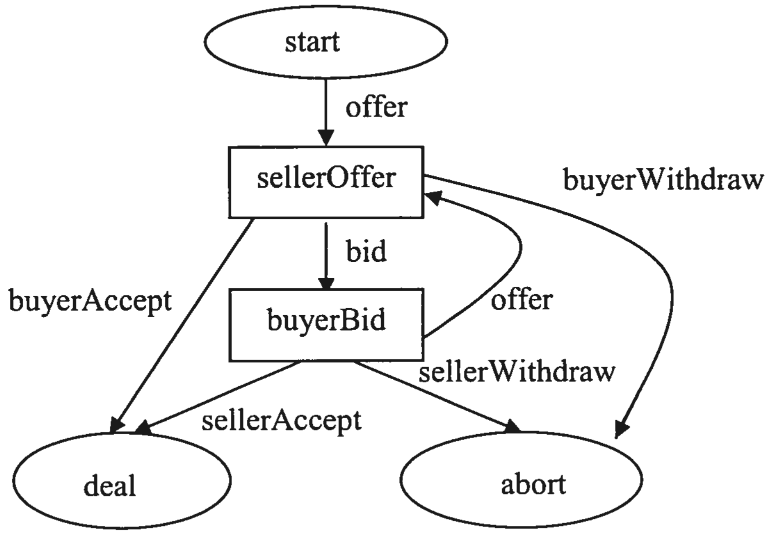


Figure 4.1: Negotiation protocol

The negotiation starts from the "start" state, when the seller offers, it is in the "sellerOffer" state, then when the buyer bids, it is in the "buyerBid" state, if the seller offers again, it turns back to the "sellerOffer" state again, this is a loop until one of the following situations: the buyer accepts the offer, or the seller accepts the bid, or the buyer withdraws the bid, or the seller withdraws the offer. Figure 4.2 shows negotiation processing sequence relative with this protocol.

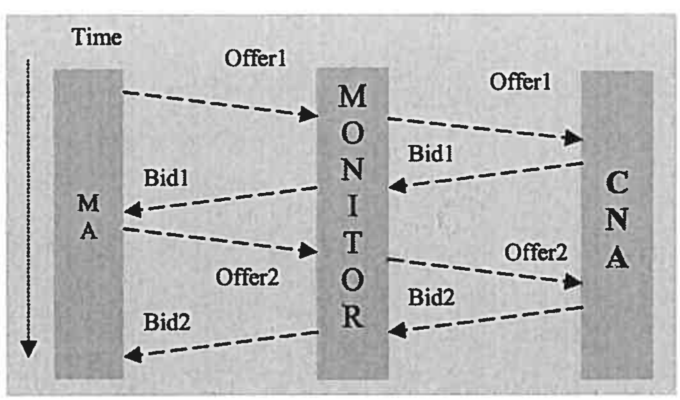


Figure 4.2: Negotiation processing sequence (MA: merchant agent, CNA: client negotiation agent)

The MA proposes a proposal-deal(offer), then the CNA proposes a counter-deal(bid), then the MA proposes counter-deal(offer), this is a loop until the negotiators make a deal or the negotiation aborts.

4.5 Negotiation Participant

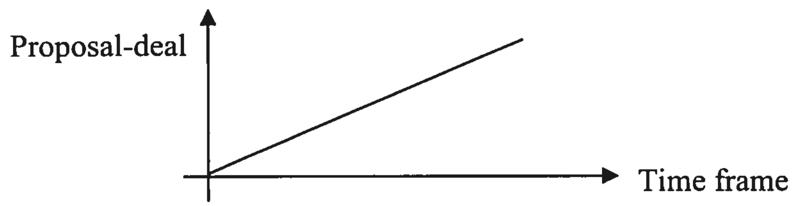
To construct a mobile negotiation agent and test the negotiation strategy, our negotiation model involves one buyer and several sellers who can negotiate at a time. Suppose a buyer gets the product and merchant brokering result from Pacha, and the result includes a set of merchants and their products, so the buyer needs to negotiate with these merchants. In our system, it is the client negotiation agent (CNA) who participate the negotiation on behalf of its user. The buyer only inputs his criteria, and the CNA collects the buyer's criteria and performs the negotiation. On the other hand, there are seller agents who negotiate with the CNA on the seller's behalf. After the negotiation, the CNA informs the buyer with its negotiation result.

4.6 Negotiation Strategy

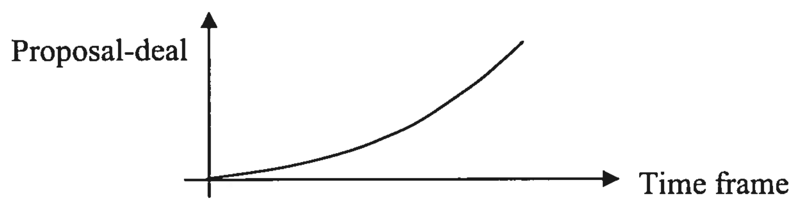
We employ two strategies. They are the Simple Negotiation Strategy and the Learning-Preference Negotiation Strategy.

4.6.1 Simple Negotiation Strategy

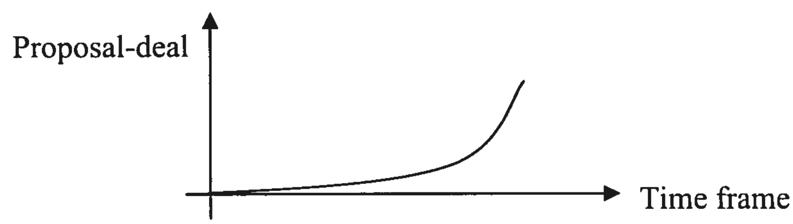
In Kasbah, there are three negotiation strategies for buyer and seller, respectively. These strategies are anxious, cool-headed and frugal, which are corresponding to a linear, quadratic, or exponential function. Please refer to Figure 4.3 for these three strategies [Chavez and Maes, 1996]. In the anxious strategy, the buying agent quickly increases its bit when time passes; in the frugal strategy, the buying agent slowly increases its bid when time passes; in the cool-headed strategy, the buying agent increases its bid quicker than the frugal strategy and slower than the anxious strategy.



Anxious buyer strategy in Kasbah



Frugal buyer strategy in Kasbah



Cool-headed buyer strategy in Kasbah

Figure 4.3: Kasbah negotiation strategies

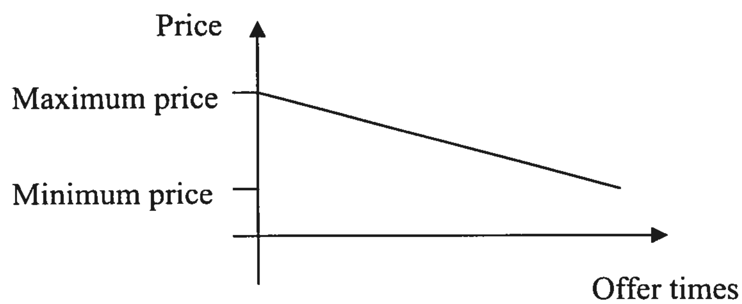
One example of Kasbah's Anxious Buyer Strategy is as table 4.1:

Table 4.1: Example for Kasbah strategy

Time	21:20:15	21:20:34	21:20:53	21:20:54	21:21:12	21:21:31	21:21:50
$Offer_{seller}$	100	96	91	91	87	83	79
Bid_{buyer}	70	73	75	75	78	81	79

Nego negotiation Simple Negotiation Strategy is indicated in figure 4.4. It is similar to the Anxious strategy in Kasbah. The buyer just specifies the maximum price, minimum price and the rate of increasing. The difference of Nego's Simple Negotiation Strategy from Kasbah is that the increment "m" of Nego's bid is a fix value, and every bid will increase "m", but the Kasbah's bid is variable with time. For example, it is 5, so bid_{i+1}-bid_i=5; in Kasbah, bid_1-bid_2 = 3 =/= bid_3-bid_2 =2.

In Nego, when the CNA and the seller agent begin negotiation, the seller agent proposes a deal, the CNA accepts or rejects. If the CNA rejects, the CNA increases the bid by a specific value and gives a new bid. The seller agent accepts or rejects. If the seller agent accepts, the negotiation finishes. If the seller agent rejects, the seller agent decreases the offer by specific value to give a new offer. The negotiation continues until the seller and buyer agents agree with each other or the negotiation stops.



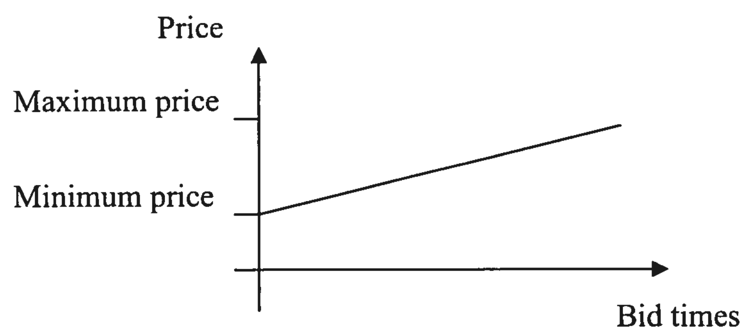


Figure 4.4: The Simple Negotiation Strategy for buyer

In the following, we will propose another negotiation strategy "Learning-Preference Negotiation Strategy" and discuss its advantage on how to decide the proposal-deal base on experience and preference.

4.6.2 Learning-Preference Negotiation Strategy

Learning-Preference Negotiation Strategy is based on the negotiator's experience and preference. We propose this strategy to develop a dynamic computational negotiation strategy that can learn from the experience.

Before the negotiator gives a proposal-deal, it must consider two factors. First, it must consider the possibility for the proposal-deal to be accepted. Second, it must maximize the negotiator's profit in the transaction. The strategy we will introduce should focus on these two factors.

Definition [Berger, 1988]: Utility is to work mathematically with ideas of "value", it will be necessary to assign numbers indicating how much something is valued. Such numbers are called utilities, and utility theory deals with the development of such numbers.

For example, when a buyer wants to buy something, and before he decides which seller he will buy from, he can give each seller several attributes that relative to his buying decision such as "delivery time (D)", "after sales service (A)", "payment term (P)", and give each attribute a utility such as D1=10, A1=50, P1=0 (e.g., utility=0 means the user doesn't care about it), D2=13, A2=46, P2=0, ...Dn=10, An=45, Pn=1; he can sums up the utilities of all attributes for each seller, and gets the seller's utility: U1=D1+A1+P1, U2=D2+A2+P2, U3=D3+A3+P3, ..., Un=Dn+An+Pn.

Utility function [Berger, 1988]: Often there is uncertainty as to which of the possible consequences will actually occur. Thus the results of actions are frequently probability distributions on R. Let P denote the set of all such probability distributions. It is usually necessary to work with values and preferences concerning probability distributions in P. This would be easy to do if a real-valued function U(r) could be constructed such that the "value" of a probability distribution p< P would be given by the expected utility Ep[U(r)]. If such a function exists, it is called a utility function.

The example of the utility function is: suppose we have the utility function U(Price) = $Price^2$+ Price+1, 10 0>Price> 200, then from it, we can have the utility for each Price.

The relative probability is up to the user's expectation, for example, when Price=150, then U(150) = 22650, when there are 5 "Price" such as Price1=120, Price2=130, Price3=130, Price4=140, Price5=150, then the probability for Price=150 is 0.2, then we can get the probability of Price(150)=0.2.

In Nego, we concern how to produce a buyer's bid(price), a proposal-deal. If we have the utility values related to the known probabilities, and can construct a set of expectation(price) of the negotiation, in which a specific "expectation(price)" value has the maximum, then the specific "price" is our preferred proposal-deal.

We define the negotiator's "expectation" as:

$$expectation = \sum_{i=0}^{n} probability(price, outcome(i)) * utility(price) \qquad (1)$$

In equation (1), the "outcome" means the negotiation result, in our project there are two possible outcomes: proposal-deal is accepted, outcome(0), and proposal-deal is rejected, outcome(1). "Utility" means the negotiator's expecting payoff. It has two meanings here. First, in the case of the proposal-deal is accepted, the "utility" means how much the negotiator will gain. Here the "utility" is passive. Second, in the case of the proposal-deal is rejected, the "utility" means how much the negotiator will lose. Here the "utility" is negative. Accordingly, equation (1) equals to:

$$expectation = probability(price, accept) * gain(price) - probability(price, reject) * loss(price)$$

$$\text{where } probability(price, accept) = 1 - probability(price, reject) \qquad (2)$$

4.6.2.1 Probability, Preference and Utility

In the following, we explain how to evaluate the probability and utility in equation (2).

The "probability" can be calculated base on the previous sequential negotiation result (they are history-based data. For a specific product, we record all previous final-deals). Suppose we have the previous sequential negotiation result, we calculate all the previous probabilities. For example, if we want to negotiate for product A, assume that we have 10 previous final-deal records for product A (they are random data just for giving an example). They are 80$ 1 time, 90$ 2 times, 100$ 3 times, 110$ 3 times, 120$ 1 times, as indicated in table 4.2, which means for this user, he has bought this products five times up to now, and his profile keeps all the five records. All of these five records will be used in equation (2) to calculate the user's utility.

Table 4.2: The previous records

Price	80	90	100	110	120
Times	1	2	3	3	1

Another more complicated example is in table 4.3:

Table 4.3: The other previous records example

Price	96	87	84	83	94	88	60	105	121
Times	2	3	5	4	4	4	1	5	1

Then we can have the following probabilities for table 4.2:

probability(80$, accept)=0.1, probability(80$,reject)=1-probability(80$,accept)=0.9,
probability(90$, accept)=0.2, probability(90$,reject)=1-probability(90$,accept)=0.8,
probability(100$, accept)=0.3, probability(100$,reject)=1-probability(100$,accept)=0.7
probability(110$, accept)=0.3, probability(110$,reject)=1-probability(110$,accept)=0.7
probability(120$, accept)=0.1, probability(120$,reject)=1-probability(120$,accept)=0.9

Then if given the gain(proposal-deal) and the loss(proposal-deal), we can use equation (2) to calculate the expectations for each price.

How to evaluate "gain" and "loss"?

"gain" and "loss" is a negotiator's expectation. They are the negotiator's expecting evaluation of the payoff subjecting to individual negotiator. They reflect the way that how the negotiator evaluates the payoff (e.g., if he accepts this price, how much he gets? If he rejects, how much he loses?). Its purpose is to reflect the negotiator's preference.

There are a lot of ways to assign the "gain" and "loss". In Nego, the "gain" and "loss" are kept with previous final deals in the profile. Nego refers to the client profile for previous evaluation of "goal" and "loss", then asks the user adjust to them:

gain=gain(in client profile)+adjust value
loss=loss(in client profile)+adjust value

The gain and loss in client profile is assigned by the user after he made a final deal. The adjust value is needed when the user wants to adjust his "gain" or "loss".

For example, the buyer can assign the "gain" and "loss" value by the following concern "1" and adjust the "gain" and "loss" focusing on concern 2 and concern 3:

1. What relation the buyer expects to with the seller. We consider if the buyer cares of the long term partnership instead of temporally lower price or not.
2. How much is the buyer's budget. That means the buyer will abort from the negotiation if the price is higher than its budget.
3. How important the product is to the buyer. If the buyer needs the product urgently it is possible for him to accept the offer in a much higher price.

For the above concerns, we let the user to choose in the user interface before the negotiation agent is created.

With the different concern, we give the different user "gain" and "loss" evaluations. We explain the way to do with the concern in the following.

Concern 1:

gain: If the buyer wants to have a long term relation with the merchant and does not care much of the price, the "gain" is high at the beginning, because it is better to get it as soon as possible.

loss: If the buyer loses the deal, he will lose the chance to build the relation with the seller. He thinks he will lose a lot.

The "gain" and "loss" in the client profile are assigned by this concern. After negotiation, we will modify the client profile. We will add the final deal in the client profile, and give it the "gain" and "loss" value. How to decide the "gain" and "loss" depends on the desired relation with the merchant. If the relation is "good", the "gain" value is high and the "loss" value is low. If the relation is normal, the gain and loss are neither high nor low. For example, if the relation is "good", gain=100, loss=20. If the relation is "normal", the gain=50, loss=40.

Concern 2: we limit the proposal-deal between the minimum and maximum scope.

Concern 3: the "gain" value and "loss" value depend on the importance of the product to the buyer.

gain: If the product is very urgent to the buyer, and if the buyer gets it earlier, so the threaten of "not get it" is smaller, therefore the "gain" value will be always higher.

loss: We evaluate the "loss" relating to the expected cost such as time and effort to get the product. The "loss" here we consider if we lose a deal now, how much time and effort we should pay later to have another round of negotiation or look for another merchant. In the lower proposal-deal the "loss" is lower, in the higher proposal-deal the "loss" is higher, this is because if the buyer fails in making a deal in a lower price (for example 100$) it is more possible in making a deal later (for example 110$) than it fails in a higher proposal-deal (for example 120$). In another point, the merchants are always competitive, if the buyer fails with one merchant in a lower proposal-deal, it is more possible for it to win in the later rounds of negotiation or get a product from other merchant than it fails in a higher proposal-deal. The "loss" here reflects the possibility to get a product. In this concern, we can suppose the rule:

If it is "urgent": gain=gain+1000.

If it is "not urgent": gain=gain.

And "loss" remains the same.

We explain it by giving an example. For concern "1", if we have the "gain" and "loss" in the client profile as in the table 4.4:

Table 4.4: Example of gain and loss

Price	80	90	100	110	120
Gain	50	100	100	110	120
Loss	40	20	20	40	40

For concern "2", suppose the user has a maximum price, say 120$. From the above data, we move the gain(140) and loss(140) because 140 is greater than 120 and it will not contribute to the acceptable proposal-deal which is less than 120$.

For concern "3", suppose the user chooses the "not urgent". Accordingly, we can have the expectation(proposal-deal) as following:

expectation (80)= probability(80$, accept) *gain(80) -probability(80$,reject)*loss(80)
=0.1*50-0.9*40= - 31

expectation (90)= probability(90$, accept) *gain(90) -probability(90$,reject)*loss(90)
=0.2*100-0.8*20= 4

expectation (100)= probability(100$,accept)*gain(100)-
probability(100$,reject)*loss(100) =0.3*100-0.7*20= 16

expectation (110)= probability(110$, accept) *gain(110)-
probability(110$,reject)*loss(110) =0.3*50-0.7*40= -13

expectation (120) = probability(120$, accept) *gain(120) -
probability(120$,reject)*loss(120) =0.1*50-0.9*40= - 31

For the above example, we summarize them in table 4.5.

Table 4.5: Example of expectation value

Price($)	80($)	90($)	100($)	110($)	120($)
times of previous deal	1	2	3	3	1
Probability(price, accept)	0.1	0.2	0.3	0.3	0.1
probability(price, reject)	0.9	0.8	0.7	0.7	0.9
gain(price)	50	100	100	50	50
loss(price)	40	20	20	40	40
expectation(price)	-31	4	16	-13	-31

4.6.2.2 Least-Squares Parabola

When we have the set of utility, how to give the proposal-deal? We want to construct a curve which can approximate the expectation value for all possible prices, and choose the price which has the maximum expectation to be our proposal-deal.

Which kind of curve is ideal to approximate the prices and their expectations? The linear function y=a+bx is simple and easy to be constructed with the raw data set, but it is difficult to find price x which has the maximum expectation y. The square parabola $y = a + bx + cx^2$ can be constructed from the raw data set and be easy to find the specific price x with the maximum expectation y.

In this section, we introduce the way to construct least square parabola using the raw data set.

The least-squares parabola is an approximation of the given set of data (x_1,y_1), (x_2,y_2), ..., (x_n, y_n), using a second degree curve which has the minimal sum of square deviation from all data points [Suykens *et al.*, 2002][Jiang, 1998] [Farebrother, 1988].

The efunda(engineering fundamentals) is to create an online destination for the engineering community [URL_23]. It gives an efficient way to approximate the least squares parabola [URL_24].

Suppose we have the second degree curve as:

$$y = a + bx + cx^2 \tag{3}$$

If the summary of square deviations should be minimum and the data set number is n, so

$$\Pi = \sum_{i=1}^{n}[y_i - f(x_i)]^2 = \sum_{i=1}^{n}[y_i - (a + bx_i + cx_i^{\,2})]^2 = \min \tag{4}$$

In (4), a, b, and c are unknown coefficients where ($x_{i,}$ y_i) is a set of prices and expectations.

To force the least squares error to be the minimum, the unknown coefficients a, b, and c should have zero first derivatives.

$$\frac{\partial \Pi}{\partial a} = 2\sum_{i=1}^{n}[y_i - (a + bx_i + cx_i^{\,2})] = 0 \tag{5}$$

$$\frac{\partial \Pi}{\partial b} = 2\sum_{i=1}^{n}x_i[y_i - (a + bx_i + cx_i^{\,2})] = 0 \tag{6}$$

$$\frac{\partial \Pi}{\partial c} = 2\sum_{i=1}^{n}x_i^{\,2}[y_i - (a + bx_i + cx_i^{\,2})] = 0 \tag{7}$$

From (5), (6) and (7) we have,

$$\sum_{i=1}^{n} y_i = a\sum_{i=1}^{n} 1 + b\sum_{i=1}^{n} x_i + c\sum_{i=1}^{n} x_i^{\,2} \tag{8}$$

$$\sum_{i=1}^{n} x_i y_i = a\sum_{i=1}^{n} x_i + b\sum_{i=1}^{n} x_i^{\,2} + c\sum x_i^{\,3} \tag{9}$$

$$\sum_{i=1}^{n} x_i^{\,2} y_i = a\sum_{i=1}^{n} x_i^{\,2} + b\sum_{i=1}^{n} x_i^{\,3} + c\sum x_i^{\,4} \tag{10}$$

Now taking account into the above equations, we suppose,

$$\mu_1 = \sum_{i=1}^{n} x_i, \mu_2 = \sum_{i=1}^{n} x_i^{\,2}, \mu_3 = \sum_{i=1}^{n} x_i^{\,3}, \mu_4 = \sum_{i=1}^{n} x_i^{\,4}, \xi_1 = \sum_{i=1}^{n} y_i, \xi_2 = \sum_{i=1}^{n} x_i y_i, \xi_3 = \sum_{i=1}^{n} x_i^{\,2} y_i, \tag{11}$$

then,

$$a=[(\mu_3{}^2-\mu_2\mu_4)(\xi_1\mu_2-\mu_1\xi_2)-(\mu_3\mu_2-\mu_2\xi_3)(\mu_2{}^2-\mu_1\mu_3)]/[(\mu_3\mu_1-\mu_2{}^2)(\mu_3{}^2-\mu_2\mu_4)-(\mu_2{}^2-\mu_1\mu_3)(n\mu_2-\mu_1{}^2)] \quad (12)$$

$$b=[(\xi_1\mu_1-\xi_2 n)(\mu_3\mu_2-\mu_1\mu_4)-(\xi_2\mu_2-\mu_1\xi_3)(\mu_1\mu_2-n\mu_3)]/[(\mu_1{}^2-n\mu_2)(\mu_2\mu_3-\mu_1\mu_4)-(\mu_2{}^2-\mu_1\mu_3)(\mu_1\mu_2-n\mu_3)] \quad (13)$$

$$c=[(\xi_1\mu_1-n\xi_2)(\mu_2{}^2-\mu_1\mu_3)-(\xi_2\mu_2-\xi_3\mu_1)(\mu_1{}^2-n\mu_2)]/[(\mu_1\mu_2-n\mu_3)(\mu_2{}^2-\mu_1\mu_3)-(\mu_2\mu_3-\mu_1\mu_4)(\mu_1{}^2-n\mu_2)] \quad (14)$$

The equations (11), (12), (13) and (14) will be used in formula (3).

4.6.2.3 How to Decide Proposal-Deal

The proposal-deal is the price whose expectation(proposal-deal) value is the maximum. To calculate the proposal-deal, we consider the flexion point in equation (3):

$y=a+bx+cx^2$.

If $y'=b+2cx=0, x=-b/(2c)$, then $x=-b/(2c)$ is the flexion point. For this flexion points, there are two possible cases, e.g., $y''>=0$ or $y''<0$. When $y''=2c<0$, y has the maximum value in equation (3). When $y''=2c>0$, y has the minimum value in equation (3). Therefore, to get the maximum value of y, we force:

$y'=b+2c=0$, if $y''=2c<0$, e.g., $c<0$,

then, $x=-b/2c$, $y=\max Value$ (15)

In this case, the price $x=-b/2c$ makes y the maximum value. Therefore x is the proposal-deal.

According to table 4.2 and equations (11),

μ_1=500, μ_2=51000, μ_3=5300000, μ_4= 560340000,

ξ_1= -139.2, ξ_2= -14006, ξ_3= - 1445580.

According to equations (13), (14) and (15),

b= 20.83,

c= -0.105<0,

x = -b/2c = 99.19048 ≈ 99($)

Because of c<0, the flexion point is the maximum value for $y=a+bx+cx^2$, therefore the proposal-deal should be 99$.

The "proposal-deal=99$" means that with the user's history based data in table 4.2 and the user's preference as "gain" and "loss" in table 4.4, the buyer's CNA bids 99$ using the Learning-Preference Negotiation Strategy. After that, this bid will be used as the user's history data for the later use. If the negotiation can not finish, the CNA will continue to bid using the same way till the negotiation finishes.

In another example, with the same rule, if the user has his history based data as in table 4.3, and his gain and loss as in the table 4.6, following, then his proposal-deal will be 92.27 ≈ 92 ($).

Table 4.6: The gain and loss relative to table 4.3

Price	96	87	84	83	94	88	60	105	121
Gain	10	20	10	30	40	45	42	70	50
Loss	100	80	30	60	60	20	30	20	40

4.6.2.4 More Proposal Deals

After the buyer negotiation agent has given its proposal deal, the seller agent will accept, or abort, or offers again. In the case of accept and abort, the negotiation finishes. But if the seller agent offers again with a newOffer, if the buyer agent thinks that the newOffer is possible to be reduced to meet the buyer's maximum acceptable price, the buyer agent will give a new proposal-deal. The rule is that, it will compare the newOffer with all the data sets used in producing last proposal-deal such as table 4.1, then it deletes the records(e.g., delete the price, its gain, and its loss) whose price is less than the newOffer to form a new set of data, which will produce the new proposal-deal by calculating probability, expectation value by using the above introduced method. If the new data set has only two records, then it takes the average price as the new proposal-deal.

4.7 Closing condition

The close condition is either of the following cases:

- The buyer and seller match,
- The buyer can not increase its proposal deal any more,
- The seller can not decrease its proposal deal any more,
- Time out.

4.8 Discussion

We have tested the Learning-Preference Negotiation Strategy with many cases of use, and we found that the output of the data can be accepted. The Leats-Squeares Parabola can work well because it approximates the data set of (utility$_i$, bid$_i$), i=1, 2,…,n, by which we can find the bid with the maximum utility. Here, "n" is the record number of the history record. From equation (4), we can see that Leats-Squeares Parabola requires to minimize the summary of square deviations, e.g.,

$\Pi = \sum_{i=1}^{n}[y_i - f(x_i)]^2 = \sum_{i=1}^{n}[y_i - (a + bx_i + cx_i^{\ 2})]^2 = \min$, i=1,2,…,n. In this case, if "min" is the same and "n" is greater, the average distance of our result from each history based data record is less, that is Π / n or $\min/2$. In brief, when we have the greater "n" in equation (4), the "bid" has less distance to our history based data and is more accurate. In our system, that means if the buyer buys one thing more often, the agent can give more accurate bid. For example, there are 5 records of the user in table 4.2, then n=5, and the result is 99$ (in section 4.6.2.3); if the "n" is increased, that is to say, the user has more records, e.g., "n" is more than 5, for example, in the table 4.3 and table 4.6, n=9, the result is 92, and we believe it is more accurate. In general, no matter what the "n" is, Learning-Preference Negotiation Strategy is an easy and available way to handle the history based data and user' preference.

From the above explanation, we see that Learning-Preference Negotiation Strategy is supported by the history based data and the user's preference. Our proposal gives an available way to reflect the buyer's budget, his evaluation of the object, how much does he get, how he likes the seller, how urgent he needs the object, and so forth. As we describe in section 3.3, to the different attributes of the negotiation, our strategy gives different utilities to follow the user's intention so that it can maximize the user's utility.

However, it is possible that the attributes that we have given utilities to are not essential to the user or not enough to the user, so the user wants to consider the other attributes which are not included in our consideration, for example, the delivery time, or the payment term, or the warranty period, and so forth. In this case, we can adopt our scheme in those attributes that the user wants. This can be realized because that utility is a term much subjects to the user, and the user can specify his personal utility for this kind of attribute such as the delivery time, or the payment term, or the warranty period, as what he can do for the "gain" or "loss" showed in the above examples. In summary, with the help of utility function, Learning-Preference Negotiation Strategy can adopt to include any attributes to give a bid, and it is the contribution of our system to the e-commerce negotiation.

As we describe in section 2.3.1, Utility Based Agent can have its utilities of an object according to its user's subject of this object, and it can specify the utility for the important attributes while ignore the unimportant attributes, and to take actions which contribute the most to its profit (e.g., maximize its utility). To realize the Learning-Preference Negotiation Strategy in our negotiation system, we create a virtual market, Nego, and a client negotiation agent (CNA) to practice our negotiation strategy.

Chapter 5 Nego Negotiation Implementation

In this chapter we introduce the architecture of Nego—a virtual negotiation market. Our goal is to construct a negotiation market to test our negotiation strategy and methodology. We build a virtual market which can implement negotiation function in a distributed system. We build agents in both the server side and client side to negotiate. Specially, we build a client's mobile agent to travel from client side to server side to implement the negotiation mission.

To understand Nego, it is necessary to have a brief look at Pacha [Mounir, 2002] which Nego continues with. First we will give a brief introduction to Pacha. After that, we will introduce the negotiation model we implement. There are a lot of negotiation protocols, our protocol will focus on the negotiation strategy base on the experience and preference. First of all, we introduce a negotiation model focus on negotiation element, negotiation process flow, negotiation strategy. We focus on the negotiation strategy, which we propose a Learning-Preference Negotiation Strategy. It combines the experience and preference as the underlying learning mechanism to decide the proposal-deal, in which we use the price as the issue of the negotiation.

5.1 Architecture

Our overall purpose is to develop a market to practice our negotiation strategy. An e-commerce market is made up of two sides: server side and client side. In the server side, the sellers want to sell their products; in the client side, the buyers search for what they need and buy. Because the rapidly decreasing of the cost of search in the internet, more and more buyers look for the suppliers in the internet, so it increases the competition among the buyers, which will result in the "winner's curse", a situation of the price of a product is much higher than its value because too many buyers compete for it [Wang *et al.*, 2002]. One cause of the "winner's curse" problem is the communication problems such as network disconnection or the time delay during a negotiation, which block the communication between the sellers and the buyer. To solve the communication problems, we create a mobile client negotiation agent (CNA) on behalf of its buyer, and send it to the server side to implement the negotiation mission. However, the mobile agent has difficulty to implement the complicated calculation needed in the negotiation.

The Remote Method Invocation (RMI) in Java can provide an available way to have the complicated computation done in the client side while the CNA is in the server side. RMI provides the mechanism by which the server and the client communicate and pass information back and forth. The server creates some remote objects, makes reference to them accessible and waiting for the client to invoke the methods in the object. A client gets the reference for the remote objects and invokes the methods in them [Oberg, 2001]. When the CNA needs the bid, CNA can invoke with the parameters the methods in the client side to check the user profile in the client side, and calculate the bid, and get the bid passed by the methods invoked.

The RMI distributed object system must use the multi-thread function in Java to enable multiple clients to access the system at the same time, and it must install the program in the client side if this computer wants to access the system. Its server can keep on working and make solution if the client is offline, when this client is online again, the server can remember it and give the solution

The advantages of the RMI distributed system is that objects are able to reside on any computer within a distributed system, and that programs can be written which enable code on other computers to send messages to them, just as if they were residing on the computer which hosts the message sending code. This feature can increase the reuse of the code, and avoid the disadvantages of application-specific protocol, because application-specific protocol encourages a monolithic form of coding which is difficult to maintain. This can also reduce the large amount of code and maintenance in the different clients because of reuse [Oberg, 2001].

The architecture of Nego is indicated in figure 5.1

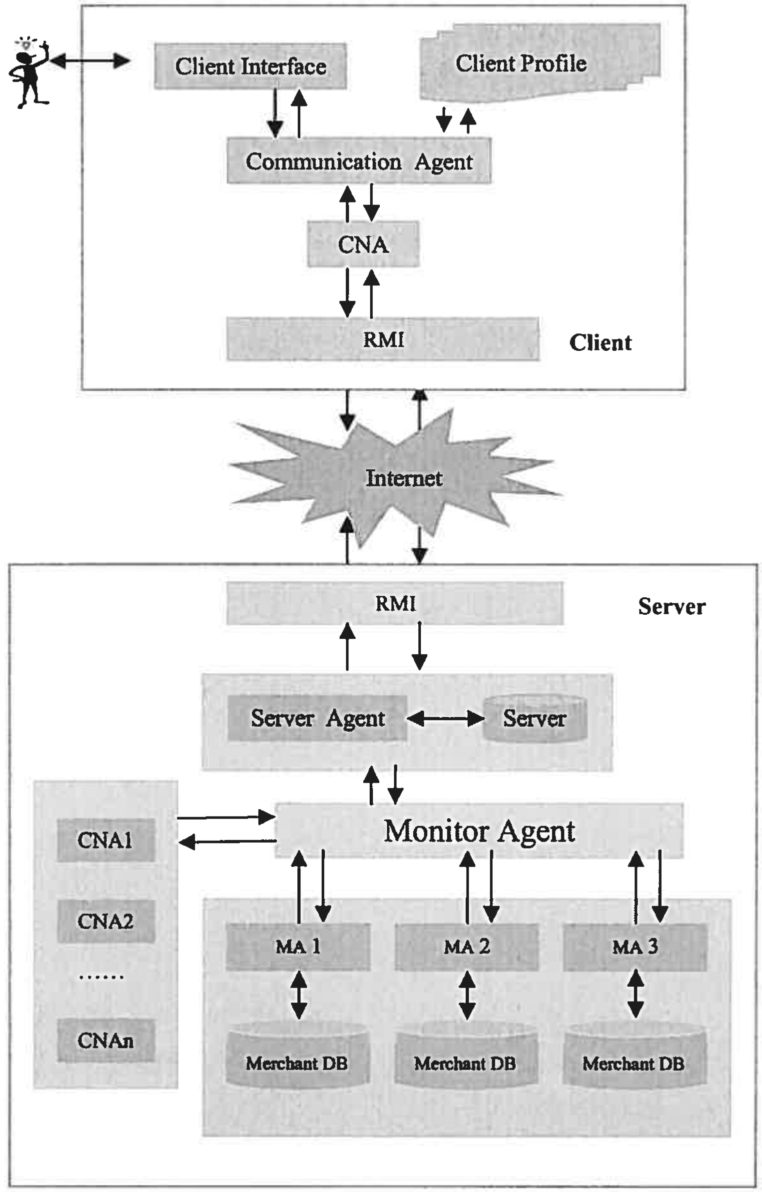


Figure5.1: Nego Architecture (MA : Merchant Agent, CNA : Client Negotiation Agent)

The agent communication language (ACL) is the agent's communication ability with other systems. To implement their missions, agents must be able to exchange information with its user (human being), resources (websites, database, and so forth), and the other agents [Luck *et al.*, 2003]. An agent-based system is requested to be scalable, inter-operable, and re-configurable, and there are two well-established ACLs, Knowledge Query and Manipulation Language (KQML) and Foundation for Intelligent Physical Agent (FIPA) ACL [Luck *et al.*, 2003] [Labrou *et al.*, 1999]. KQML includes three layers: Content layer, expressed in languages such as SOL; Message layer, identifies network protocol and indicates message types such as query, command; Communication layer, lower level communication parameter such as sender and receiver [Labrou *et al.*, 1999] [Finin *et al.*, 1996]. FIPA ACL has the similar concepts and principles with KQML, but different details [URL_37]. There two kinds communication protocol in KQML. The first is the pint-to-point communication protocol, in which agent A is aware of agent B obtaining information X and knows it is appropriate to query B about X; the second is to use the communication facilitator F as an intermediary: when agent A does not know which agent can provide X and asks F, for F ask all the agents, and if agent B replies F that it can provide X, F passes B to A to let A and B communicate [Finin *et al.*, 1996].

Because Nego is based on RMI technology in Java, the agents in Nego communicate by calling the methods. The agents in Nego communicate with the user (human being), the database (resources), and other agents which involve in the negotiation. They implement two kinds of communication protocol. The first kind is as in KQML: point-to-point communication protocols. CNA follows the point-to-point communication protocol when it communicates with the user (CNA calls the user interface or displays the negotiation result to the user) or the communication agent (which sends CNA to the server side and communicates with it when CNA is in the server side) in its client side. This is because CNA awards of the methods of the user interface and the methods of the communication agent. The second kind is similar to, but is different to the other kind communication protocol in KQML: mediator communication protocol. When the CNA (like agent A in KQML) negotiates in the server side, the Monitor agent works as the communication facilitator (like agent F in KQML) to find the seller agent (like agent B in KQML), and passes CNA's bid to seller agent, then the seller agent passes its offer to Monitor agent, and the Monitor agent passes the seller agent's offer to CNA, and so on until the negotiation finishes. Here, the Monitor agent works as a intermediary between these two agents in their communication.

Although KQML and FIPA ACL are full-specified agent communication languages, Nego uses its own way to communicate because of the following two reasons. First, our purpose is to find a good negotiation strategy to improve the negotiation on behalf of the user, and we focus on testing our strategy. Implementing a fully-specified KOML will make our project much larger and without focus. Second, Nego continues Pacha [Mounir, 2002], which we will introduce in section 5.2.1. We follow the Pacha's agent communication way, which is based on the Java RMI technology and uses method call. It follows the point-to-point and the mediator communication protocols. Therefore, Nego has to follow the same agent communication way as Pacha because if we implement KQML in Nego, we must migrate Pacha into KQML before we implement Nego whose workload is out of our range.

5.2 Background

Our project will implement the Negotiation stage, which is an important stage in E-Commerce Consumer Buying Behavior(CBB) model. In the CBB model, Negotiation stage begins after the Need Identification stage, Product Brokering stage, and Merchant Brokering stage. Negotiation is the most complicate, influencing stage. First, it is because that the negotiation involves uncertainty, and it is driven by desire that maximize the utility of the negotiator and there is always inconsistence between these two negotiators. Second, it is influence because that the result of negotiation decides the transaction price, the negotiator may change Product Brokering and Merchant Brokering stage's result to have a better transaction term.

We implement the negotiation function base on the result of project Pacha [Mounir, 2002]. Pacha implements two stages in CBB model: product brokering and merchant brokering. Nego uses the brokering result to begin negotiation. Because of this, in the following section, it is necessary to introduce Pacha and describe the connection between Pacha and Nego.

5.2.1 Introduction to Project Pacha

Pacha [Mounir, 2002] implements the product and merchant brokering stages in the CBB model in E-Commerce. Its purpose is to search the product and merchant, which is needed by the user, from the market.

Pacha is a virtual market which allows the mobile agent to be sent by the user to the server side to accelerate the process of searching specified product in server side. The agent will send back the searching result from the server side when it finishes its mission.

In Pacha, the agent searches what the user specifies, then it uses intelligent techniques such as Case-Based Reasoning (CBR) to filter the searching result. In summary, Pacha can offer the customer on its expertise level solutions given by the CBR technology.

In CBR, there is a case base, in which each case has problem description and solution. The knowledge and reasoning process used by an expert to solve the problem is not recorded, but is implicit in the solution. To solve a current problem, the problem is matched against the cases in the case base, and similar cases are retrieved [URL_29] [URL_30]. CBR is better used to search, for example, a user searches for a product as in Pacha because it matches the condition to find the solution. As to the negotiation, because it concerns the continuous price, we can not enumerate all the cases of the price, so CBR is not usable in Nego.

5.2.2 The Relation between Pacha and Nego

CBB model(refer to section 2.4.2) has six fundamental stages, which is a model of consumer buying behavior. These six stages are: Need Identification, Product Brokering, Merchant Brokering, Negotiation, Purchase and Delivery, Service and Evaluation. Pacha respects the 2nd and 3rd stages of model CBB. Nego continues the function in the forth stage (e.g., Negotiation stage) in CBB model.

Nego will implement the intelligent mobile agent technology in e-commerce price negotiation. The mobile negotiation agent will work independently in the server side to implement the negotiation mission on behalf of its user. After the agent finishes negotiating, it will send back the result of negotiation to the user in client side.

The search result of Pacha is a set of product description, which includes product name, product's merchant information, and product negotiable selling price. The Nego can begin when Pacha finishes its mission.

5.3 Function of Components

Nego is a distributed system which consists of client side and server side. The server in a network is a computer which provides the specific service for another computer, the later is the client. In the view point of location, a distributed system is one in which the computer power is distributed geographically around a number of computers which share the processing load of the system[Ince, 2001] .

In the following sections, we will introduction the functionality for each individual part both in the client side and server side.

5.3.1 Client Side

The "client side" provides service to the user by collecting the input criteria and displaying the output result. The client side acts as intermediary between the user and the server, it communicates with the user, builds up the client profile and responds to the user. There is no negotiation in client side. The negotiation takes place in server side.

In detail, first of all, the client side interacts with the user, and collects the input data from the user and displays the output data to the user. Secondly, the client side creates, sends and receives the mobile negotiation agent. To do that, it creates the mobile negotiation agent, encodes the negotiation strategy in it, sends it to the server side to negotiate and receives the result sent by the mobile negotiation agent from the server side. Thirdly, the client side preserves and updates the client profile, which preserves the user's personal data, and modifies the user's profile. With the above functions, the components in the client side include client interface, client profile, communication agent, client negotiation agent(CNA). We will discuss each of them in the following sections.

- **Client Interface**

The client interface is a graphic interface. It collects the negotiation criteria from the client and displays the result to the client in a suitable manner. It works as an intermediary between the user and our project. In this point, it must work effectively and efficiently.

We provide the appropriate attractive user interface. With the personalized style, our project provides the corresponding respond and the supporting illustration for each negotiation.

- **Client Profile**

The purpose of setting up the client profile is to feature the specific user in order that we can get client information by referencing to client profile.

The user wants to negotiate to get a "good deal" for the product he wants. The client profile is a tool to better serve the user. By referencing to the client profile, we can get information of the user.

According to the utility function mentioned in 4.6.2 Learning-Preference Negotiation Strategy, the client profile must provide the information of probability(price, accept), probability(price, reject). To get probability(price, accept), probability(price, reject), we need to keep all information of the user's previous negotiations such as the price the user accepts before, and how many times he accepts the specific price.

On the other hand, we need to keep the preference of each previous negotiation, which the user assigns "gain" and "loss" value to evaluate the user's preference.

To summary, we keep all of final prices in the client profile, all the numbers of each final price, all gains, and all losses, which the user made in the previous negotiations.

- **Communication Agent**

The function of communication agent is to co-operate the communication between the client side and the server side.

Before the negotiation begins, communication agent asks the user for the negotiation parameter. The negotiation parameter includes: the expect price, the maximum price, the increasing rate, and the prefer negotiation strategy, etc. After it gets the parameter, the communication agent creates and sends the client negotiation agent(CNA) to the server side to negotiate.

Before the CNA finishes its mission in server side, the communication agent monitors the server side to accept the CAN's result sent back at any time.

- **Client Negotiation Agent (CNA)**

The CNA is generated by the communication agent and sent to the server side to negotiate. It is autonomous to work independently from its sender in the server side. It is proactive also to negotiate for the best term on behalf of the user.

After the product brokering stage, CNA is created given a set of product (the product name, its price and its merchant) and its user's preference. The mission of CAN is to negotiate with the merchant to best meet its user's preference.

When the CNA is created, it is set with some parameters to guide it. For example, the user can set the parameters depends on the user preferences:

- ✓ Desired price,
- ✓ Maximum acceptable price,
- ✓ Price increase rate (if he chooses the simple negotiate strategy),
- ✓ How does it desire to buy with each merchant (e.g., how it wants to buy from each merchant),
- ✓ How does it need the product (e.g., if it is urgent or not),
- ✓ Which negotiation strategy to choose.

After CNA is created, it gets the user parameters. Then it is sent to the server side. In the server side, the CNA negotiates with the merchant agent (e.g., MA). The CNA proposes a deal, then the MA accepts or rejects. If the MA rejects, MA responds with a counter proposal-deal. CNA checks it and gives its proposal-deal if it can not accept it, and so on.

CNA can negotiate with multiple MAs in parallel. Whenever there is a possible negotiable merchant, the CNA copies itself to negotiate with the merchant agent (MA) under the negotiation monitor agent's monitor. The best result can be chosen after the CNA finishes each negotiation.

After the negotiation, the CNA sends back the new negotiation result to the communication agent.

5.3.2 Server Side

The sever side is where the negotiation takes place. We suppose that in this virtual market, there are some merchants with certain products available to sell. In this virtual market, the buyer searches the specific product which meets its criteria, then negotiates for the transaction price.

When the CNA comes, the server side hosts the CNA. Then the negotiation takes place in the server side. The components in server side include server agent, monitor agent, merchant agent(NA)s, merchant database and server database. We will discuss each of them in the following sections.

- **Server Agent**

The server agent hosts the CNA. When the CNA arrives, the server agent passes the CNA to the monitor agent, which starts a new negotiation and monitors it.

- **Monitor Agent**

The goal of the monitor agent is to monitor the negotiation. When the server agent passes the CNA to the monitor agent, CNA provides the merchant name list to the monitor agent. The monitor agent contacts the merchant agent for the negotiation. Both MA and CNA pass the negotiation termination criteria to the monitor agent. The monitor agent creates a new negotiation processor sLNegoProcess to start the negotiation. The sLNegoProcess passes the CNA's bid to the MA, if the MA can not accept the bid, it will give a new offer. If the offer and the bid can not match, then the sLNegoProcess passes the offer to the CNA, and so on till the offer and the bid are matched or other termination condition matches.

- **Merchant Agent (MA)**

The merchant agent (MA) negotiates on behalf of the merchant to sell the item. It should be set with some parameters as listed in the following, which MA considers:

- ✓ Desired price,
- ✓ Minimum acceptable price ,
- ✓ The previous transaction data,
- ✓ How does it eager to sell (e.g., if it wants to sell it as soon as possible)

The MA adopts the negotiation strategy mentioned in 4.6.2 Learning-Preference Negotiation Strategy. The MA proposes its deal base on desire price, experience, and preference.

The MA modifies the merchant database to remember all final deal (e.g., the price it offered, the gain and loss values) after the negotiation finishes.

- **Merchant Database**

The merchant database is used to keep the product information and the merchant's profile. As the MA decides the dynamic proposal-deal, the MA needs to refer to the merchant's profile.

After the negotiation finishes, the MA modifies the merchant database with the final deal's information.

- **Server Database**

Server Database is to keep the information of merchants in the server side.

- **Client Negotiation Agent** (CNA)

Client negotiation Agent comes from the client side to stay temporary in the server side to negotiate. Refer to section of Client Negotiation Agent.

5.4 Client Negotiation Agent Working Procedure

When the negotiation begins, a user interface is created to let the user input his parameters. With the brokering result brokeringResult from Pacha, Nego creates a CNA and builds the instruction in it according to the brokeringResult and user's parameters.

After that, Nego sends the CNA to the server side to negotiate with the specific merchants, and the CNA returns the negotiation result when the negotiation finishes. The procedure can be explained as in figure 5.2.

In figure 5.2, after the CNA arrives at the server side, the server agent registers it to the monitor agent, and the monitor agent begins the processor processNego(negoAgent) to process it.

The message processNego(negoAgent) in figure 5.2 is explained in figure 5.3.

The monitor agent gets the user preferred negotiation strategy (either "Simple Negotiation Strategy" or "Learning-Preference Negotiation Strategy" as explained section 4.6.2), and also the merchant list, which the CNA wants to negotiate with, from the CNA. For each merchant, the monitor agent creates a processor sLNegoProcess to lead the negotiation between the CNA and the merchant agent.

The processor sLNegoProcess works as: asks the CNA for the sellerOffer and its buyerBid, then it compares them, and add them to its result. If the sellerOffer and the buyerBid can match, this negotiation finishes. Otherwise, the sLNegoProcess informs the seller agent with the buyerBid and the sellerOffer, the seller agent replies with a new sellerOffer if it can not accept the byerBid, and sLNegoProcess adds the new sellerOffer to its result. The sLNegoProcess checks again to match the new sellerOffer and the buyerBid. If it can not match them, it will inform the CNA with the seller's new sellerOffer and the buyerBid. If the CNA can not accept the new sellerOffer, it will ask the communication agent in the client side to get the proper buyerBid. The communication agent queries the user profile, fixes a buyerBid, and replies to the CNA with a new byuyerBid. The CNA replies the sLNegoProcess with the new buyerBid, and the sLNegoProcess adds the new buyerBid to its result. The sequence of giving sellerOffer and buyerBid is a loop and stops when one of the following conditions appears: the sellerOffer and the buyerBid match; or the seller agent withdraws the negotiation; or the CNA withdraws the negotiation.

When the negotiation between the CNA and a seller agent finishes, the sLNegoProcess inform the finish of this negotiation and pass the result to the CNA, and the CNA adds the result to its subResult. The sLNegoProcess informs the monitor agent of the finish of negotiation.

Each subResult in the CNA is the result of the negotiation between the CNA and one of the merchant in the CNA's merchant list. The negotiations between the CNA and each of the merchants in the CNA's merchant list are in parallel. Each negotiation is processed by a negotiation processor sLNegoProcess, and the CAN's result is a summary of all subResults. The CNA will compare all of the subResults and find the best merchant when the negotiation finishes.

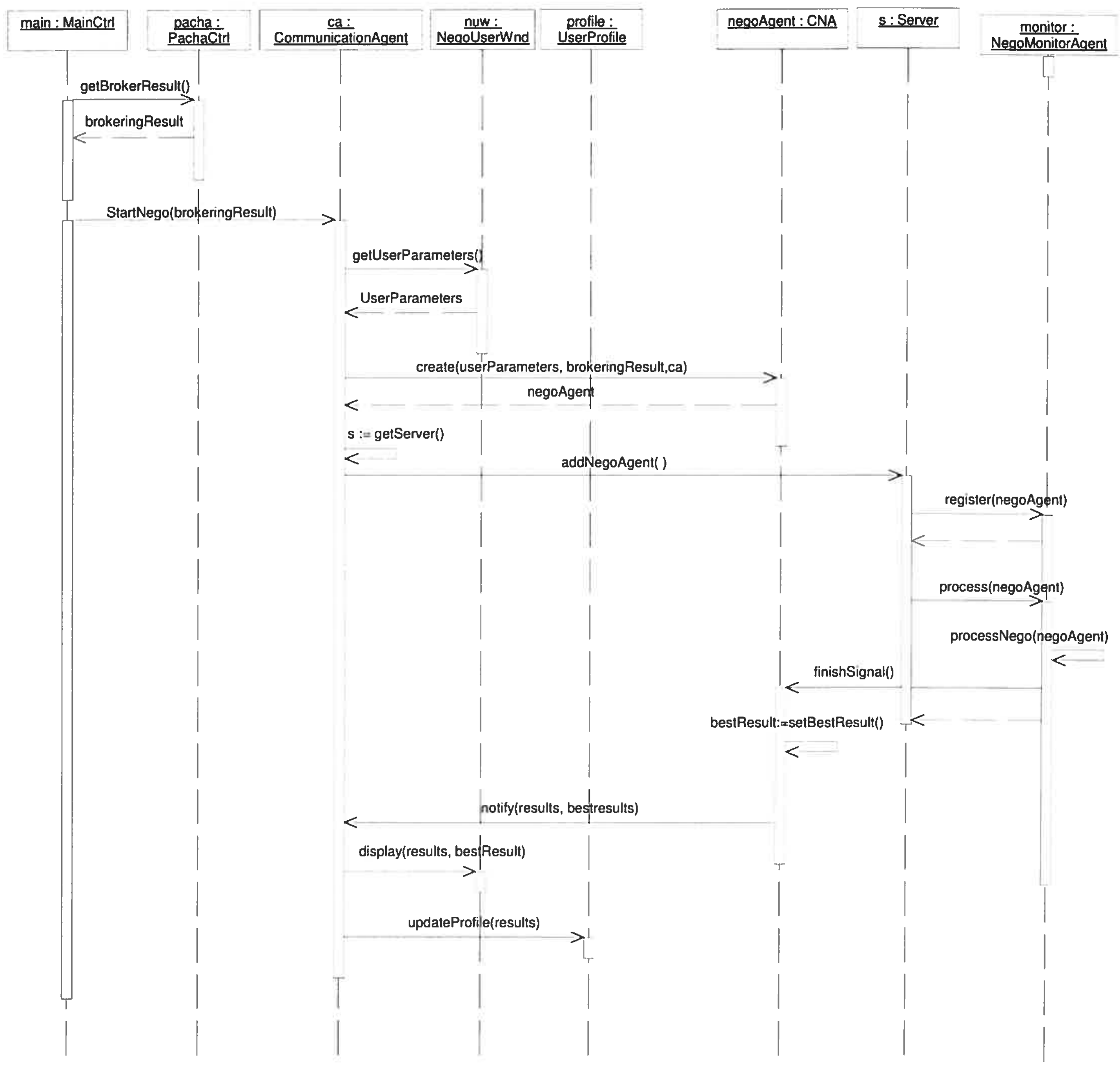


Figure 5.2: Mobile agent (CNA) working sequence diagram

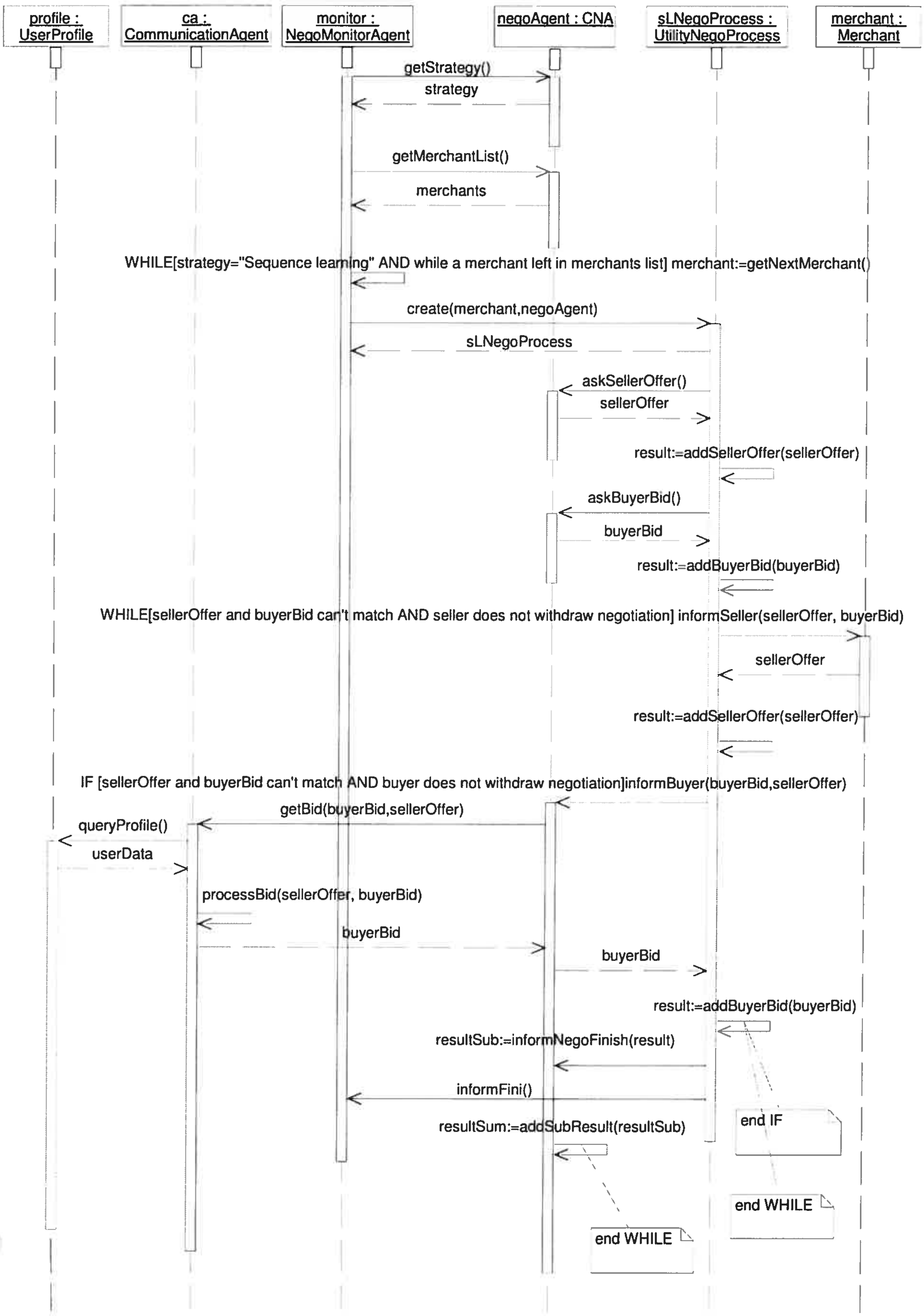


Figure 5.3: Negociation process sequence diagram

5.5 Result Data

We have two kinds of negotiation strategies: Simple Negotiation Strategy and the Learning-Preference Negotiation Strategy. The negotiator can adopt one of these two strategies. In section 5.5.1 and 5.5.2, we try with them and give the result respectively. Suppose the negotiation begins when we have the Pacha's product search result as shown in figure 5.4.

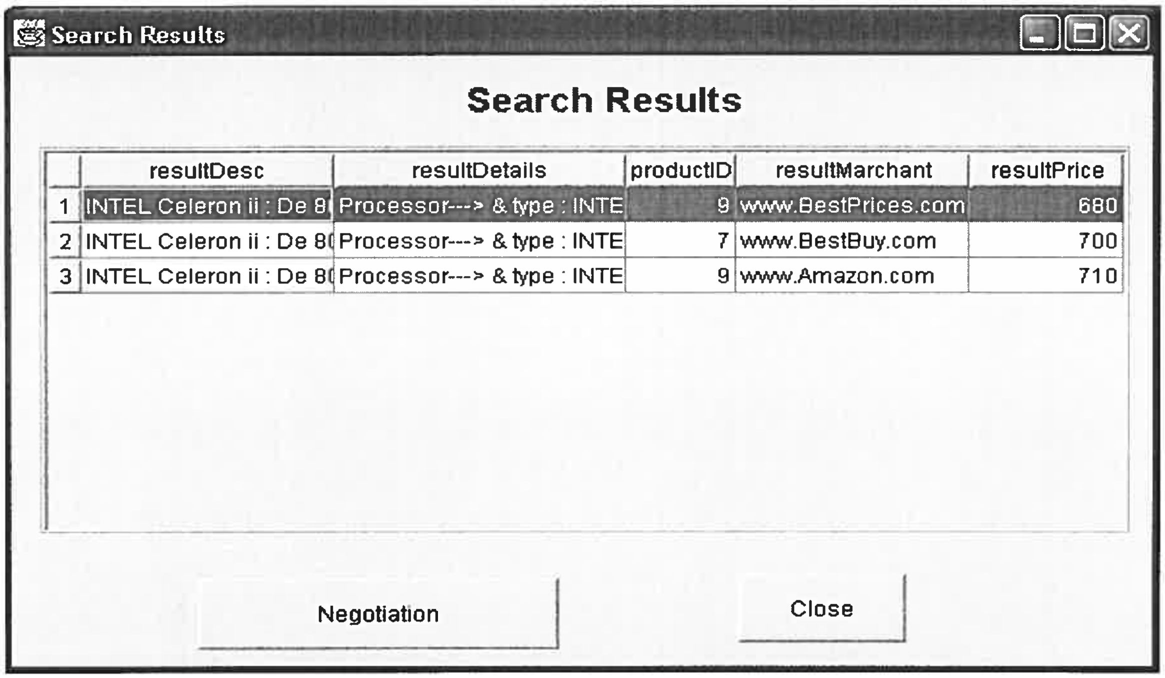


Figure 5.4: Brokering result—products, merchants and their prices to be negotiated

5.5.1 Simple Negotiation Strategy Result

The client considers the maximum price, desired price and the increase rate only. He does not need to specify the other parameters. The client interface can be as figure 5.5.

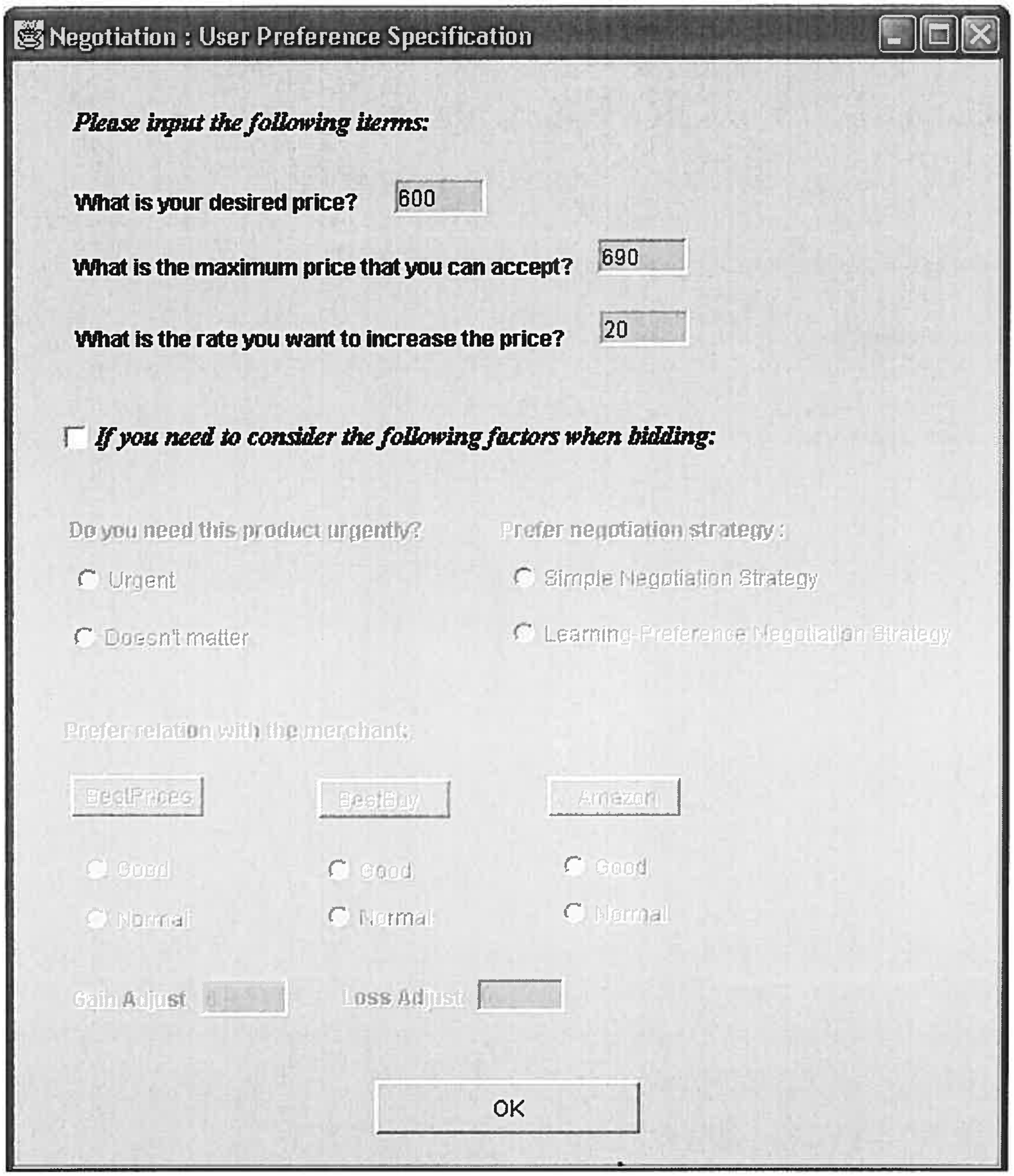


Figure 5.5: Simple Negotiation Strategy's user input interface

Figure 5.5 is the user interface for the negotiation parameters. If the user chooses the Simple Negotiation Strategy, only first three parameters should be input. If the user inputs the desired price, maximum price, and increasing rate as 600.00, 690.00, and 20.00, the figure 5.6 is the running results:

Agent-Negociation-orchidee-132.204.27.192-1072569183155: Result

Negotiation Result

Record 1 of 17

	merchantName	sellerOffer	buyerBid
1	www.BestPrices.com	680	600
2	... ...	670	620
3	final offer:	670	620
4	www.BestBuy.com	700	600
5	... ...	694	620
6	... ...	688	640
7	... ...	682	660
8	... ...	676	676
9	final offer:	676	676
10	www.Amazon.com	710	600
11	... ...	703	620
12	... ...	696	640
13	... ...	689	660
14	... ...	682	680
15	... ...	680	0
16	final offer:	680	680
17	Choose merchant: www.BestBuy.com	676	676

OK

Figure 5.6: Simple Negotiation Strategy's result

The Nego's results have been tested for over three months since we had decided the Simple Negotiation Strategy and the Learning-Preference Negotiation Strategy. During the test period, we tried different cases of use, and each day we test our strategies from 5 to 20 times. We have found that the output can meet our expectation. The following explanation about "Simple Negotiation Strategy" and "Learning-Preference Negotiation Strategy" is based on these random data.

In figure 5.6, the merchant "BestPrices" has minimum offer at 670.00 and maximum offer at 680.00, its decreasing rate is 10.00. The negotiation fails.

The merchant “BestBuy” has minimum offer at 620.00 and maximum offer at 700.00, its decreasing rate is 6.00. The negotiation matches at 676.00.

The merchant “Amazon” has minimum offer at 620.00 and maximum offer at 710.00, its decreasing rate is 7.00. The negotiation matches at 680.00.

The CNA makes final deal with the merchant “BestBuy” at 676.00.

The Simple Negotiation Strategy is not intelligent. It just increases or decreases the same value when negotiating. After the CNA finishes negotiating with all the merchants, the agent takes the best offer.

5.5.2 Learning-Preference Negotiation Strategy Result

As mentioned in section 4.6 Negotiation Strategy, we have two strategies: “Simple Negotiation Strategy” and “Learning-Preference Negotiation Strategy”. The default setting is “Simple Increase”, but if the user chooses the “Sequencial Learning” strategy, the client considers not only the maximum price and desired price but also other factors such as indicated in figure 5.7.

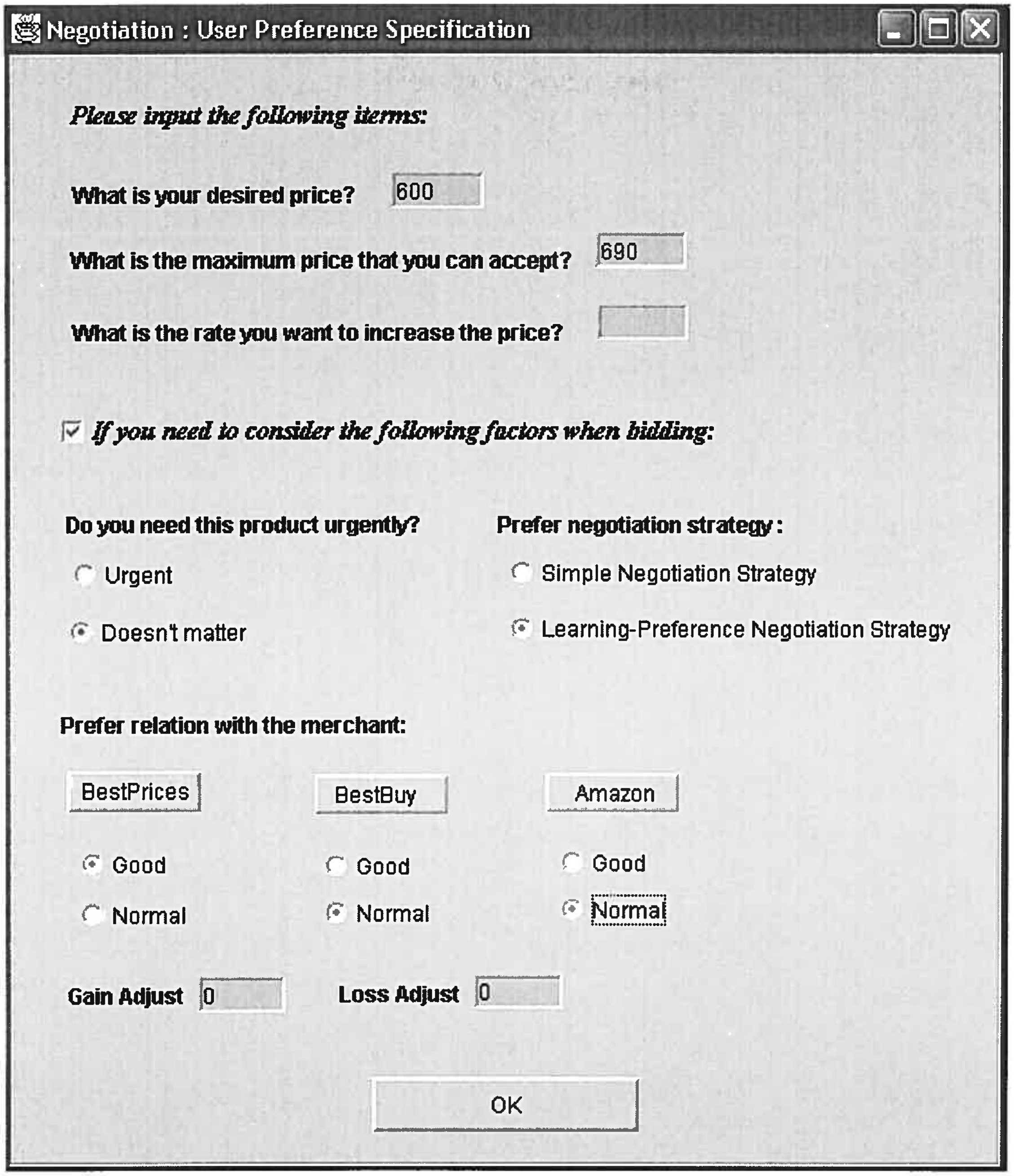


Figure 5.7: Learning-Preference Negotiation Strategy's user interface (product not urgent)

Learning-Preference Negotiation Strategy will need the following user information:

- **Maximum Price**:

The Maximum price is the maximum buyer acceptable price. Please refer to section 4.6.2.1 Probability, Preference and Utility, concern "2".

- **Desired Price**:

The Desired Price is the minimum price which the user can accept. Please refer to section 4.6.2.1 Probability, Preference and Utility, concern "2".

- **How the user needs the product**

The need for the product has two states: "urgent" or "doesn't matter". If choose "urgent", the increase of proposal-deal can be higher than choose "doesn't matter". In this case, the client eager to make a deal, he would provide a higher bid. If "doesn't matter", the proposal-deal is normal. Please refer to section 4.6.2.1 Probability, Preference and Utility, concern "3".

- **Negotiation strategy**

We have Simple-Negotiation-Strategy and Learning-Preference Negotiation Strategies. The user can choose one of them. For the detail of our strategies, please refer to section 4.6.1 Simple Negotiation Strategy, and section 4.6.2 Learning-Preference Negotiation Strategy.

- **Negotiation protocol**

The negotiation starts when the Client Negotiation Agent starts to give its first bid, if the bid can match the seller's offer, the negotiation succeed; if the bid can not match the offer, the seller agent replies with an new offer, if the new offer can not match, the buyer agent give the next bid, so it continues till the bid and the offer match, or the buyer withdraws the negotiation, or the seller withdraws the negotiation, or the negotiation time finishes.

- **The resource of the data**

The data comes from two resources: first, the history data comes from the user's profile including "price", "time", "gain", and "loss", which are kept in the user's profile every time the negotiation finishes; second, the current user's preference comes from the user input in the user's interface before the negotiation including "if it is urgent" and "relation with the user". In general, the data used in our experiments are random data.

- **Relation with the merchant**

The relation with the merchant has two states: "good" or "normal". If the relation is "good", it will be more possible for the buyer to accept a higher increase offer than in "normal". Refer to section 4.6.2.1 Probability, Preference and Utility, concern "1".

Suppose we have the user input as in figure 5.7, and we have the user previous record as in table 5.1:

Table 5.1: Client profile example

Price	700	695	690	685	680	678	675	670	665	660	650	620
Gain	10	20	10	30	40	42	45	70	70	65	40	10
Loss	100	80	30	60	60	30	20	20	40	50	80	100
Times	2	3	5	4	4	1	4	5	1	4	1	1

Then, we have the result as in figure 5.8.

Agent-Negociation-orchidee-132.204.27.192-1074292361746: Result

Negotiation Result

Record 1 of 11

	merchantName	sellerOffer	buyerBid
1	www.BestPrices.com	680	600
2	... ...	672,3	665,679
3	final offer:	672,3	665,679
4	www.BestBuy.com	700	600
5	... ...	677,462	665,679
6	... ...	668,948	668,948
7	final offer:	668,948	668,948
8	www.Amazon.com	710	600
9	... ...	697,081	665,679
10	final offer:	697,081	665,679
11	Choose merchant: www.BestBuy.co	668,948	668,948

OK

Figure 5.8: Learning-Preference Negotiation Strategy's result (product not urgent)

The merchant "BestPrices" offers 680.00, the CNA bids 600.00. The BestPrice offers 672.3. The buyer agent bids 665.679. "BestPrices" can't decrease the offer any more. The negotiation stops.

The merchant "BestBuy" offers 700.00, the buyer CNA bids 600.00. The "BestBuy offers 677.462 again. The buyer agent bids 665.679. "BestBuy" offers 668.948 again. The buyer agent agrees 668.948. It's a possible deal.

The merchant "Amazon" offers 710.00, the CNA bids 600.00. The merchant offers 697.081. The buyer agent bids 665.679. "Amazon" can't decrease its offer. The negotiation stops.

After the above three sub-results, the CNA compares the possible deal and decides that the final deal will be made with "BestBuy" by 668.948.

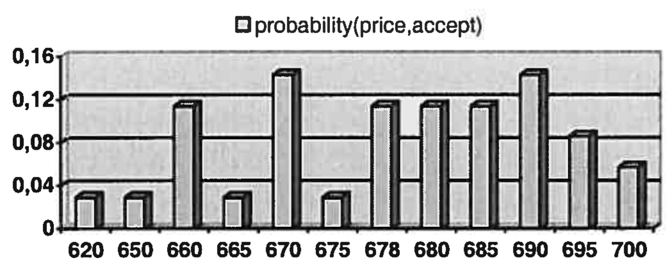
Let's see why the CNA agrees the bid 668.948.

The probability(price, accept) and probability(price, reject) deducted from client profile (refer to table5.1) are shown in table 5.2 and figure 5.9.

Table 5.2: Probability in client profile

Price	700	695	690	685	680	678
probability(price,accept)	0.057	0.086	0.143	0.114	0.114	0.114
probability (price, reject)	0.943	0.914	0.857	0.886	0.886	0.886

Price	675	670	665	660	650	620
probability(price, accept)	0.029	0.143	0.029	0.114	0.029	0.029
probability(price, reject)	0.971	0.857	0.971	0.886	0.971	0.971



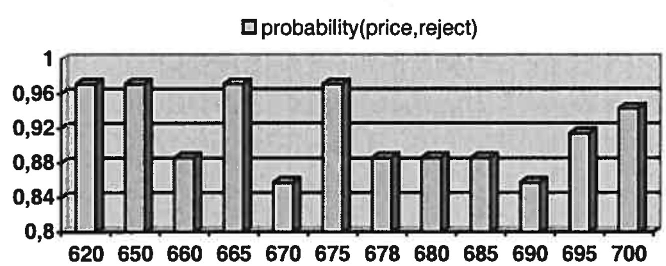


Figure 5.9: Probability(price, accept) and probability(price, reject) distribution

From the client profile(refer to table 5.1), the "gain" and "loss" are as in table 5.3:

Table 5.3: Gain and loss in the client profile

Price	700	695	690	680	680	678	675	670	665	660	650	620
Gain	10	20	10	30	40	42	45	70	70	65	40	10
Loss	100	80	30	60	60	30	30	20	40	50	80	60

Figure 5.10 shows the "gain" and "loss" of this example.

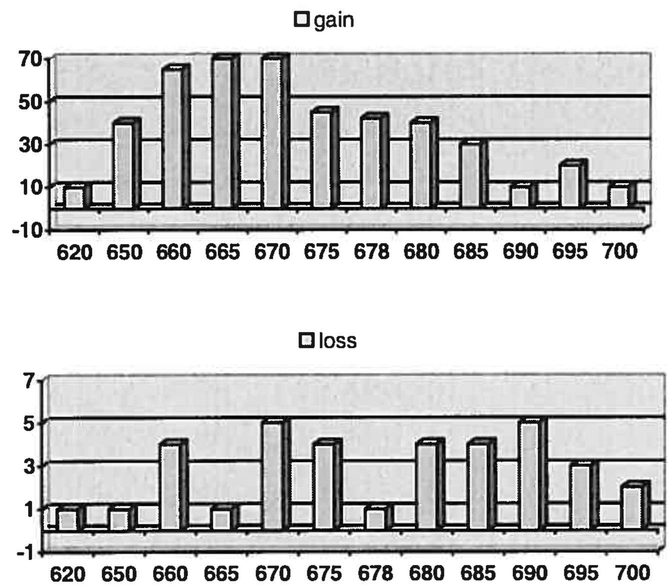


Figure 5.10: Gain and loss in client profile for negotiation

According to equation(2) and table 5.2, table 5.3, the expectations are as in table 5.4 and figure 5.15, the expectation(665.679) is the maximum value, therefore the bid will be 665.679.

Table 5.4: Expectation Example

Price	700	695	690	685	680	678	675	670
Expectation	-93.71	-71.43	-24.29	-49.71	-48.57	-27.94	-12.57	-7.14

Price	665	660	650	620
Expectation	-36.86	-36.86	-76.57	-96.86

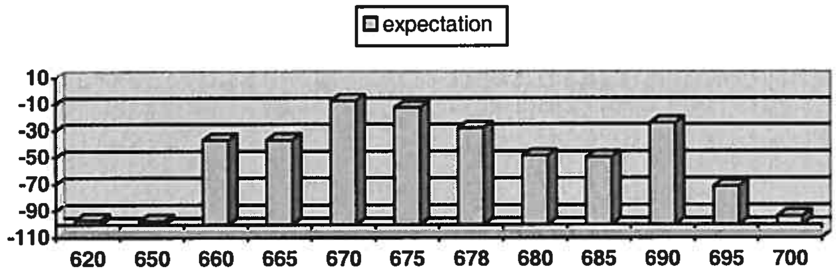


Figure 5.11: Expectation value for negotiation

According to figure 5.8, "BestBuy" offers 677.462, but the buyer bids 665.679. The "BestBuy" checks that it is possible to decrease the offer. Then, it gives again the offer at 668.948. The CNA finds that this offer is less than its maximum acceptable offer which is 690, so it figures out its new bid and finds its new bid is higher than the offer. Then it accepts the new offer at 668.948.

Let's see how the CNA has the bid 665.679. According to table 5.4, quotation (3), (11), (13) and (14),

$$\mu_1 = 8{,}068.00, \mu_2 = 5{,}429{,}684.00, \mu_3 \approx 3{,}657{,}566{,}752, \mu_4 \approx 2{,}466{,}064{,}872{,}356$$
$$\xi_1 \approx -582.5142857142856, \xi_2 \approx -389{,}958.1142857143, \xi_3 \approx -261{,}487{,}434.34285712$$
$$b \approx 49.64076846068029, c \approx -0.03728583575962078$$

Therefore, the proposal-deal should be:

$$-b/(2*c) = -49.64076846068029/(2*(-0.03728583575962078)) \approx 665.679$$

After the CNA gives the bid 665.679, the "BestBuy" offers 668.948. How does the CNA get its new bid, how much is the new bid? To give a bid when the merchant offers 668.948, the CNA changes its profile data as in table 5.1 into a new set in this way: it deletes the records which "price" is less than the current buyer bid, and add the record of the newest bid which is (price: 665.679, gain:50, loss:40), to form table 5.5. Compared with table 5.1, the record with price of 665.00, 660.00, 650.00 and 620.00 are removed, and the previous bid(665.679) is added.

Table 5.5: Data which will form new bid after previous bid (base on table 5.1 with offer 668.948)

Price	700	695	690	685	680	678	675	670	665.679
Gain	10	20	10	30	40	42	45	70	50
Loss	100	80	30	60	60	30	20	20	40
Times	2	3	5	4	4	1	4	5	1

The "665.679" is the previous bid as showed in figure 5.11. How do we have it(price: 665.679, gain:50, loss:40)? We suppose that the "gain" and "loss" values are assigned by what the user's relation with the merchant in interface(figure 5.7) as:

- "good" relation
  gain=100,
  loss=20.
- "normal" relation
  gain=50,
  loss=40.

Because in figure 5.7, the user chooses the relation with "BestBuy" as "normal", the "gain" is 50 and "loss" is 40. With table 5.5 and equation (2), we have the expectation as in table 5.6.

Table 5.6: The expectation values related to records in table 5.5

Price	700	695	690	685	680	678	675	670
Expectations	-92.41	-69.65	-23.10	-47.58	-46.20	-27.51	-11.03	-4.48

Price	665.679
Expectations	-55.86

According to equations (11), (13), (14), (15) and table 5.6, the buyer agent has:

$\mu_1 \approx 6{,}138.67863438426, \mu_2 \approx 4{,}188{,}087.0442756927, \mu_3 \approx 2{,}858{,}018{,}998.3708105,$

$\mu_4 \approx 1{,}950{,}847{,}415{,}354.6$

$\xi_1 \approx -385.8965517241379, \xi_2 \approx -246{,}728.2806479709, \xi_3 \approx -169{,}661{,}235.84051695$

$b \approx 124.42111510703405, c \approx -0.09244449543035006$

Therefore, the proposal-deal should be:

$-b/(2*c) = -124.42111510703405/(2*(-0.09244449543035006) \approx 672.950$

According to figure 5.8, the "BestBuy" offers 668.948 which is less than 672.950, the CNA accepts 668.948. Therefore they reach an agreement at 668.948.

If the user chooses that he needs the product urgently, as shown in the figure 5.13. Let's see how the agent gives the bid. What does the "urgent" means here? The "urgent" means the user wants to get the product even the price is a little higher. We increase the gain of the client profile, which means getting the product is a little achievement. We increase the gain by 1000. It can be increased by 10, or 10000, this depends on the designer. For example, we increase by 1000, the gain will be changed as in table 5.7:

Table 5.7: The changed gain by the user's preference

Price	700	695	690	685	680	678	675	670	665	600	650
Gain	1010	1020	1010	1030	1040	1042	1045	1070	1070	1065	1040

Price	620
Gain	1010

And the other parameters such as "loss" and "probabilitiy"s remain the same. And the expectation(price) in table 5.4 are as changed in table 5.8 and figure 5.12. The result is as in figure 5.14.

Table 5.8 Expectation example

Price	700	695	690	685	680	678	675
expectation	-36.57	14.29	118.57	64.57	65.71	0.629	101.71

Price	670	665	660	650	620
expectation	135.71	-8.29	77.43	-48	-68.29

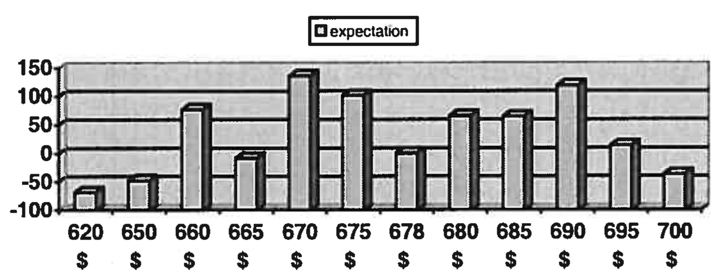


Figure 5.12: The expectation value in negotiation (the bid is 672.950)

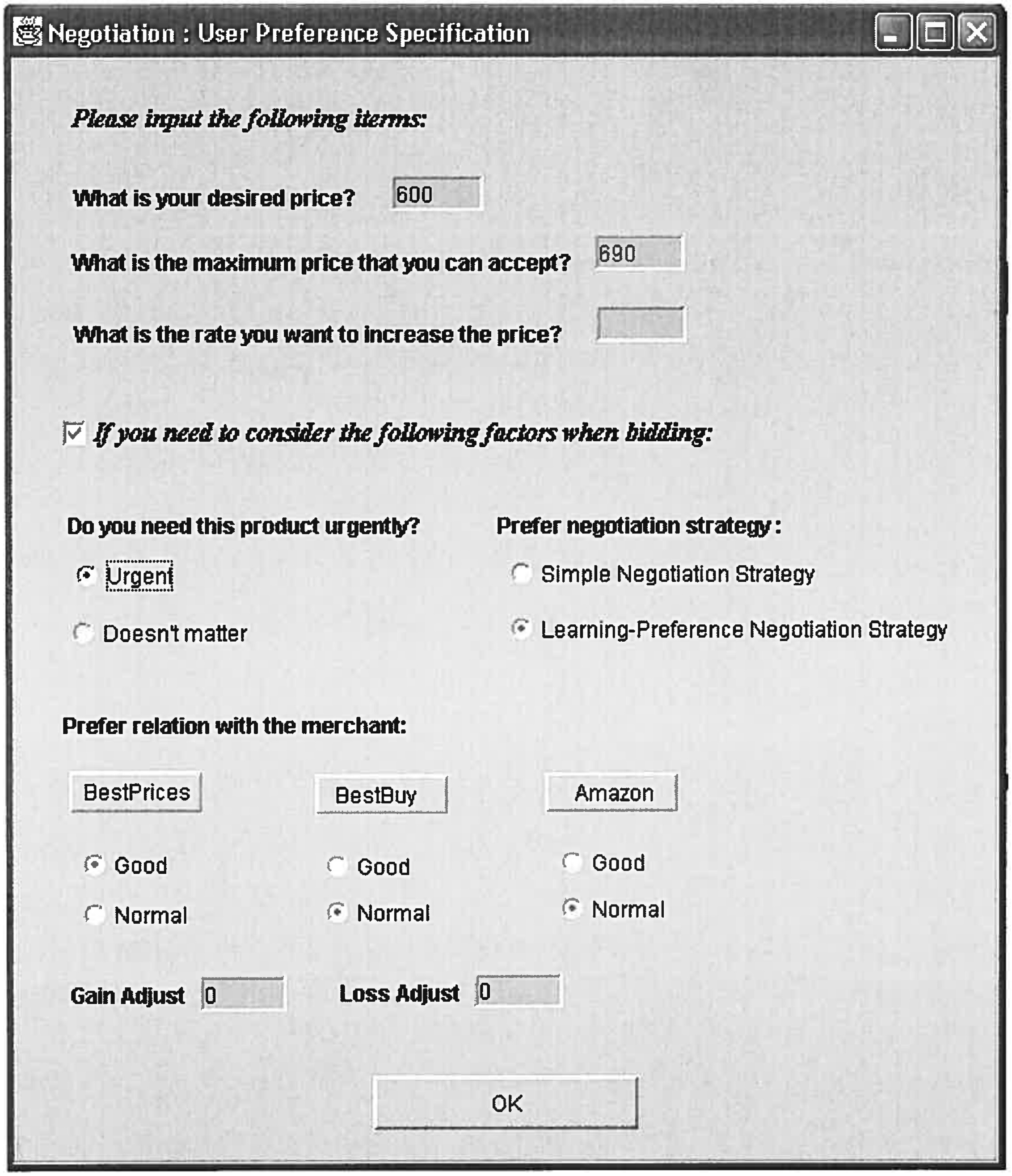
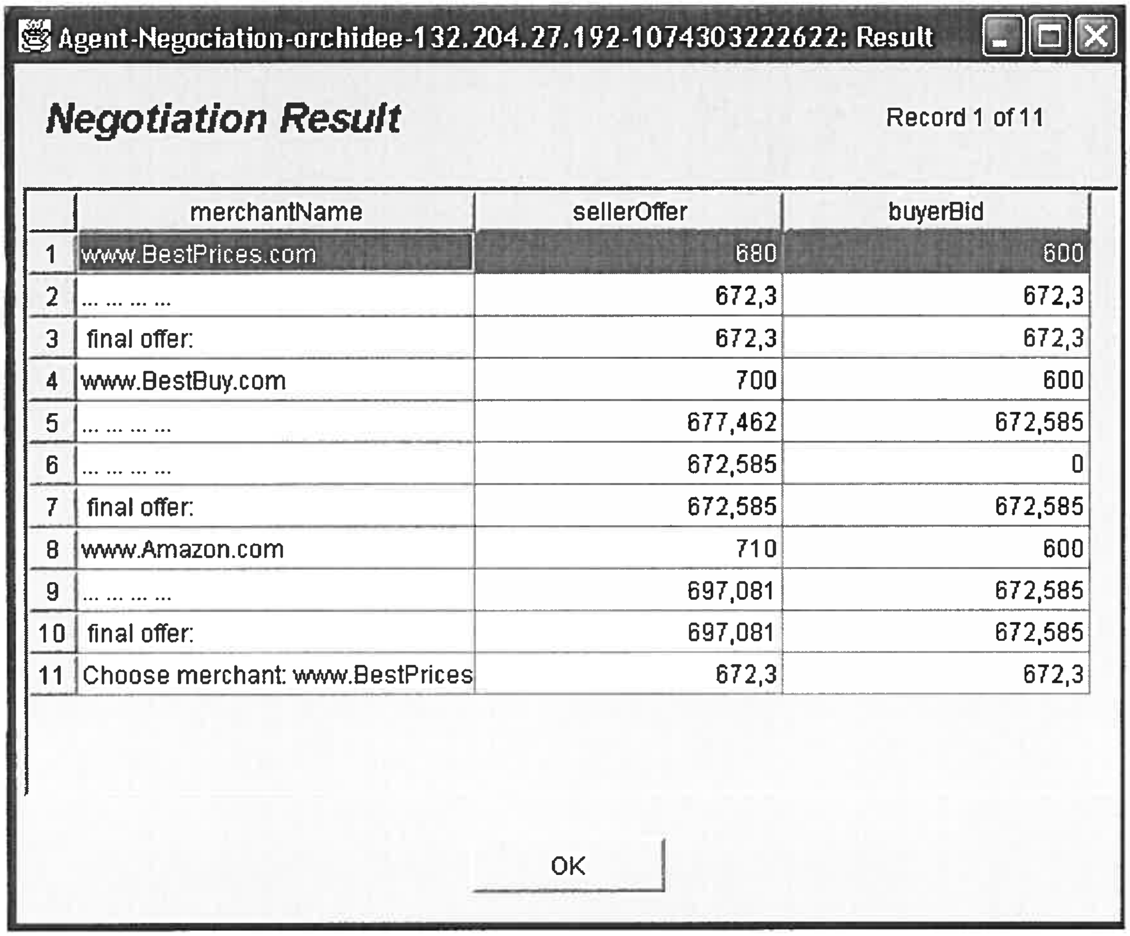


Figure 5.13: Learning-Preference Negotiation Strategy's user interface (product is urgent)



	merchantName	sellerOffer	buyerBid
1	www.BestPrices.com	680	600
2	... ...	672,3	672,3
3	final offer:	672,3	672,3
4	www.BestBuy.com	700	600
5	... ...	677,462	672,585
6	... ...	672,585	0
7	final offer:	672,585	672,585
8	www.Amazon.com	710	600
9	... ...	697,081	672,585
10	final offer:	697,081	672,585
11	Choose merchant: www.BestPrices	672,3	672,3

Figure 5.14: Learning-Preference Strategy Negotiation result (product is urgent)

The testing data used here come from history-based data and user's preference (gain, and loss) are random data. From the Price and the Gain in table 5.7, and the Loss in table 5.3 and user's Expectation in table 5.8, we have the result in figure 5.14.

The merchant "BestPrices" gives the offer 672.300. The CNA prefers the bid 672.585, but finds that 672.585 is higher than 672.300, it agrees the offer by 672.300.

The merchant "BestBuy" gives the offer 677.462. The buyer agent bids 672.585. "BestBuy" checks its profile to find that it can decrease to 672.585, it agrees.

The merchant "Amazon" gives the offer 697.081. The buyer agent bids 672.585. "Amazon" checks its profile to find that it can not decrease its offer. The negotiation stops.

Let's see how the CNA fixes its proposal-deal 672.585 after the "BestPrice" offers 672.30. According to table 5.8, we have:

$$\mu_1 = 8068.0, \mu_2 = 5429684.0, \mu_3 \approx 3{,}657{,}566{,}752, \mu_4 \approx 2{,}466{,}064{,}872{,}356,$$
$$\xi_1 \approx 417.4857142857143, \xi_2 \approx 287{,}127.60000000003, \xi_3 \approx 197{,}206{,}394.22857141$$
$$b \approx 73.14368100927106, c \approx -0.05437502694253778$$

Therefore, the proposal deal should be:

$$-b/(2*c) = -73.14368100927106/(2*(-0.05437502694253778)) \approx 672.585$$

The negotiation between the CNA and the merchant "BestPrices", the negotiation between the CNA and the merchant "BestBuy", and the negotiation between the CNA and the merchant "Amazom" are processed in parallel. That means the CNA negotiates with all of then in the same time. When all of them finish, the buyer CNA verifies and choose the best offer. In this case, it chooses the "BestPrices" with the deal 672.3.

5.6 Comparison with other Systems

Though online auction exhibits attractive features for retail negotiation such as fairness and openness, it suffers from the problem such as reversed consumer-buyer relation and low performance [Wang *et al.*, 2002]. Particularly, the low performance exhibits such as the buyer can only give the fix increment in his negotiation in the e-market, instead of the flexible bid. The examples are Kasbah [Chavez and Maes, 1996], eBay and Yahoo auctions [URL_03] [URL_04]. Kasbah is the earliest virtual market place for negotiation, and it plays fundamental role in the development of the online negotiation research. With the development of the intelligent agent and the e-commerce, more and more research have been put on improving the performance of the online auction, among them the Bazaar is an outstanding one, and it focuses on how to give a bid based on the learning.

The mobile agent seems can make the internet commerce more functional; however, it can not perform the negotiation well due to its limited computational ability in the Internet. Although there is much research focusing in the mobile agent's function in the negotiation, the limitation of computational ability in mobile agent is still a major concern in the online negotiation. [Wang *et al.*, 2002] proposes a "Mobile Agent with Security Agent-Mediated Auction-like Negotiation Protocol", which implemented in the system called Security Agent-Mediated Auction-like Negotiation (SAMAN). SAMAN gives a way to negotiate with the sellers by the opinion of finding the most potential sellers based on the learning.

Nego's purpose is to find the seller which can meet its user's requirement and maximize its user's utility after negotiation. Based on this opinion, we look for the most similar and functional systems: Kasbah, Bazaar and SAMAN to compare with Nego to find the advantages and the disadvantages. We focus on the negotiation strategy and agent intelligent when comparing. The following is the compare with Kasbah, Bazaar, and SAMAN.

5.6.1 Kasbah

Kasbah is an earlier negotiation model. Its strategies are not intelligent, it adopts three kinds of strategies such as linear, quadratic, or exponential function to increase its bid when negotiating. Its strategy can be pre-computed [Chavez and Maes, 1996]. Table shows an example of Kasbah linear function, and we can see that the buyer's bid increase with the time passes.

Table 4.1: Example for Kasbah strategy

Time	21:20:15	21:20:34	21:20:53	21:20:54	21:21:12	21:21:31	21:21:50
$Offer_{seller}$	100	96	91	91	87	83	79
Bid_{buyer}	70	73	75	75	78	81	79

It follows the equation: $Bid_{buyer} = 3/19 * Time_{second} + 66.63$, which means the bid of the buyer in a specific time can be calculate in advance.

Nego's strategy is more learning comparing with Kasbah negotiation Strategies. Nego's strategy is used dynamically base on the previous data and the user's preference, and it is learning. First, the bid is produced by the least-squares parabola using the user's series utility sets ($utility_i$, bid_i), i=1, 2,..., n, in which the user's maximum utility decided the bid. The example of the utility sets is the expectation as in the following table 5.8:

Table 5.8 Expectation example

Price	700	695	690	685	680	678	675
expectation	-36.57	14.29	118.57	64.57	65.71	0.629	101.71

price	670	665	660	650	620
expectation	135.71	-8.29	77.43	-48	-68.29

Second, the $utility_i$ can be calculated as the expectation in equation(2):

$$expectation = probability(price, accept) * gain(price) - probability(price, reject) * loss(price)$$

where probability(price, accept)=1-probability(price, reject) (2)

In equation(2), the probability(price, accept) and probability(price, reject) are the user's history based data, and can be calculated from table 5.8, which are kept in the user's profile.

Third, in equation(2), the gain(price) and loss(price) are the user's preference and they are made up as the following,

gain(price)=gain(in client profile)+gain(urgent)+adjust value

loss(price)=loss(in client profile)+adjust value

In the above two equations, the "gain" comes from the user's profile "gain(in client profile)", and the user's preference as "urgent or not" in the user's interface "gain(urgent)", and the user's adjust value in the user's interface "adjust value". If in the user's interface, the user chooses "not urgent", "gain(urgent)"=0. The "adjust value" input in the user's interface is an additional way to adjust the gain by the user. On the other hand, the "loss" comes from the user's profile "loss(in client profile)", and the use's adjust value input by the user in the user's interface. The example is as in the following table 5.3:

Table 5.3: Gain and loss in the client profile

Price	700	695	690	680	680	678	675	670	665	660	650	620
Gain	10	20	10	30	40	42	45	70	70	65	40	10
Loss	100	80	30	60	60	30	30	20	40	50	80	60

Finally, the agent's new bid is kept and used in the next round bid, and the final bid will be kept to be used in the next time negotiation, so the agent in later negotiation can learn from the previous negotiation and make its bid from them.

From the above description, we can see that Learning is an important aspect of Nego. In this point, Nego is experiment based and personal preference based.

5.6.2 Bazaar [Zeng and Sycara, 1998]:

Bazaar is a sequential decision making negotiation learning model (e.g., the current decision based on the previous data set, each time when it gets the result, it will update its knowledge base for the later use). It believes that the learning can give a bid more close to the seller's preserve price (e.g., minimum selling price). It uses the Bayesian network to update the knowledge and belief that each agent has about the environment and other agents, and to produce the estimation of the seller's preserve price.

Equation (16) is the buyer's Bayesian rule, in which "e" is the seller's offer, H_i is a set of buyer's hypotheses of seller's reserved price (e.g., the minimum selling price), i=1,2,3,.... P is the buyer's hypotheses probability.

$$P(H_i \mid e) = \frac{P(H_i)P(e \mid H_i)}{\sum_{k=1}^{n} P(e \mid H_K)P(H_k)} \qquad (16)$$

An example is given in the following. Suppose i=1,2, e_1 denotes the event that the supplier asks $117 for the good, e_2 denotes the event that the supplier asks $152.1, H_1 denotes the seller's reserved price is $100, H_2 denotes the seller's reserved price is $130, $P(e_2|H_2)$=0.95, $P(e_1|H_1)$=0.95, $P(e_1|H_2)$=0.75, $P(H_1)$=0.5, $P(H_2)$=0.5, then,

$$P(H_1 \mid e_1) = \frac{P(H_1)P(e_1 \mid H_1)}{P(H_1)P(e_1 \mid H_1) + P(H_2)P(e_1 \mid H_2)} = \frac{0.5*0.95}{0.5*0.95+0.5*0.75} = 55.9\%$$

$$P(H_2 \mid e_1) = \frac{P(H_2)P(e_1 \mid H_2)}{P(H_2)P(e_1 \mid H_1) + P(H_2)P(e_1 \mid H_2)} = \frac{0.5*0.75}{0.5*0.95+0.5*0.75} = 44.1\%$$

Therefore, the bid =55.9%* 100+44.1%* 130 =$113.23

Bazaar updates its buyer's knowledge every time after the seller offers. After each round of negotiation, the buyer gets closer to the seller's preserved price.

The similarity of the Bazaar and Nego is that both of them use the sequential decision making negotiation learning model. For example, Bazaar uses the H_1, H_2, e_1, e_1, $P(e_2|H_2)$, $P(e_1|H_1)$, $P(e_1|H_2)$, $P(H_1)$, $P(H_2)$ to have the new bid. However, the way Nego calculates the proposal-deal is different. Nego uses (2) to form the user's expectation, and simulate the function, then find the bid with the maximum user's utility.

$$\begin{aligned} \text{expectation(price)} = {} & \text{probability(price, accept)} * \text{gain(price)} \\ & - \text{probability(price, reject)} * \text{loss(price)} \end{aligned} \qquad (2)$$

The probability, gain, and loss in equation (2) comes from the user's profile such as in the table 5.8 and table 5.3, and the user's input from the user's interface. The more Nego considers than Bazaar's is the user's preference such as the gain and loss in table 5.3, and in this way, Nego can give the different utility to the different situation such as the good is urgent (utility is higher), which seller is better (the better one has higher utility) in order to make the decision in order to maximize its utility. Bazaar is just learning, but not user's preference based.

5.6.3 Mobile Agent with Security Agent-Mediated Auction-like Negotiation Protocol (SAMAN)

[Wang *et al.*, 2002] presents a mobile agent negotiation system. Suppose there are sellers which hosts in its personal server. The buyer puts the sellers' list into its VKB(virtual knowledge base) and sorts/clusters all the sellers to a fuzzy set by its belief that the seller will offer a cheap product such as bread, milk, biscuit, etc. According to VKB, the buyer selects the most potential sellers and sends its negotiation agent to each of them. The negotiation agent goes through all the potential sellers. In each seller, if it finds it is possible to have a get a better offer (the agent has a current acceptable offer (CO) which is the current minimum one and the decrease rate (DR), if there is an other offer which is less than the CO-DR, it is a better offer and it becomes CO), it will copy itself in this server to negotiate there and continue to go to next seller. After the copies of agent have their result, they compare with each other to find the best result.

We can see that, SAMAN uses the pre-computed strategy to give the bid. However, it uses the first offer, and compares it with other offers in order to find an offer less than CO-DR, in which DR is a fix value. If the current seller can not give a lower offer, it ignores it and continues to go to next one till go through al the sellers. Nevertheless, it does not use the user's preference such as gain and loss in its decrease rate. In this point, Nego is better than it. Moreover, it uses experience when it decides the fuzzy set of the most potential sellers. It uses the following rule to decide the most potential sellers,

$$PR_{new} = \frac{PR * TRC + PR'}{TRC + 1}$$

PR is the price rank in the profile, PR_{new} is the new PR after a current round of negotiation, and TRC is the round number, PR' is the PR in the last negotiation. The agent ranges the seller by the new PR value after each negotiation, and the seller with the lowest PR is visited first in the next negotiation. In this point of view, SAMAN is experience based. Although Nego is experience based also, it uses the experience in different way to negotiate.

Besides, Nego is similar to SAMAN in the following way:

- Both of them send the negotiation agent to the server side to negotiate, and the agent comes back with the best result,
- The negotiation agent does the negotiation in parallel.

The difference of SAMAN from Nego is that it adopts fuzzy set to cluster the sellers into categories, so the negotiation can take place in the most potential sellers first; the agents in Nego does not order the sellers. The advantage of Nego is in its negotiation strategy: its negotiation strategy considers the user's preference such as gain and loss as a very important factor, and the user can adjusting its proposal-deal by adjust its preference value; SAMAN does not consider the user's preference.

After compare with three similar and available negotiation systems, we find that Nego is good in considering the user's preference such as the different utilities given to different sellers, and different utilities given to the different situations of the good's urgent level. The user's different preference is adopted in the Nego's strategy in order to made the decisions maximized the user's utility and consistence with the user's specific situation. On the other hand, none of the three systems, Kasbah, Bazaar, and SAMAN, considers the user's preference in the negotiation. This advantage of Nego is because of the mobility of the CNA: CNA can let the agent in the client side to do the complicate computation in the client side in order to solve the limitation of computation problem of the mobile agent in the server side. In summary, with the help of the mobile agent, differing the user's utility in different situation and maximizing the user's utility are the advantage of Nego.

5.7 Discussion

The technical support of Nego is Java RMI (Remote Method Invocation). Java RMI is a mechanism that allows one to invoke a method on an object that exists in another address space, and it is object-oriented, and it uses the Internet Inter-ORB Protocol (IIOP) of CORBA as the underlying protocol for RMI communication [URL_36]. The advantage of RMI is that the method in the client side can invoke the method in the server side. When the client negotiation agent (CNA) works in the server, it can inquiry the data or call the methods in the client side. For example, the equations (11), (12), (13), and (14) used in least-squares parabola involves many data (bid_i, $utility_i$, $gain_i$, $loss_i$), $i=1,2,\ldots, n$, so it is better to have a method to perform the least-squares parabola in the client side and have a result based on least-squares parabola. When the CNA needs the bid produced by least-squares parabola and it can call the method in the client side to do it and pass to it. RMI can realize this. By RMI, the bid is passed from the client side to the server side to the CNA when it asks. In this way, it can avoid passing original data such as (bid_i, $utility_i$, $gain_i$, $loss_i$), $i=1,2,\ldots, n$ between the client side and the server side, which can avoid the problems during data transporting in the network. However, RMI's disadvantage is to require the code to be installed in the client side and the server side before it works.

Except the RMI technology, the multi-thread in Java also makes its contribution. For example, when there are several sellers negotiating with a buyer, the buyer can give different bid based on the buyer's intention to each specific seller, which is realized by the multi-thread technology. In this way, the CNA can arrange time to bid with the different seller and avoid seller's long time waiting.

Compared with the Simple Negotiation Strategy, Learning-Preference Negotiation Strategy is based on previous experience and the user preference. On one hand, it supposes that the current decision is made on history based; on the other hand, it considers the user's preference by assigning each previous result with its heuristic value "gain" and "loss", and assigns the values of intend-to-buy to different merchant. Therefore, it is more reasonable and personal.

Compared with other e-commerce negotiation systems, our system considers the user utility in the negotiation while most of the e-commerce negotiation systems do not. For example, our system reflects the buyer's budget, his evaluation of the object, how much does he get, how he likes the seller, how urgent he needs the object, and so forth. Furthermore, in our platform, we can add any attributes into our consideration. There is no utility in most of the other negotiation systems. We compare with three similar and available systems. The first is the Kasbah. Kasbah only simply increases the bid with the time varies. Its strategies do not give different user the different bid based on a set of attributes such as "gain", "loss", "prefer seller", "if it is urgent to him", and so forth. The second is Bazaar. Bazaar considers the user's history based, and it has its bid based the history based bids and their probabilities. However, it does not consider the use's preference such as "gain", "loss", "prefer seller", "if it is urgent to him", and so forth. Therefore, it is history based but not user preference based. The third is Mobile Agent with Security Agent-Mediated Auction-like Negotiation Protocol (SAMAN) [Wang *et al.*, 2002]. It uses the history based data and fuzzy logic to category the sellers, and choose the most potential seller to negotiate first, but it does not consider the user's preference.

From the above comparison, we can find that the advantage of our system than the other systems is the consideration of the user's preference in utility based function. By this way, our negotiation agent can give different sellers different bids and maximize its user's profits.

Chapter 6 Conclusion and Future Work

Nego is an e-commerce negotiation application which applies agent technology into the e-commerce negotiation. Its purpose is to find an efficient way to make negotiation in e-commerce more competitive and autonomous, and to solve one of the problems of the e-commerce negotiation: how to fix the proposal-deal to succeed in negotiating. Nego adopts a methodology based on the user experience to fix the proposal-deal in a dynamic manner.

6.1 Nego Agent Evaluation

The agent in Nego is autonomous, proactive, learning, reactive, communicative, mobile and collaborative.

- **Learning**

  The CNA learns from the previous experience and it modifies its profile to help the next negotiation. In the equation (2), The CNA uses the probabilities, gains and loss from the user's profile, which has examples as table 4.1 and table 4.2. Because the user's profile is updated by the result of each negotiation, the CNA learns from the history.
- **Communicative:**

  There are two meaning of communications in Nego: the communication within a single negotiation and agents communicate between the client side and server side. The Communication Agent in section 5.3.1 communicates with the server side and the CNA to send and receive the agent and its messages, and it is a n example of communication. The Monitor Agent in section 5.3.2 communicates with the CNA and the Merchant Agent (seller agent) to control the negotiation, and it is also a kind of example.

  The first kind of communication takes place between the communication agent and CNA and monitor agent, monitor agent and merchant agent, server agent and monitor agent, etc. The communication agent creates, sends the CNA and receives its negotiation result. The server agent receives the CNA and passes it to the monitor agent. The CNA and the monitor agent communicate to pass the buyerBid to the monitor agent and get the sellerOffer from merchant agents. The monitor agent communicates with the merchant agents to pass the buyerBid to the merchant agents and gets the sellOffer from them. When the negotiation finishes, the monitor agent informs the CNA, merchant agent and server agent.

The second kind of communication takes place between the client and server. The client side sends the CNA to the server side to communicate with server side's agents. During the time it stays in the server side, the CNA can get information from the client side, pass information forward and back between the server side and the client side.

- **Agent Mobility**

  The mobile negotiation agent travels to the server side to work. In "Figure 5.2: Mobile agent (CNA) working sequence diagram" in section 5.4 Client Negotiation Agent Working Procedure, the CNA is a mobile agent, and it is sent to the server side to negotiate with the seller agent. While it works in the server side, it invokes the method in the client side to let the method do the complicate computation for it and pass the result to it.Compares with the non-mobile agent, the advantage of the agent's mobility is,

  - Passes message between the client and server,
  - Utilizes the resources in the server side,
  - Carries the user's requirement and implements user's mission independently in the server side,
  - Utilizes the resources in the client side while the agent is in the server side, for example, to query the client profile when it is in the server side.
  - Uses the client side to implement the complicate computation to reduce the workload of the agent

6.2 Negotiation Strategy Analyze

The Simple Negotiation Strategy is not learning. The reason we adopt the Simple Negotiation Strategy is to implement negotiation in a remote environment, to make use of an agent's proactive, communicative, and competitive in the negotiation. And we finally find that the Learning-Preference Negotiation Strategy is better when we compare the Simple Negotiation Strategy and the Learning-Preference Negotiation Strategy.

The Learning-Preference Negotiation Strategy has the following properties:

- The agent learns from previous transaction records. It interpolates these data to have least-squares parabola $y(x)=a+bx+cx^2$, where "x" is price and "y(x)" is user expectation value in price "x". Here "a" and "b" are dynamically determined. Maybe least-squares parabola is not the best curve to find the price relative to the user's expectation, but it is an easy way to find that price.
- The client profile is updated after succeeding in negotiating to keep the final-deal, gain and loss for later use. Thus, the new final-deal is dynamic and it is sequence learning.

6.3 The Contribution of Nego

The utility based agent is the development trend in the agent technology now. Its advantage is to adopt the human preference in its action. Our contribution is to construct a utility based agent to simulate human preference in negotiation, using previous negotiation experience as decision base. And we also use mobile agent technology to make the negotiation autonomous. Besides, we propose a method to fix the bid of the buyer. All in all, the contribution of Nego can be summarized as the following:

1. Adopt the least-square parabola to create an expectation-based parabola in order to find the specific proposal-deal which has the maximum expectation value. In this way, we solve the problem of dealing with abstract concepts such as experience, final-deals and user preferences into mathematical way.

2. Adapt a dynamic negotiation strategy that considers the user's preference and previous experience. Thus can support the buyer's decision reasonably, with human aspect, and dynamically. By this way, the user's decision is more reasonable and personal. And he can improve his decision eventually by accumulating experience.

3. Make the advantage of the mobile agent to solve the complex computation problem in the e-commerce. Mobile agent can be a problem solver traveling between the client side and server side, it passes messages between them, and it implements its mission in the server side, and it makes use of resource in the client side. In this way, the CNA can transfer the complicate computation to the client side to improve its work.

4. Find a way to reduce the negotiation time. For a single mobile CNA, multiple negotiations with different sellers take place at the same time, therefore save the negotiation time. Because the CNA can involve in each negotiation with different seller, each negotiation is independent.

6.4 Problem and Future Work

While Nego has the above mentioned achievement, it also has the following problems. The future work can focus on the following problem solving:

1. The least-square parabola is a practical way to simulate the utility value, it requests that the number of the discrete pares (expectation, price) is the more the better. The error of it is O(n), "n" refers to the number of the discrete pares. If "n" is small, for example, say 100, the result is not as good as the result of "n" is 1000. Therefore, it has limitation in the number of the dataset.

2. E-commerce negotiation concerns multiple terms such as price, payment term, delivery and so on. Since the price is the most important term in the transaction terms, Nego explores the negotiation on price. Because Nego's utility evaluation method can be used in any emotional evaluation, we can adopt the way of evaluation the user preference in Nego to the negotiations on other terms.

3. Nego practices the negotiation strategy concerning the user's preference, experience and the counter-negotiator's feedback. The future work can expand the negotiation agent's concern to other competitors. For example, the agent can also consider its competitor's situation, such as the other CNA's threaten.

We are hoping that Nego can be more autonomous and functional after the above mentioned efforts.

References:

[Adam *et al.,* 1998] Adam, N. R., Dogramaci, O., Gangopadhyay, A. and Yesha, Y.: *Electronic Commerce - Technical, Business, and Legal Issues*. Prentice Hall PTR, New Jersey. 1998.

[Anthony and Jennings, 2003] Anthony, P. and Jennings, N. R.: *Developing a Bidding Agent for Multiple Heterogeneous Auctions*. ACM Transactions on Internet Technology, Vol. 3(3), pages 185-217. ACM Press. August, 2003.

[Beam and Segev, 1996] Beam, C. and Segev, A.: *The Rise of Electronic Commerce: Contributions from Three Factors*. The Fisher Center for Management and Information Technology, Haas School of Business, University of California, Berkeley. 1996.
Available: http://groups.haas.berkeley.edu/citm/publications/publications/papers/wp-1015 .pdf, Sept 16, 2004.

[Berger, 1988] Berger, James. O.: *Statistical Decision Theory and Bayesian Analysis*. Springer-Verlag, New York. pages 47. 1988.

[Chajewska *et al.,* 2001] Chajewska, U., Koller, D. and Ormoneit, D.: *Learning an Agent's Utility Function by Observing Behaviour*. 18th International Conference on Machine Learning, pages 35-42. 2001.

[Chavez and Maes, 1996] Chavez, A. and Maes, P.: *Kasbah: An Agent Marketplace for Buying and Selling Goods*. Proceedings of the First international Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology. London, UK. pages75-90. April, 1996.

[Dholakia *et al.,* 2002] Dholakia, N., Fritz, W., Dholakia, R. R. and Mundorf, N.: *Global E-Commerce and Online Marketing: Watching the Evolution*. Quorum Books. 2002.

[Erkol, 1998] Erkol, E.: *A Multi-Agent Extension of Negoplan and Its Application to a Business Strategy Game*. A thesis from the school of Graduate Studies and Research in Ottawa-Carleton Institute for Computer Science. 1998.

[Farebrother, 1988] Farebrother, R. W.: *Linear Least Squares Computations*. Marcel Dekker, Inc. New York. 1988.

[Finin *et al.*, 1996] Fini, T., Labrou, Y. and Mayfield, J.: *KQML as an agent communication language*. Invited chapter in Jeff Bradshaw (Ed.), Software Agents, MIT Press, Cambridge, (1996).

[Franklin and Graesser, 1996] Franklin, S. and Graesser, A.: *Is it an Agent, or just a Program?: A Taxonomy for Autonomous Agents.* Proceedings of the Third International Workshop on Agent Theories, Architecture, and Languages. Springer-Verlag. New York. pages 21-35. 1996.

[Guttman *et al.*, 1998] Guttman, R. H., Moukas, A. G., and Maes, P.: *Agent-mediated Electronic Commerce: A Survey.* Knowledge Engineering Review, June, pages 143-152, 1998.

[Hayes-Roth, 1995] Hayes-Roth, B.: *An Architecture for Adaptive Intelligent Systems.* Artificial Intelligence Vol. 72(12), pages 329-365. 1995.

[He *et al.,* 2003] He, M., Jennings, N. R., and Leung, H. F.: *On Agent-Mediated Electronic Commerce*. IEEE Transactions on Knowledge and Data Engineering, Vol. 15(4), pages 985-1003, 2003.

[He and Jennings, 2002] He, M. and Jennings, N.R.: *SouthamptonTAC : Designing a Successful Trading Agent.* Fifteenth European Conference on Artificial Intelligence, Lyon, France. 2002.

[He and Jennings, 2003] He, M. and Jennings, N. R.: *SouthamptonTAC: An Adaptive Autonomous Trading Agent.* ACM Transactions on Internet Technology, Vol. 3(3), pages 218-235. August, 2003.

[Ince, 2001] Ince, D.: *Developing Distributed and E-commerce Applications.* Addison Wesley, Boston. 2001.

[Jiang, 1998] Jiang, B.: *The Least-Squares Finite Element Method: Theory and Applications in Comutational Fluid.* Springer-Verlag. Springer-Verlag.1998.

[Jennings and Wooldridge] Jennings, N. R. and Wooldridge, M. J.: *Applications of Intelligent Agents.* In Agent Technology: Foundations, Applications, and Markets (eds. N. R. Jennings and M. Wooldridge) 3-28, 1998.

[Kumar and Feldman, 1998] Kumar, M. and Feldman, S. I.: *Business Negotiation on the Internet.* Proceeding INET'98 (International Networking 1998), Geneva, Switzerland. 1998.

[Labrou *et al.*, 1999] Labrou, Y., Finin, T. and Peng, Y.: *Agent Communication Languages : The Current Landscape*. IEEE Intelligent Systems, 14(2) :45-52, March-April, 1999. http://ieeexplore.ieee.org/xpl/abs_free.jsp?arNumber=757631.

[Lomuscio *et al.,* 2003] Lomuscio, A. R., Wooldridge, M. and Jennings, N. R.: *A Classification Scheme for Negotiation in Electronic Commerce.* International Journals of Group Decision and Negotiation, Vol. 12(1): pages 31-56. 2003.

[Luck *et al.*, 2003] Luck, M., McBurney, P., and Priest, C.: *Agent Technology: Enabling next Generation Computing.* AgentLink. Part 2.2.3 Agent Communication Languages (p.18). Jan., 2003. http://www.agentlink.org/roadmap/roadmap.pdf.

[Ma, 2001] Ma, Y.: *XML Finder: an Intelligent Agent based on CBR for E-commerce.* Department d'informatique et de recherche operationnelle, Faculté des Arts et des Sciences, Universite de Montreal. 2001.

[Maes, 1995] Maes, P.: *Artificial Life Meets Entertainment: Life like Autonomous Agents.* Communications of the ACM, Vol. 38(11), 108-114. 1995.

[McAfee and McMillan, 1987] McAfee, R. P. and McMillan, J.: *Auctions and Binding.* Journal of Economic Literature, Vol. 25(2), pages 699-738.1987.

[Mounir, 2002] Youssef, M. : *Pacha: Marche Virtuel pour Agents Mobiles.* Department d'informatique et de recherche operationnelle, Faculté des Arts et des Sciences, Universite de Montreal. 2002.

[Nwana, 1996] Nwana, H. S.: *Software Agents: An Overview.* Knowledge Engineering Review. Vol. 11(3), pages 205-244. 1996.

[Oberg, 2001] Oberg, R.: *Mastering RMI—Developing Enterprise Applications in Java and EJB.* John Wiley & Sons, Inc. 2001.

[Parsons and Wooldridge, 2002] Parsons, S. and Wooldridge, M.: *Game Theory and Decision Theory in Multi-Agent Systems.* Autonomous Agents and Multi-Agent Systems, Vol. 5(3), pages 243-254. 2002.

[Pivk and Gams, 2000] Pivk, A. and Gams, M.: *Intelligent Agents in E-commerce.* Electrotechnical Review, Vol. 67(5) pages 251-260. 2000.

[Russell and Norvig, 1995] Rueesll, S. and Norvig, P.: *Artificial Intelligence: A Modern Approach.* Prentice Hall. 1995.

[Scerri *et al.,* 2000] Scerri, P., Y'dren, J. and Reed, N.: *Layered Specification of Intelligent Agents.* The Sixth Pacific Rim International Conference on Artificial Intelligence, Melbourne, Australia. 2000.

[Segovia *et al,* 2002] Segovia, J., Szczepaniak, P. S., Niedzwiedzinski, M. and Dimitrov, V. S.: *E-Commerce and Intelligent Methods.* Springer-Verlag, New York. 2002.

[Shang *et al.,* 2001] Shang, Y., Shi, H. and Chen, S. S.: *An Intelligent Distributed Environment for Active Learning.* Journal of Educational Resources in Computing, Vol. 1(2), pages1-17. 2001.

[Sierra *et al.,* 1997] Sierra, C., Jennings, N. R., Noriega, P. and Parsons, S.: *A Framework for Argumentation Based Negotiation.* Proceeding Fourth International Workshop on Agent Theories, Architectures and Languages (ATAL-97), pages 167-182. 1997.

[Simon *et al.*, 1986] Simon, H. A. and Associates: *Decision Making and Problem Solving.* National Academy of Sciences. Published by National Academy Press, Washington, DC. 1986.

[Suykens *et al.*, 2002]Suykens, J. A. K., Gestel, T. V., Brabanter, J. D., Moor, B. D. and Vandewalle, J. *Least Squares Support Vector Machines..* World Scientific. 2002.

[Turban *et al.*, 2002] Turban, E., King, D., Lee, J., Warkentin, M. and Chung, H. M.: *Electronic Commerce: A Managerial perspective.* Prentice Hall, 2nd edition. 2002.

[Wang *et al.*, 2002] Wang, X., Yi, X., Krishnan, R., Siew, C. K. and Khosla, P. K.: *Mobile Agent Based Auctionlike Negotiation in Internet Retail Commerce.* Book Chapter. Segovia J., Szczepaniak P.S., Niedzwiedzinski M.(Eds.) E-Commerce and Intelligent Methods, Studies in Fuzziness and Soft Computing, Vol. 105, pages: 342-362. ISBN 3-7908-1499-7. Springer-Verlag. New York. 2002.

[Wooldridge and Jennings, 1994] Wooldridge, M. and Jennings, N. R.: *Agent Theories, Architectures, and Languages: a Survey.* Proceeding ECAI-Workshop on Agent Theories, Architectures and Languages (eds. M.J. Wooldridge and N.R. Jennings), Amsterdam, The Netherlands, pages 1-32. 1994.

[Wurman *et al.*, 1998] Wurman, P. R., Wellman, M. P. and Walsh, W. E.: *The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents.* Proceedings of the Second International Conference on Autonomous Agents, pages 301-308. 1998.

[Zeng and Sycara, 1998] Zeng, D. and Sycara, K.: *Bayesian Learning in Negotiation.* International Journal of Human Computer Systems, Vol. 48, pages 125-141. 1998.

[Zhang *et al.*, 2000] Zhang, X., Podorozhny, R. and Lesser, V.: *Cooperative, MultiStep Negotiation over a Multi-Dimensional Utility Function.* Proceedings of the IASTED International Conference, Artificial Intelligence and Soft Computing, pages 136-142. Banff, Canada. IASTED/ACTA Press, 2000.

Useful URL:

[URL_01] [http://ecommerce.about.com]

[URL_02] [http://www.nsf.gov/od/lpa/news/publicat/nsf0050/start.htm]

[URL_03] [http://pages.ebay.ca/education/bidding_1.html]

[URL_04] [http://auctions.yahoo.com/phtml/auc/us/tour/1-3-auto.html]

[URL_05] [http://ai.eecs.umich.edu/cogarch0/common/prop/mea.html]

[URL_06] [http://www.agentwebranking.com/description.htm]

[URL_07] [http://www.idc.com]

[URL_08] [http://www.copernic.com/en/index.html]

[URL_09] [http://searchenginewatch.com]

[URL_10] [http://www.amazon.com]

[URL_11] [http://www.salesmountain.com]

[URL_12] [http://www.activebuyersguide.com]

[URL_13] [http://www.cdrom-guide.com/bargainfinder.htm]

[URL_14] [http://www.jango.com]

[URL_15] [http://www.auctionBot.com]

[URL_16] [http://www.onsale.com]

[URL_17] [http://pages.ebay.com/liveauctions]

[URL_19] [http://www.ge.com]

[URL_20] [http://www.boeing.com]

[URL_21] [http://pages.ebay.com/community/aboutebay/overview/index.html]

[URL_22] [http://www.ebay.com]

[URL_23] [http://www.efunda.com/about/about.cfm]

[URL_24] [http://www.efunda.com/math/leastsquares/lstsqr2dcurve.cfm]

[URL_25] [http://www.searchengineworld.com/spiders/index.htm]

[URL_26] [http://w3.informatik.gu.se/~dixi/agent/class.htm]

[URL_27] [http://www.usna.edu/Users/polisci/purkitt/6]

[URL_28] [http://www.u.arizona.edu/~dusana/psych325presession/notes/CH12.ppt]

[URL_29] [http://www.aiai.ed.ac.uk/links/cbr.html]

[URL_30] [http://www.wordiq.com/definition/Case-based_reasoning]

[URL_31] [http://www.cs.berkeley.edu/~russell/aima1e/chapter02.pdf]

[URL_32] [http://www.allBookstores.com]

[URL_33] [http://pages.ebay.com/help/welcome/bid.html]

[URL_34] [http://auctions.yahoo.com/phtml/auc/us/tour/1-2-bid.html]

[URL_35] [http://auctions.yahoo.com/phtml/auc/us/tour/1-3-auto.html]

[URL_36] [http://www.ccs.neu.edu/home/kenb/com3337/rmi_tut.html]

[URL_37] [http://www.fipa.org]