AN ELECTRONIC MARKET FOR AGENT-SUPPORTED BUSINESS NEGOTIATIONS

Eva Chen

A Thesis
In
The John Molson School of Business

Presented in Partial Fulfilment of the Requirements
for the Degree of Master of Science in Administration (DS/MIS) at
Concordia University
Montreal, Quebec, Canada

November 2003

Your file Votre référence
ISBN: 0-612-90980-8
Our file Notre référence
ISBN: 0-612-90980-8

Abstract

An Electronic Market for Agent-Supported Business Negotiations

Eva Chen

As the information age grows, electronic commerce expands to every facet of business, creating new marketplaces for buyers and sellers to transact at their convenience. Electronic negotiation is essential to generating competitive markets for business transactions. Furthermore, the use of multi-issue negotiation allows for cooperative and competitive discussions. However, multi-issue electronic negotiation is a complex process that requires an agent system to support novice and sometimes experienced users.

This thesis involves: (1) the design and implementation of an electronic market (eAgora) that allows individuals to negotiate on multiple issues and (2) the development of an intelligent agent system capable of assisting market participants in their negotiation activities. This is achieved by utilizing an innovative framework based on Agent-Oriented, Fusebox and traditional systems development methodologies. Both the electronic market and agent system implementations are verified in a preliminary usability test.

Acknowledgment

This thesis is made possible through the support of so many wonderful people. Dr. Kersten, your encouragement and foresight have been the guiding light throughout my journey. I cannot thank you enough. Dr. Vahidov, your knowledge of agents is second to none. Dr. Laframboise, your patience and eye for detail have truly elevated this work. I could not have wished for a better team. Thank you so much gentlemen.

To my friends and colleagues, your support kept me going through thick and thin. "Not to mention your constant questions about my progress really forced me to work." Min, Erica, Yves, Maria, Sarah, Maggie, Derek, Brinda, Benoit, Bernice, Josh, Paul, Ka Pong and Norma, thank you.

To my best friend for always pushing me to do better, Phil, you are the best.

Mom, Dad, Wendy, Man Fai and Cherlyn, this thesis is completed because of your financial help and most of all, your love.

And finally, Jeff, for all that you have done, your name should be on the title page next to mine. "The bunny could not have finished the race without the turtle."

Table of Contents

List of Figures

List of Tables

1 Introduction

After ten years of existence, the idea of electronic commerce is no longer a novelty. Billions of dollars in products and services are bought and sold daily by businesses, consumers, governments and organizations over the Internet. It is the medium of the new economy that functions 24 hours a day, 7 days a week, 365 days a year all around the world (Hartman, Sifonis, et al., 2000).

The evolution of electronic business started with humble static pages, on which merchants posted advertisements for their goods and services. As computation power grew, databases were added to Websites in order to create dynamic pages that interact with the user. Virtual stores were formed, allowing buyers to shop and pay for products digitally. This advancement led to countless dot-com vendors selling a multitude of products over the Internet (Turban, Lee et al., 2000).

Frustrated with performing extensive searches for the right product at the right price, consumer-oriented groups invented software agents to help users crawl the Web for the best offer. Agents are software programs that run continuously and autonomously with assigned goals. Some sellers retaliated by blocking these agents from their site, but this action also prevented exposure of their site to buyers, while others introduced the idea of negotiable prices. For example, sites such as priceline.com require that buyers bid for the products instead of posting fixed amounts for them. This forces the shopper to commit to the transaction process rather than just browse through prices.

However, the focus on price alone limits the possibility for mutually beneficial agreements because sellers and buyers are pinned as adversaries fighting over one issue. Alternatively, multi-issue negotiation allows issues to be added to the bargaining table such that parties can derive creative and mutually favorable solutions (Lewicki, Saunders et al., 1997; Raiffa, 1998; Thompson, 1998). For example, when buying a car a single-issue negotiation obliges the buyer and dealer to fight over the price of the car where one gains through the loss of the other. However, in multi-issue negotiation the dealer may suggest to add an anti-theft device if the buyer accepts the dealer's proposed amount. Even though this offer is more expensive than the buyer's preferred price, the buyer may agree to this offer given that the anti-theft device means greater saving on the insurance payment over the life span of the car.

Nevertheless, researchers have found that practice of electronic negotiation is difficult for inexperienced individuals and the addition of multiple issues complicates matters even more. They believe that an agent supporting multi-issue negotiation can improve the process for users (Maes, Guttman, et al., 1998; Kersten and Lo 2001). They argue that the agent's role in negotiation is to monitor user's actions by providing a warning when a mistake is about to be committed, advise participants on offers received and suggest offers that negotiators may propose to their counter-part.

This thesis aims to design and implement a Web-based environment where individuals can negotiate over any product with multiple issues. Furthermore, an agent system is introduced to support the market participants in their negotiation activities. The work is based on three areas of research: (1) electronic markets (Section 2), (2) business negotiation (Section 3) and (3) agent technology (Section 4). The details of the research objective are explained in Section 5. The complexity of these three areas requires that an innovative framework be used to develop the system. Section 6 introduces the framework and various methodologies employed to construct it. The analysis, design and implementation of the contrived system (eAgora) are presented in Section 7 in accordance with the framework. Usability testing on the system is illustrated in Section 8, and Section 9 concludes the thesis.

2 Electronic Commerce and Electronic Markets

This section outlines the fundamentals of electronic commerce and the factors affecting electronic markets in competition. In order to set the grounds for discussion, Section 2.1 introduces the history and characteristics of electronic commerce. This is followed by Section 2.2, which examines the different types of electronic markets formed by the growth of electronic commerce and the need for negotiation in the marketplace.

2.1 Electronic commerce

In the past few years, both practitioners and academics have deemed electronic commerce (EC) to be the driving force for a new economy based on information. A new wealth of available information precipitates transformations encompassing everything from business processes to the structure of entire industries and even to the very core of worldwide socioeconomic (Tapscott and Caston, 1993; Applegate and Holsapple, 1996; Katz and Aspden, 1997).

The practice of doing business on an electronic network is not a recent phenomenon. Applications of EC began in the early 1970s with large corporations looking for means to share information across a wide span of subsidiaries. These first systems were implemented on expensive private networks allowing access to selected members (Scott-Morton and Allen, 1994). Examples of such rudimentary EC systems are electronic fund transfer (EFT) that permitted banks to transfer financial data and electronic data interchange (EDI) that allowed companies to exchange transactional information (Swatman and Swatman, 1992). However, with the commercialization of the Internet, the notion of EC became solidified. Suddenly, millions of customers and suppliers are able to link together through a rapidly expanding, inexpensive and easily accessible network. The growth of public networks (Worldwide Web), standardization of protocols (Hypertext transfer protocol) and languages (Hypertext markup language) permit companies of any size to enter the realm of EC (Zwass, 1996). By the mid 1990s, following the path of visionary start-ups such as Amazon.com, ebay.com and Dell.com, businesses of every facet developed an interest to expand operations onto the Internet creating a vast array of dot com companies (Hartman, Sifonis, et al. 2000).

2.1.1 Electronic commerce classification

To simplify the taxonomy of electronic businesses, the classification of EC is purely based on the nature of transactions between participants (Applegate and Holsapple, 1996; Turban, Lee et al., 2000). The following five commercial types have been identified:

- Business-to-business (B2B): This is the most common type of business transaction and it involves companies trading with companies.
- Business-to-consumer (B2C): This type is seen in retailing businesses that sell to individuals.
- Consumer-to-consumer (C2C): Instances of individuals trading directly with other individual such as through classified advertisements are examples of C2C.

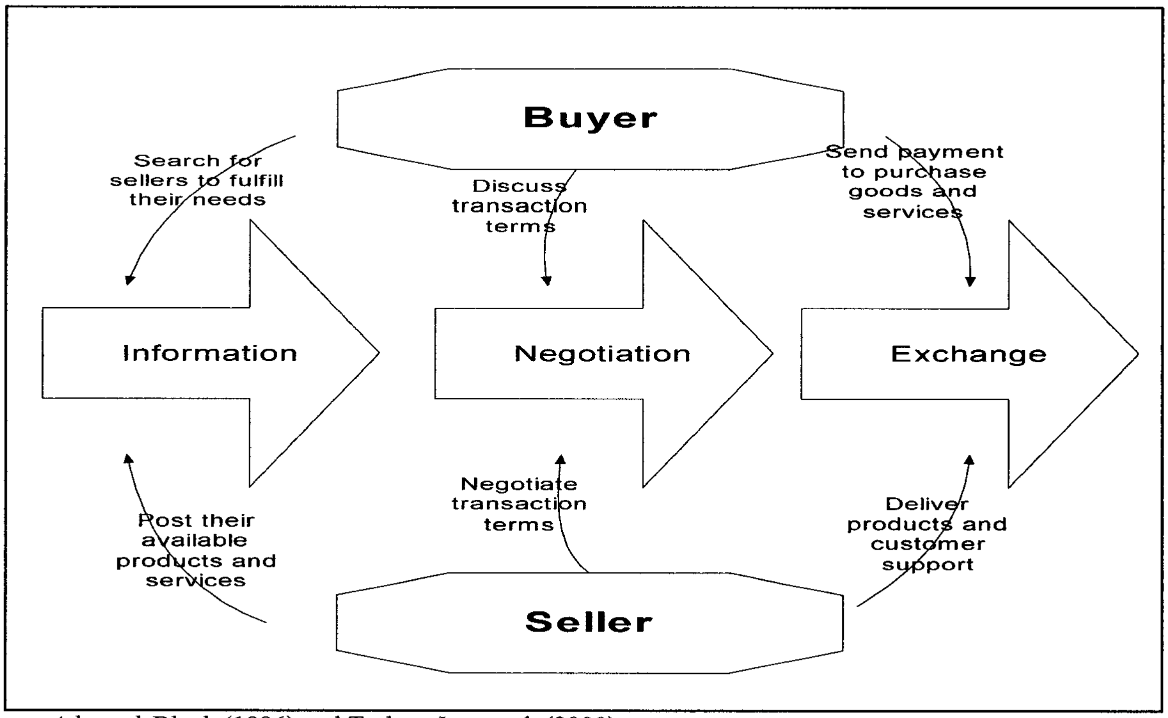
- Consumer-to-business (C2B): Rarely observed, this arrangement has individuals selling to businesses.

- Business-to-administration (B2A): Transactions that take place between the internal units of an organization are such type of commerce.

An increasing number of government organizations also engage in EC with businesses to reduce cost and increase accessibility, which results in G2B and B2G transactions.

2.1.2 Electronic business transactions

In traditional commerce, participants communicated to transact in person, by telephone or by fax. Now with the advances in telecommunication technology, the activities required in the supply chain can be accomplished over the Internet. The central activities related to any type of EC are presented in the three fundamental stages of electronic business transaction on Figure 1 (Bloch, 1996; Turban, Lee et al., 2000).



Adapted: Bloch (1996) and Turban, Lee et al. (2000)

Figure 1. Electronic commerce transaction life cycle

The fundamental stages are:

- Information: Participants search for trade partners in this initial stage. Buyers explore the Internet to find suppliers that meet their product needs. Sellers advertise their products or services to attract potential buyers.

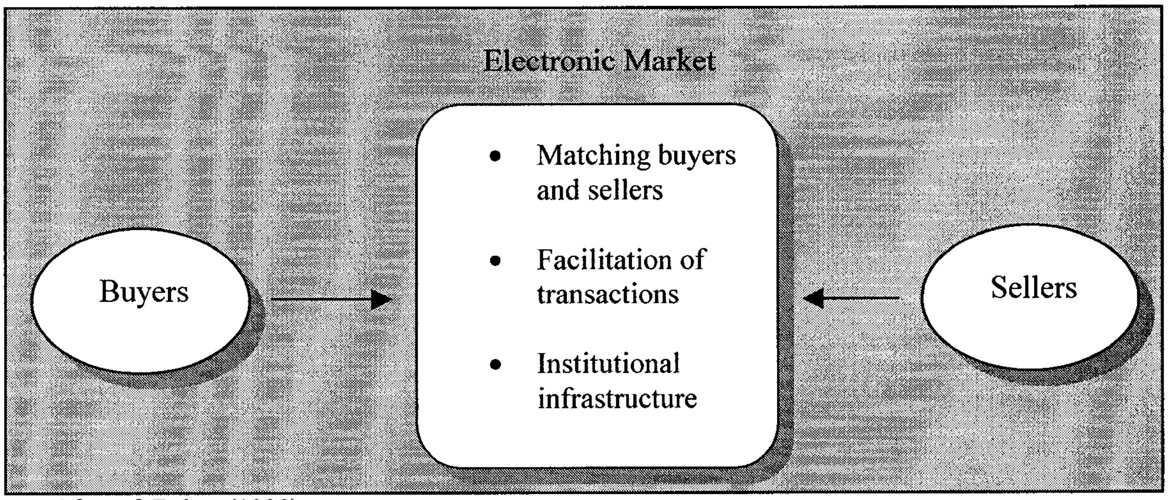
- Negotiation: Once trading partners are identified, this decision-making stage allows buyers and sellers to negotiate possible beneficial solutions. Negotiation can involve a single-issue (price) or multiple issues (price, delivery date, warranties, etc.). In most retail sites, back and forth negotiation is not available, as buyers can only reject or accept the terms posted by the seller.

- Exchange: The instance a deal is struck products or services are exchanged for payment and possible after sales service whereby the seller provides customer support services.

As the virtual trading places replace traditional marketplace, greater market efficiency is achieved by increased transparency and lower transaction cost (Malone, Yates et al., 1987; Kambil and Wilson, 1999). Consumers can buy directly from manufacturers' sites instead of paying the retailer. Businesses no longer need to limit their dealings to suppliers with whom they have physical contact because interaction and relationships can be formed with any suitable provider at any time over a large digital network. As more and more companies log onto the Internet, ensembles of business transactions are created to give rise to the electronic market.

2.2 Electronic markets and competition

An electronic market (EM) is a virtual marketplace where demand and supply meet together. Advances in information technology (servers, database system, etc.) have led to increased transactional effectiveness, decreased transaction cost and overall more efficient, frictionless marketplaces for buyers and sellers (Choi, Stahl, et al., 1997). Business participants see much potential in EM because such markets serve to match buyers and sellers, facilitate transactions and provide institutional infrastructure, all at a higher convenience and lower cost (Bakos, 1998). These three functions of EM are depicted in Figure 2 and described in the following:



Adapted: Bakos (1998)

Figure 2. Electronic market and functions

- Matching Buyers and Sellers: This function encompasses the business transaction stages of information and negotiation. Markets generate information to help sellers determine the product offerings that buyers may show interest in. Buyers seek out sellers with conditions that best fit their needs. Once seller offerings and buyer preferences are matched, negotiation enables parties to determine the price and specifications at which trade occurs.

- Facilitation of Transactions: The match of supply and demand requires the support of logistics (delivery of goods) and settlement (payment) to finalize trade. In addition, trust is necessary during exchange and hence, financial institutions and business associations play a pivotal role in seeing that transactions are honored.

- Institutional Infrastructure: Laws, rules and regulations are determined in the institutional infrastructure to facilitate the grounds for which market participants interact. For example, Visa and MasterCard have developed a security protocol, called SET (Secure Electronic Transaction) to protect credit card information used for Internet payments.

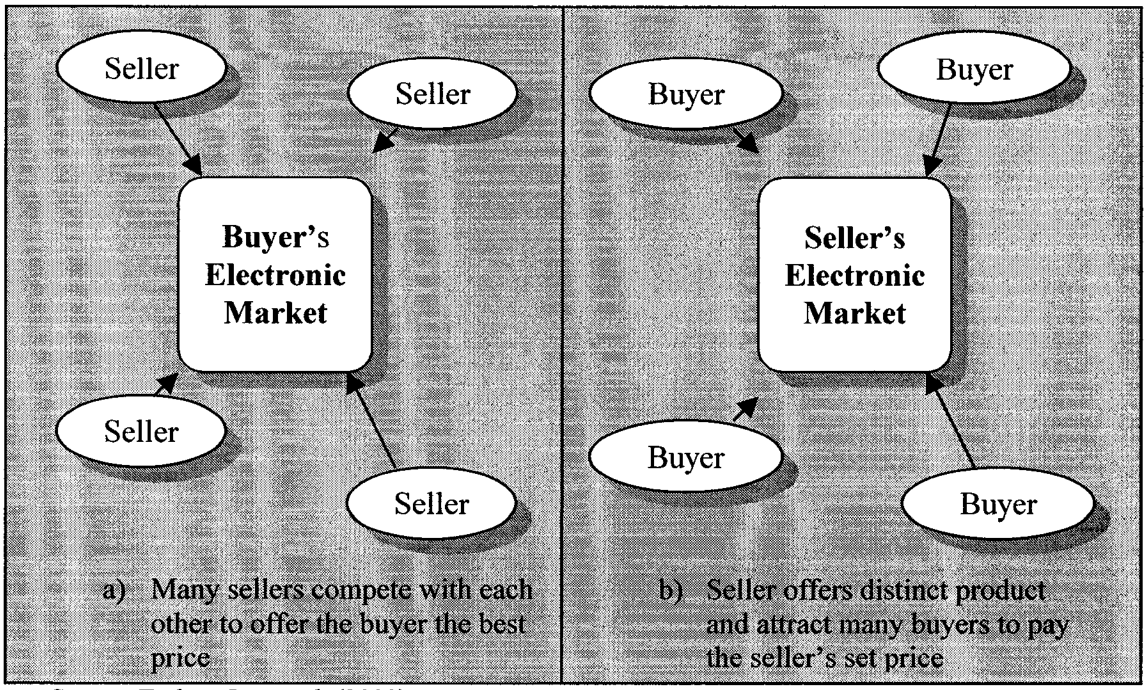
The Internet affects competition in the marketplace by providing customers and suppliers an abundance of information that allows for lower search cost, speedy comparisons and increased product differentiation. In a fixed price structure, these factors affect various forms of EM permitting participants on one side more power over the other (Bakos, 1998; Clay, Krishnan, et al. 2001). This can be seen in buyer's electronic market (2.2.1) and seller's electronic market (2.2.2).

2.2.1 Buyer's electronic market

As traditional markets evolve into EM, buyers no longer need to run from store to store in search of the perfect deal. They can easily employ technology to help find and compare products instantly. A classic example of technology used to foster this market structure is BargainFinder.com, where the software searches the Web to provide consumers with various suppliers' price for a given CD. Buyers' EM are mainly buyer-oriented sites that search and weigh price postings by sellers (see Figure 3a). Sellers in such markets encounter the burden of determining the value of a good in hopes that their posted price can compete with others around the world. On the other hand, buyers are advantaged because numerous sellers offer their desired product to them at low prices, and they choose the price they want to pay. Therefore, in a buyer's EM, the suppliers are disfavored since they are unable to charge according to buyers who value the products the most and as a result the allocation of resources is unfairly distributed (Parkin and Blade, 1997).

2.2.2 Seller's electronic market

In order to combat the buyer's EM power, sellers use the Internet to personalize and customize products to distinguish their goods from other firms and turn the market influence to their advantage (Warkentin, Bapna, et al., 2000). Retailers such as Dell.com that allow customers to build their own computer system on-line restrict buyers with specific requirements to limited offerings. In addition, through the advent of technology such as cookies (user preferences that sellers store on consumers' computers) and data mining (extracting valuable information from massive databanks for user preferences), sellers are empowered to target and attract buyers to pay for differentiated products. In short, the suppliers exert greater authority in this market because they host the sites and restrict consumers to the set price (Parkin and Blade, 1997; Bakos, 1998; Clay, Kishnam, et al. 2001). This market is depicted by Figure 3b.



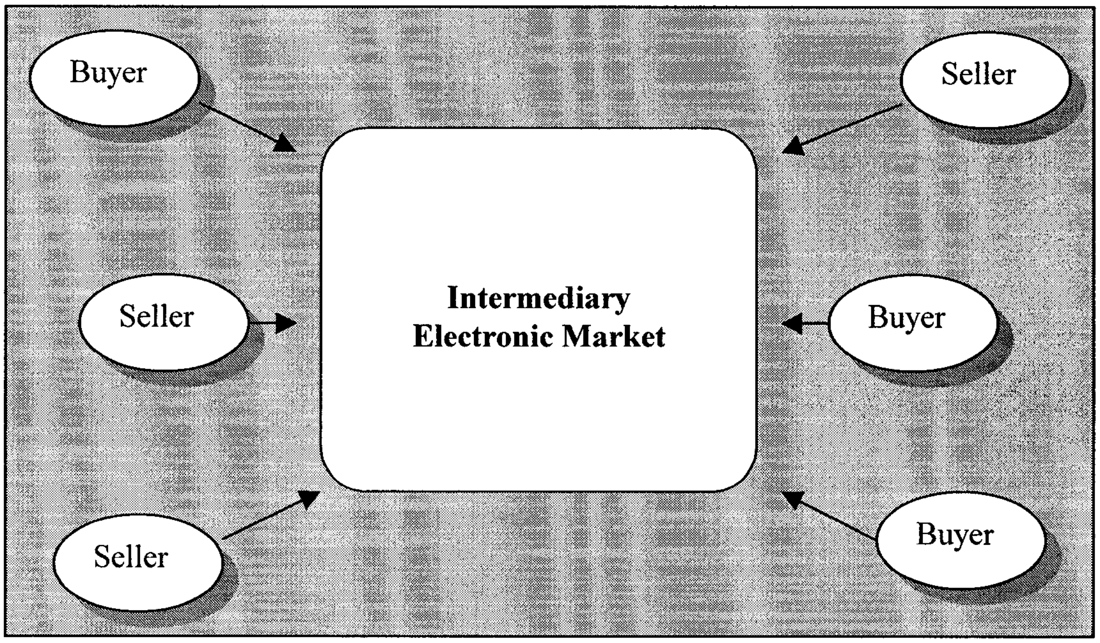
Source: Turban, Lee et al. (2000)

Figure 3. Buyer's and Seller's electronic markets

When prices are fixed, sellers are disadvantaged in a buyer's EM because profits are marginalized in fierce competition. Conversely, buyers are powerless in the seller's EM because they are required to pay the supplier's price. In order to create an equal environment for trade (Sarkar and Bulter, 1995), the intermediary EM (see Figure 4) is suggested by Bakos (1998).

2.2.3 Intermediary electronic market

In an intermediary electronic market, a neutral party establishes the platform for numerous buyers and sellers to negotiate over the price and conditions. Since back and forth discussions are allowed, different price mechanisms are discovered to enrich both sides of the market. The ability for electronic negotiations gives buyers more bargaining power, and sellers benefit from higher profits by collecting different prices from different buyers (Bakos, 1998).



Source: Turban, Lee et al. (2000)

Figure 4. Intermediary electronic market

In EC, countless participants transact together (whether it is in B2B, B2C, C2B, C2C or B2A relationships) through fundamental stages of interaction (information, negotiation, exchange). As EC soars to new heights, different EM can form depending on participant's ability to employ information and technology under a fixed price mechanism structure. However, since intermediary EM provides many buyers and sellers the possibility to negotiate, business transactions are enriched for both parties. Therefore, negotiation becomes essential to the formation of intermediary EM and topic of the next section.

3 Business Negotiation

Negotiation is an important component in establishing an equal opportunity EM for all participants to exchange goods, services and information with each other. Negotiation is defined as a decision making process by which two or more parties exchange ideas and offers in order to resolve initial differences in preferences (Lewicki, Saunders, et al., 1997; Raiffa, 1998; Thompson, 1998). It can be categorized either as bargaining or auctioning depending on various characteristics: issues (single-issue vs. multi-issue) (Bui, Yen, et al., 2001); number of negotiating parties (bilateral vs. multilateral) (Thompson, 1998), approaches (competitive vs. cooperative) (Strobel, 1999); technologies (Web-based NSS, automated negotiation, etc.) (Kersten, 2002); and many other characteristics.

Electronic business negotiation between buyers and sellers is a complex process that takes place in three stages, involves multiple issues discussed under mixed motives and uses various technologies for support (Benyoucef, Alj et al., 2001). These main characteristics serve as the premise of this thesis. Section 3.1 examines the general motives of negotiation, Section 3.2 categorizes the different forms of negotiation, Section 3.3 relates the essential stages of the bargaining process and Section 3.4 reviews electronic negotiation technology.

3.1 Negotiation approach

Negotiators are often described as having either a competitive or a cooperative motivation behind negotiations. By exploring these conflicting approaches, a better understanding of real world business negotiation can be achieved (Thompson, 1998).

3.1.1 Competitive negotiation

The competitive negotiation approach is often described as distributive where two parties bargain over sharing a "fixed pie". The negotiators' interests are purely opposite; such that whenever one party wins a bigger share the other party loses (Bazerman and Neale, 1992). Hence some researchers refer to this as a "win-lose" situation because the utility (i.e. satisfaction a party derives from negotiation) of the buyer and that of the seller are limited to the boundaries within a small bargaining zone (Thompson, 1998). An example of this situation is when a company (let it be referred to as the buyer) looking to outsource their information technology (IT) functions negotiates with a service supplier (seller) over the price of the contract. If, by receiving a high price the seller's utility is maximized (i.e. the seller obtains 60% of the fixed pie) then, by paying the elevated sum the buyer's utility is minimized (i.e. the buyer gets only 40% of the same pie).

Negotiators using the competitive negotiation approach often view the relationship with the other party as unimportant, and their main concern is maximizing the value obtained in a single deal at the other's expense. In addition, competitive bargaining usually revolves around a single issue such that resources for discussion are limited (the fixed pie), and information exchange is reserved to a bare minimum in order to prevent the opponent from gaining the upper hand by knowing one's bargaining power.

3.1.2 Cooperative negotiation

The cooperative negotiation approach is the opposite of the competitive form. In cooperative negotiation both parties confer in order to reach a common objective. It is also known as integrative negotiation that leads both sides to a "win-win" situation (Lewicki, Saunders, et al., 1997). Following the example provided above, the seller and buyer may engage in cooperative negotiation when discussing the date for IT service to commence. Since both parties may wish to start services quickly, the seller wants to receive initial payments as soon as possible, and the buyer promptly needs servicing. This form of negotiation expands the boundaries of the bargaining zone, in a way that an agreement is reached as the buyer's utility is being capitalized along with the seller's utility (Thompson, 1998).

Cooperators utilize multiple issues in order to find solutions that meet the needs of both sides. Researchers such as Raiffa (1998) have even suggested that as a negotiation diverges from a single-issue into a multi-issue problem, it moves from a distributive towards a more integrative approach. However, others argue that unless some common value is created by both parties multi-issue negotiation does not result in cooperative discussions (Kersten and Noronha, 1998). In fact, the acknowledgement and pursuit of a common goal by both sides is the essence of integrative negotiations (Lewicki, Saunders, et al., 1997). To reach a common goal the free and open flow of information is necessary form each of the parties, such that they are both aware each other's situation.

3.1.3 Mixed motive negotiation

Business deals most frequently observe the mixed motive negotiation approach where negotiators discuss a variety of issues that may be distributive or integrative. In reference to the previous example, seller and buyer settle on a contract under mixed motive negotiation, such that the issue of price is negotiated competitively and start date for servicing is discussed cooperatively. Therefore, mixed motive negotiation requires the existence of two or more issues, which are competitively and cooperatively discussed.

Even though these three approaches are very different, bargainers still have difficulties knowing how, when and with which issue to apply the different approach in order to achieve the most favorable agreement possible (Kersten and Noronha, 1998).

3.2 Negotiation classification

There are many ways to categorize negotiations, but for the purpose of this work, classification is focused on: (1) the number of negotiating parties (bilateral vs. multilateral) (Thompson, 1998), (2) the order of offer exchange (sequential vs. parallel) and (3) the number of issues on the bargaining table (single-issue vs. multi-issue) (Bui, Yen, et al., 2001).

Bilateral vs. Multilateral: These attributes of negotiation relate to the number of parties involved in the negotiation. When only two parties participate in a one-to-one discussion, the negotiation is deemed as a bilateral negotiation (bargaining). Multilateral negotiation is either a one-to-many or many-to-many negotiation, such as in auctions or stock exchanges. Multilateral negotiations generally engage few issues (price and perhaps quantity) in a competitive public arena, whereas bilateral negotiations are more private by nature and allow for a more cooperative negotiation setting (Ströbel, 1999).

Sequential vs. Parallel: Sequential negotiation refers to the chronological order in which offers and counter-offers are made. This characteristic of negotiation requires that offers are made one after another in the manner that only one proposal is on the bargaining table at once. Sequential discussion is a simple method of offer exchange that is most utilized in bilateral negotiation. Parallel negotiation allows many proposals to be discussed at one instance. Multilateral negotiations often involve paralleled bidding with an auctioneer (or some form of match-making system) to account for all the bids and reward the product to the best offer.

Single-issue vs. Multi-issues: Single-issue negotiation is frequently distributive, since the gain achieved by one side results necessarily in loss for the other. On the other hand, multi-issue negotiation is seen as integrative, such that possible joint gains can be reached for all parties.

Ströbel (1999) differentiates bargaining from bidding (auction) as two distinct types of negotiation based on the different characteristics of negotiation. The emphasis is that bargaining is a form of bilateral multi-issue negotiation and auction conversely consists of multilateral single-issue negotiation. However, from observation of real negotiation processes, it is revealed that bargaining can turn to single-issue negotiation when one issue becomes the sole subject of discussion while other issues are temporarily stabilized.

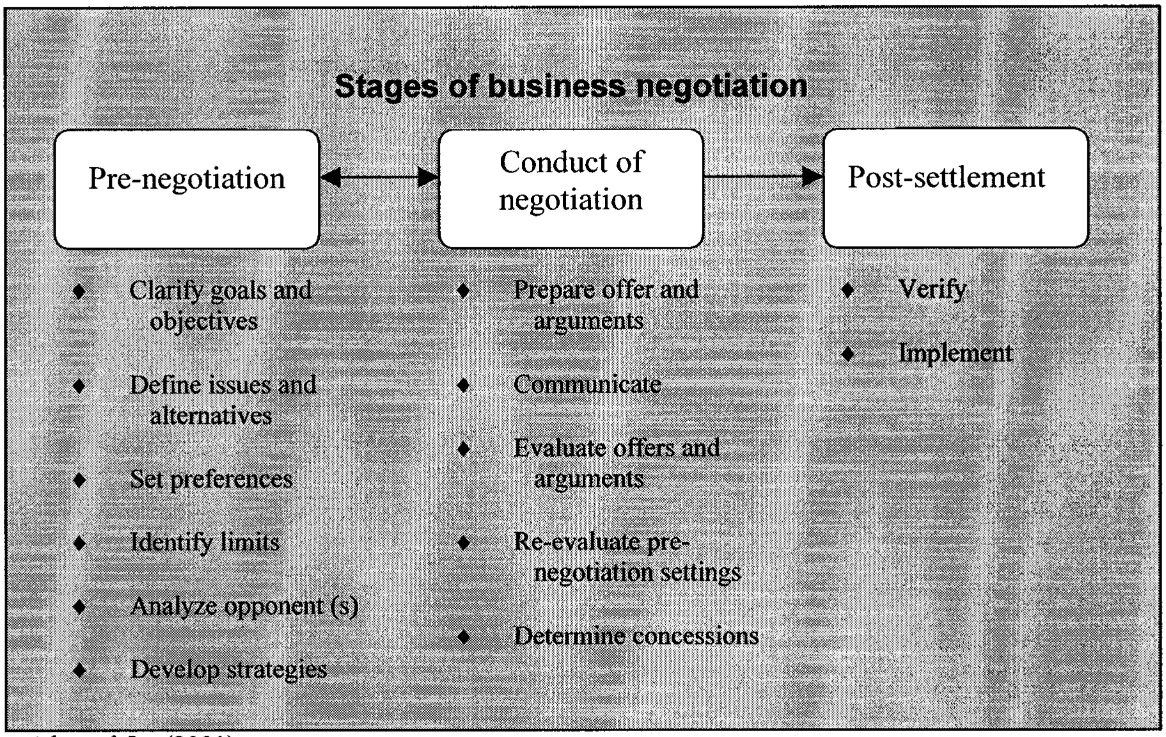
3.3 Negotiation stages

Business negotiations follow three basic stages: pre-negotiation, conduct of negotiation and post-settlement as described in previous studies (Lewicki, Saunders, et al., 1997; Thompson, 1998; Kersten and Lo, 2001). These stages are described in Figure 5 in addition to the key tasks involved in each stage.

Pre-negotiation refers to the preparatory stage of negotiation. In the beginning negotiators determine their goals and objectives, and then they define the issues and alternatives. Sequentially, limits and preferences are set. The opponent is analyzed and strategies are developed for the conduct of negotiation.

The conduct of negotiation is the process in which two parties prepare and swap offers and arguments. These offers and arguments are evaluated and at times, revisal of pre-negotiation settings is necessary. Concessions are made and possibly a settlement is agreed upon.

In the post-settlement stage after an agreement is reached, verification and implementation of settlement terms is actualized.



Adapted: Lo (2001)

Figure 5. Tasks in business negotiation stages

Negotiators most often make the mistake of neglecting important tasks in these stages; especially those in pre-negotiation, which may consequentially lead to undesirable settlements or negotiation breakdown. However, with the advent of negotiation support technologies, these outcomes may be avoided (Kersten and Lo, 2001).

3.4 Electronic negotiation

Electronic negotiation (e-negotiation) not only implies carrying out negotiations over the Internet but also the possible usage of support technologies in negotiation. E-negotiation combines the existing benefits of traditional negotiation and those derived through deployment of the Internet medium, such as decreases in cost and increase in the number of participants. As a whole, the advantages for e-negotiation are: (1) the potential creation of a "win-win" settlement between trading parties, (2) the prospective of establishment of strong partnerships between buyers and sellers, (3) a decrease in coordination and transaction cost and (4) the engagement of more participants to the process of negotiation. In contrast to electronic auctions, e-negotiations are more relationship focused, where offers can involve various issues as opposed to simple exchanges on price alone. This provides opportunity to create a cooperative situation where negotiators make concessions on less valued issues in favor of better deals on preferred issues and discuss issues that may lead to Pareto optimality[1] (Raiffa 1998; Rosenchein and Zlotkin, 1994). Since cooperative negotiation requires a rich exchange of information and establishment of a common goal, strong partnerships are thus permitted to form between participants (Kersten and Noronda, 1999). Given that no effort is needed to structure face-to-face meetings, coordination and transaction costs are lower for the overall procedure and accessibility is granted to a greater number of participants (Teich, Wallenius, et al., 1999). In addition to generating these advantages, some types of e-negotiation technology may overcome negotiator pitfalls relating to the different approaches and tasks in the bargaining stages.

---

[1] Simply means the maximization of both sides' satisfaction.

3.4.1 Electronic negotiation technologies

There are various forms of e-negotiation technologies that generate different benefits for participants. These technologies include: electronic message exchange, Web-based Negotiation Support System (NSS), Web-based NSS with Agent assistance and automated negotiation, all of which are categorized in Figure 6.

Figure 6. Electronic negotiation technologies

Electronic message exchange is simply negotiation through electronic mail (email). This is the basic form of e-negotiation, where parties submit offers and messages to each other's electronic mailboxes without the need of specialized negotiation software (Cronson, 1999; Lee, 1998; Thompson and Nalder 2002). Electronic message exchange provides little support to negotiators beyond the management of their email. In addition, the information submitted is unstructured and unorganized, which can be confusing for the negotiator and prolong the negotiation process (Kersten, 2002).

Alternatively, **Web-based NSS** provides users with an on-line support system that utilizes decision-making and negotiation analysis to structure and organize information. The Inspire system was the first Web-based NSS developed to train negotiators through the three stages of business negotiation (Kersten and Noronha, 1999). Participants are provided with problem structuring and analysis tools in hopes of helping them visualize different angles of negotiation. This is achieved by eliciting negotiators' preferences to construct a utility function (linear regression of utility vs. possible offers). Moreover, issues and alternatives to each issue are separated from messages in order to facilitate better communication between negotiators. This system has been proven successful in researching and teaching negotiation to business, education and other organizations (Kersten and Noronha, 1998).

As more and more participants engage in electronic negotiation, problems such as information overload and lack of negotiation knowledge will impede users in realizing effective and efficient negotiations. Even if Web-based NSS provides an organized and well-defined approach to negotiation, some participants may have difficulty in finding the precise knowledge required for their exchange. Furthermore, some users do not have the formal experience, training in decision-making to define the specific knowledge for negotiation or willingness to engage in bargaining tactics. In other words, agent technology is used in conjunction with Web-based NSS to present users with information specified and sorted to their exact requirements either in the form of agent technology integrated to Web-based NSS or automated negotiation.

Due to the employment of **agent technology integrated to Wed-based NSS**, information is better managed and knowledge is presented to the precise needs of the user. The Atin agent was developed to assist users in the Inspire environment while offering active and context-dependent advice (Kersten and Lo, 2001). Participants in a field test found the agent to be helpful during negotiation, and it supplied a more practical environment compared to Inspire alone. However, Atin does not provide negotiators with timely suggestions on possible offers to propose to their adversary nor does it critique the offers received by the users. Atin is somewhat restricted to functioning only in the Inspire environment, in a way that its reasoning abilities cannot extend to other NSS. Therefore, the results also demonstrated that an improvement on the agent's reasoning ability and better integration of the agent to NSS are required for greater functional and behavioral support. In addition, researchers in agent technology have stressed that a valuable agent need possess not only intelligence, but also adaptive abilities. In the sense that, for example, offer suggestions should be tailored to the various factors in the negotiation situation (Negroponte, 1997; Maes, Guttman, et al., 1998; Li, Huang, et al., 2002).

Automated negotiation is a term given to negotiation that employs agent technology with little or no human interaction in the process. This technology is not designed to help human-to-human negotiations in EC, but rather agent-to-agent transactions in a **multi-agent system** (MAS) (Nunamaker, Dennis, et al., 1991; Foroughi, 1995; Beam and Segev, 1996).

MAS is a solution in a complex market environment comprised of many negotiators and agents representing their interests, where a system is used to manage the multipart problems that include distributed data, knowledge and even control. The first MAS were implemented to negotiate in enclosed system environments. PERSUADER is a system developed by Sycara (1991) in association with Rosenchien (1994) that conducts negotiations in the area of labor disputes. In this system, self-interest agents utilize a middle agent to manage requests. TEAM is a multiple agent framework infrastructure designed by Lander and Lesser (1993) that allows for communication and cooperation among heterogeneous and reusable agents by coordinating the activities of agents with individual roles. In the late 1990's, researchers such as Jennings and Wooldridge (1998) argued that MAS is ideal in combining intelligent agent technology with e-negotiations. The department of Computer Science and Engineering at the University of Minnesota put forward MAGNET, a multiple agent negotiation test-bed which uses an administrator agent to coordinate the activities of various agents in order to support planning and contracting in EC (Collins, Youngdahl, et al., 1998). Kasbah and Tête-@-Tête are the two most notable examples to emerge in MAS in e-negotiation, where an agent is first given pre-negotiation settings and a strategy to follow and then searches for other agents to negotiate and form settlements with (Maes, Guttman et al., 1998). These agents act in a simplified environment in comparison to humans, and they follow specific protocols that may force them to reach a negotiated agreement, which does not necessarily reflect the real world (Winoto, et al., 2002). Due to the degree of great uncertainty and mistrust surrounding completely automated negotiation, its usage has remained mostly in academic research. The next section examines agents in detail from the traits required to assist users in negotiation to the mathematics that give agents intelligence.

4 Agent Technology

Over the past years, extensive research has been directed to the topic of agent technology. In this section, some of these studies are outlined along with features essential to agent design. Section 4.1 gives a general view of how agents have already been employed in EC. Section 4.2 examines the important traits of agents for negotiation. Section 4.3 explores the concept of intelligence in agent technology utilized in electronic negotiation.

4.1 Agents in electronic commerce

As more people log onto the Internet, more information becomes available and users face the exacerbating problem of information overload. Not only does the Web provide an enormous volume of data, but also the data exists in a broad range of formats (text, image, audio, video, etc.). In addition, the information posted on Websites changes at a rapid rate. Thus, the solution to managing this magnitude of data is to use intelligent and software agents to assist in the activities of EC. Agents serve to conduct routine task, search and retrieve information, support decision-making and act as a domain expert. They sense users' needs and act autonomously without human intervention, resulting in significant time saved (Moukas, Guttman, et al., 1998).

Agents range in a variety of types, starting from those with no intelligence (software agent) to smart agents that exhibit intelligent behavior and even in some cases, the ability to learn. According to MIT Media Lab researcher Pattie Maes (1994), an intelligent agent modifies its activity to provide the appropriate help to users by any one of these methods: "looking over the shoulder" of the user, employing direct and indirect user feedback, learning from examples given by the user, and asking the agents of other users. Most importantly, the difference between the functioning of a smart agent and that of a software agent rests in the way logic rules are created and/or managed. For example, a search engine is a software agent that looks for related information according to the word(s) given by the user. The exploration is performed by comparing keys works on a Website to the input provided by the user (if a match is found, then that site is displayed to the user).

On the other hand, a smart agent monitors the sites that a user visits and offers related sites that may interest the user without any human intervention (Lai and Yang 2000). The search in both cases is identical, but the intelligent agents determine when the rules are executed as opposed to software agents that require direct user input.

According to Maes, Guttman and Moukas (1998), agents have the role of mediator in EC, which can assist users in the various stages of business transactions. In the information stage, agents serve first to help identify buyer needs (needs identification), which is beneficial to both buyer and seller by creating a market that matches these needs with the right product. They can further aid buyers in selecting the appropriate product (product brokering) and seller to purchase from (merchant brokering). Alternatively, suppliers benefit by using agents to identify the appropriate customers to sell to (buyer identification) (Turban, Lee et al. 2000). In the negotiation stage, agents may assist discussions between both parties (Section 3.3). During the exchange stage, buyers and sellers can utilize agents to track and process payment and delivery, in addition to supporting after sales services.

Some commercial models of smart and software agents are shown in Table 1, according to their assistance in different stages of electronic business transaction.

	Buyers	**Sellers**
Information	▪ BargainFinder: searches the price of a specific CD from on-line retailers and displays a list of prices.	▪ ZineZone: gathers information about users and formulates profiles in order to send personalized ads to the browser.
Negotiation	▪ Kasbah: negotiate price with other Kasbah agents using three possible strategies. ▪ Tête-@-Tête: bargains with other agents in the system over multiple issues.	
Exchange	▪ Gator.com: automatically fills order forms with user's personal information.	▪ eResponse Smart Agent: answers simple customer service questions and frees up time for human service representative to handle difficult problems.

Table 1. Agents assisting buyers and sellers in electronic commerce

Furthermore, researchers such as Chavez et al. (1997) have suggested that the role of agents be limited to mediator in EC, which implies that the agent only acts as a broker between a group of information sources and a group of applications while humans still command the business process. Without this control, users are less likely to accept the agent's assistance or can even misconstrue the agent's intentions.

4.2 Agent traits

Agent traits have long since been a popular topic of investigation in agent design (Lui, 2001). In this section, basic traits are explained through studies relating to the fundamental defining elements of an agent. In addition, adaptive traits are examined to reveal the essential qualities an agent needs to progress through different environments and time. Finally, the most notable negotiation traits are described to show the rudiments of building an intelligent agent for negotiation. These three groups of traits are depicted in Figure 7 to exhibit the whole required to develop intelligent agents for e-negotiation.

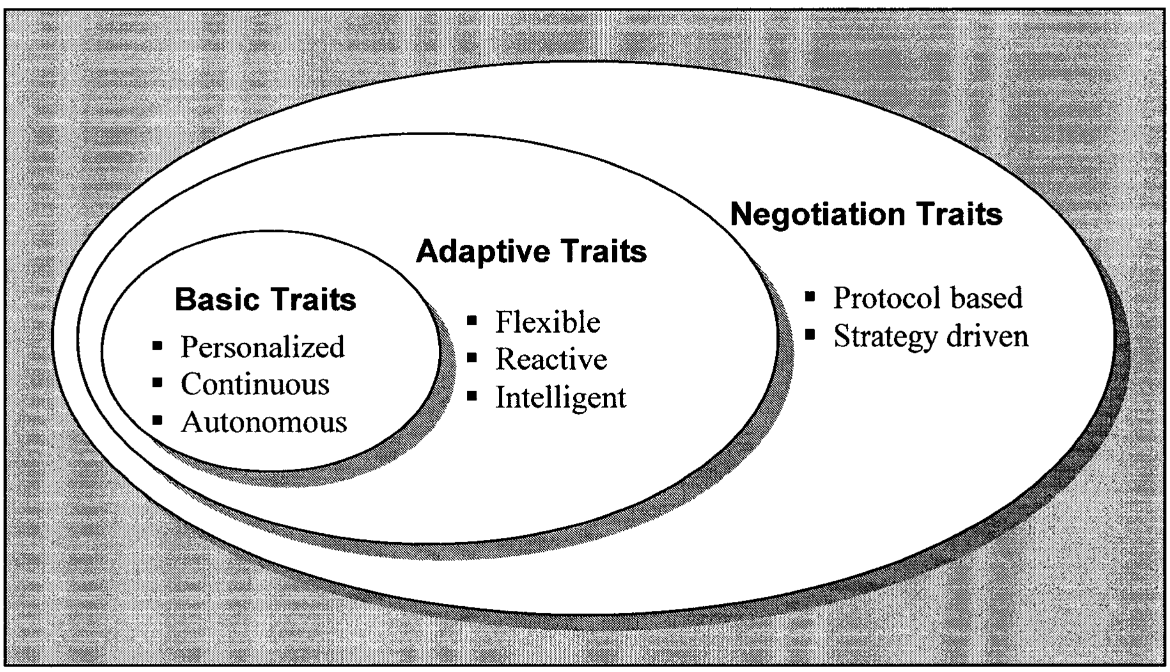


Figure 7. Intelligent agent traits for electronic negotiation

Basic traits

Web-based agents are software programs that reside in a local or remote site to help humans perform repetitive tasks. These agents are personalized, continuous and autonomous (Maes, Guttman, et al. 1994; 1998). In essence, agents must possess a certain level of autonomy in order to initiate the communication process without human intervention. This is achieved mostly by building the agent separate from the Website or the system that manages the Web pages. Agents also need to operate in a continuous manner in the background without any monitoring. In addition, since the goal of EC agents is to assist buyers and sellers in various business activities, these agents should be personalized to reflect the specific requirements of the user.

4.2.1 Adaptive traits

Even though there are many published works on numerous characteristics an agent should exhibit to better assist users, flexibility, reactivity and intelligence are three design issues that have been repeatedly mentioned because these allow the agent system to adapt to different situations and through time (Lui, 2001).

Flexibility refers to designing an agent that can expand its knowledge-base to better serve the user over time. As EC progresses, the requirement on an agent system will change such that the behavior and functions of the system need to be easily updateable by the developer in a logical approach to match the change in demand (Jennings, Sycara, et al. 1998).

Reactivity describes an agent's capability to modify its behavior without direct user intervention to the environment of the system in which it functions, in such a way that there is a continuous interaction between agent and environment. Reactive agents necessitate some learning mechanism within that permits adaptability to the real-time needs of EC. For example, a reactive agent in negotiation is a learning agent that automatically provides users with specific, expert and progressive bidding advice as offers and counter-offers are being proposed (Weiss, 1998).

Intelligence refers to an agent's ability to reason with imprecise, uncertain and partial truth associated with almost every aspect of real-world problems. Intelligent agents use heuristic techniques to assess numerous input parameters and present users with various feasible solutions under different situations. Heuristic techniques such as knowledge-based reasoning, fuzzy logic, neural networks, genetic algorithms or a combination of these are constructed to model human cognitive capabilities in the decision-making process (Azvine, Azarmi, et al. 1997; Lai and Yang 2000). (A further discussion on intelligent agents is provided in Section 4.3)

4.2.2 Negotiation traits

The employment of artificial intelligence to negotiation was first studied in distributed artificial intelligence and multiple agent systems (Lander and Lesser, 1993; Nunamaker, Dennis, et al., 1991; Sycara, Roth, et al., 1991). Subsequently, the introduction of intelligent agents to EC aimed to revolutionize the way exchanges are conducted over the Internet (Maes, Guttman, et al., 1994; 1998; Lo, 2001). As a result of these studies, the concepts of protocol and strategy have become essential to the development of negotiating agents in general. (Beam and Segev, 1996; Jennings, Sycara, et al., 1998; Jennings and Faratin, 2001)

Agents in negotiation must follow protocols and strategies. Protocols are defined as the rules of the negotiation process, which enable parties to negotiate fairly, whereas strategies are the specific tactics, used by a party to accomplish a goal. If negotiation can be considered to be a game, then protocols determine "the rules of the game" and strategies are "the methods used by the players" in the game.

Protocols dictate communication patterns and the scope of the decision-making process involving parties and e-negotiation system. They can be classified in two perspectives: axiomatic and strategic (Winoto, et al., 2002). Axiomatic protocols follow bargaining theory that relies on a centralized decision maker to determine the division of goods based on the intentions submitted by each party. Conversely, strategic protocols require that negotiators make offers and counter-offers to resolve their division dilemma. The first studies employed techniques and assumptions from game theory to predict the outcome of strategic bargainers. Agents were programmed to behave perfectly rationally and with faultless foresight. In addition they would search in an exhaustive fashion for the ideal solution (Rubinstein, 1982; Sandholm, 1999). However, the results from these studies deviated greatly from real-life negotiations since humans rarely know the preferences and full intentions of their counter-parts. Furthermore, users are often uncomfortable negotiating in an environment where their preferences are revealed to their counter-part, since this would limit the variety of strategy for negotiators (Oliver, 1997).

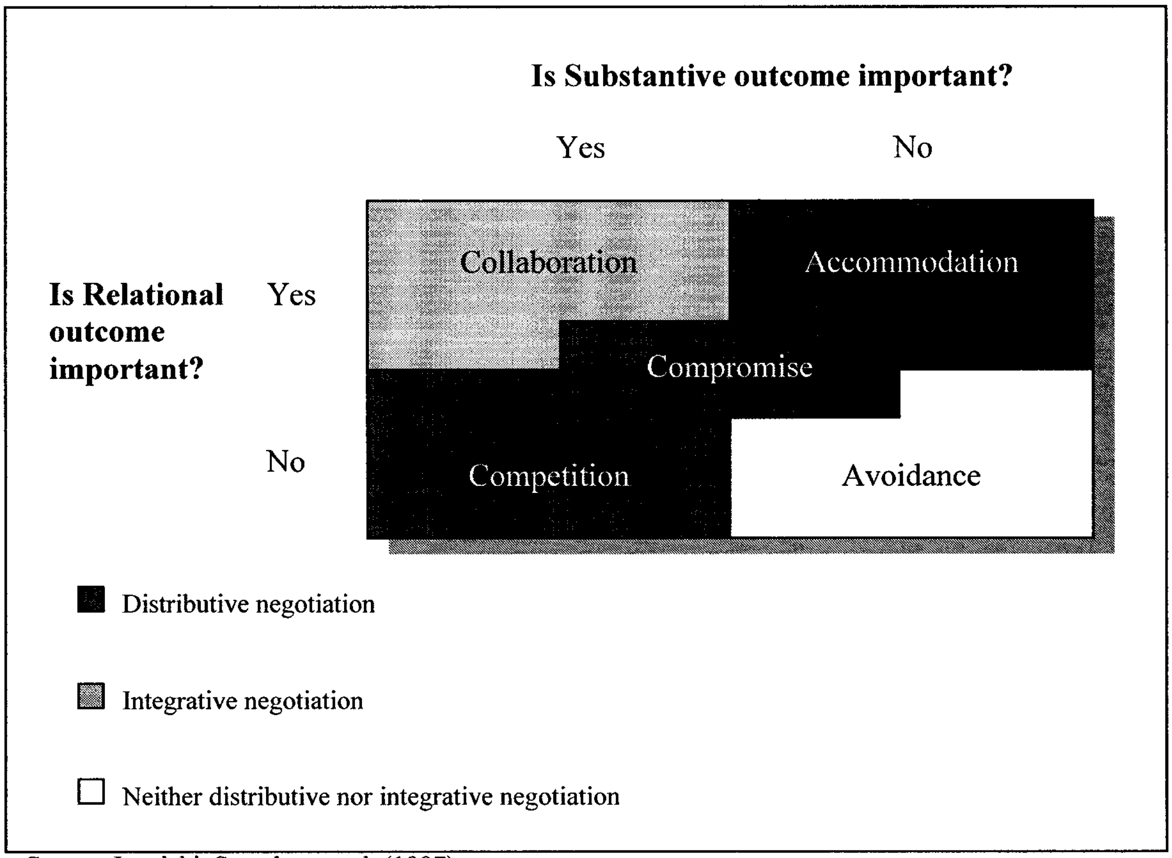
Heuristic techniques were later used to determine negotiation results. Agents in these studies were designed using competitive strategies to find a solution in an acceptable bargaining zone instead of the ideal one (Zeng and Sycara 1998; Sandholm and Vulkan, 1999). Nevertheless, these researches do not reflect realistic bargaining since they limited the number of issues and focused on competitive strategies.

Recently, argumentation-based models have been utilized to mimic human negotiations that center further on natural language resembling negotiations, and successful results from these studies show that more flexibility need be allowed in the negotiation process (Jennings and Faratin 2001). Moreover, investigators such as Norman (1994) and Holsapple (1996) argue that agents must remain subservient to users in order for negotiators to trust and utilize them. The agent's advice is offered to the user who retains the choice to accept or reject it. The ability to turn off the agent provides user with a greater sense of control, which in turn creates an agent that is more likely to be adopted.

In order to achieve more flexible and realistic negotiations, Oliveira (1999), Ströbel (2001), and Winoto (2002) propose that the following protocols be included in agent-assisted negotiations:

- Permit negotiations without disclosing user preferences.
- Let negotiators bargain on multiple issues.
- Encourage the usage of various strategies.
- Give negotiators time to evaluate offers.
- Let revision of offers before settlement is reached.
- Allow users to trust and utilize agent in e-negotiation

Where protocols apply to all participants in negotiation, strategy is individualized depending on the goals of the negotiator. The strategy employed by participants is based on their decision over the importance of substantive and relational outcome. Substantive outcome refers to the gain achieved from bargaining, whereas relational outcome emphasizes the relationship established in the process. Accommodation, avoidance, collaboration, competition and compromise are five basic strategies relating to these two outcomes (Lewicki, Saunders, et al., 1997), all of which are depicted in Figure 8.



Source: Lewicki, Saunders, et al. (1997)

Figure 8. Five basic strategies

The following provides descriptions of the five strategies:

- Avoidance: is equivalent to non-negotiation, since neither substance nor relationship is important to the negotiator. This strategy is often found in situations where the participant is indifferent to negotiation such that their involvement is purely for the sake of being present.

- Competition: is the strategy used when the negotiator is concerned only with winning this bargain, such that small or no concession are made. Competition is employed in clearly distributive negotiations, which cast no regards to the effect of the relationship with the counter-part or subsequent exchanges that may follow.

- Accommodation: is the opposite of competition where relationship building is more important than substantive outcome. Negotiators may sacrifice gains in large concessions in a distributive negotiation to secure future deals with the other party.

- Compromise: is the midpoint between competition and accommodation in "win-lose" bargaining, where concessions match the ones made by the opponent. When negotiator cannot reach an agreement, they often compromise to "split the difference".

- Collaboration: is employed in integrative negotiations. Participants work together on achieving a common goal (see Section 3.1.2). The focus is not on the size of concession made but rather on Pareto optimality. Collaborative strategies require a great amount of trust between negotiators to share private information on each party's preferences and constraints.

The composition of negotiation agents needs to be multi-dimensional, because along with basic and adaptive traits, agents must encompass negotiation traits (i.e. protocol based and strategy driven) in order to properly assist users through the bargaining process.

4.3 Intelligent agents in negotiation

Numerous researchers have noted intelligence as the fundamental trait of an agent, to the extent that some have expressed that any system without intelligence should not be deemed an "agent" (Negroponte, 1997; Jennings and Wooldridge, 1998). In order for any software to exhibit machine intelligence, it must contain some reasoning mechanism (heuristic technique) based on a collection of rules or mathematical functions that generate the best solution(s) under some given circumstance. The two most employed heuristic techniques that allow for symbolic knowledge extraction are knowledge-based reasoning (Section 4.3.2) and fuzzy logic (Section 4.3.3). Before any of these techniques can be described, multi-criteria decision analysis (MCDA) need first be explained as the basis for which intelligent agents are able to address multiple issues in their reasoning (Section 4.3.1).

4.3.1 Multi-criteria decision analysis

MCDA is a field in operations research that focuses on aiding humans in any decision-making process involving multiple attributes. It relates to helping decision makers: (1) formulate the problem, (2) identify the boundaries of the problem, (3) evaluate the criteria or attributes and (4) explore or assess possible solutions (Belton and Stewart, 2001). Hence, modeling techniques from MCDA can provide the agent with means to support users in multi-issue negotiations.

The **Value Based Approach** allows decision makers to associate a numerical score or value v_i to each option (a_i) and a weight w_i to level the individual scores of every issue on to a commensurate scale, in hopes of establishing a *value function* $V(a)$. The value function presents a quantifiable measurement to evaluating each offer, which permits agents to handle the uncertainties of users' desires in a single overall equation:

$$V(a) = \sum_{i=1}^{n} w_i v_i(a_i)$$

This additive model structure is theoretically sound and is easily understood by the decision maker, in terms of relating the values and weights to the possible offers generated by the function. This means that there is no complicated mathematical computation behind the scenes (Stewart, 1992). The InterNeg Group (http://interneg.org) has successfully used this model in both Inspire and Aspire to help users conceptualize negotiations.

When using MCDA methods, researchers are often confronted with the problems of helping decision makers assign weights to criteria. The **Swing Weight Method** provides a dependent scale of scoring as well as an intrinsic importance to each criterion. Users are presented with a visual scale and slider, which they move from left to right in order to assess the relative magnitude associated to the different attributes. For example, Figure 9 illustrates the swing weights that a negotiator has given to five issues concerning the purchase of a car.

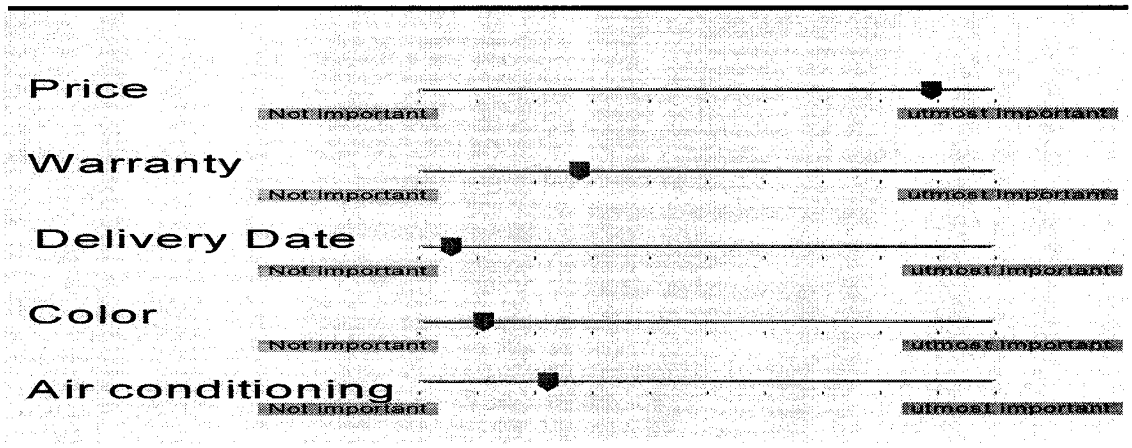


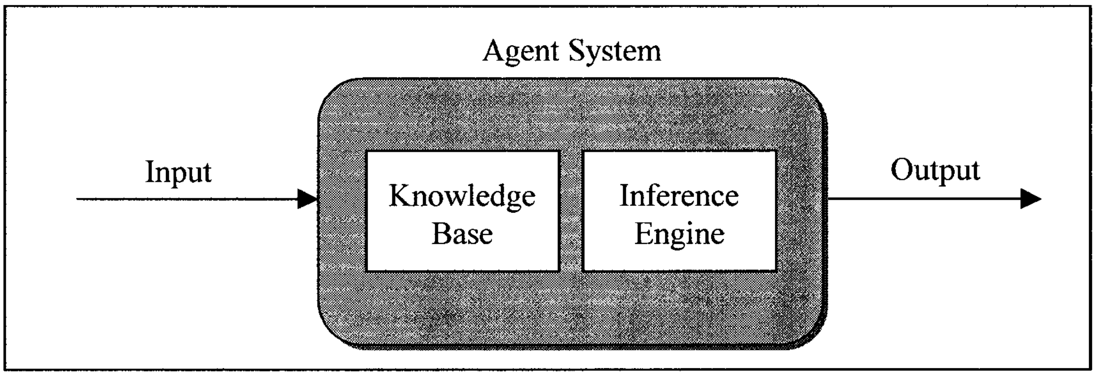
Figure 9. Swing weight method

Belton and Stewart (2001) have found that this method provides a comfortable working and visual representation for decision makers to assign weights without the need for numerical precision.

However, in order to provide timely suggestions, advice and offers, agents require a reasoning method to work in conjunction with MCDA techniques. Heuristic approaches are borrowed from the field of artificial intelligence and partnered with MCDA. Agents can then be equipped with the intelligence to handle multi-issue bargaining.

4.3.2 Knowledge-based reasoning

Knowledge-based reasoning is most often associated with expert systems that support, critique, teach or even diagnose users in a very narrow field of uncertainty. This technique is based on “if-then” rules, such that **if** an antecedent (or hypothesis) is entered **then** a consequence (or conclusion) is derived. The rules are assembled in a decision tree like structure to form a knowledge-base through which an inference engine is used to search for the final conclusion (refer to Figure 10). The conclusion is the output of the system that is derived from the input of one or more antecedents. Atin, the agent developed by the InterNeg Group to assist users in the Inspire environment, is an agent that combines utility analysis and knowledge-based reasoning to critique negotiators in their decision making processes (Kersten and Lo, 2001). It gathers user’s decisions and the negotiating criteria as inputs and provides a critique of the chosen action as an output. The critique can be presented in many forms: (1) no action by the agent when the user is performing well, (2) an indication of possible errors committed by the user or (3) suggestions for potential improvements to the user’s decision (Vahidov and Elrod, 1999). In addition, Atin supported negotiators with helpful advice when they are faced with an unfamiliar situation, such as a deadlock in negotiation or the pre-negotiation process. The disadvantage with knowledge-based reasoning is that it restricts an agent’s ability to grow or learn without any direct human involvement. Once a knowledge-base is implemented any additional rules requires a programmer to intervene.

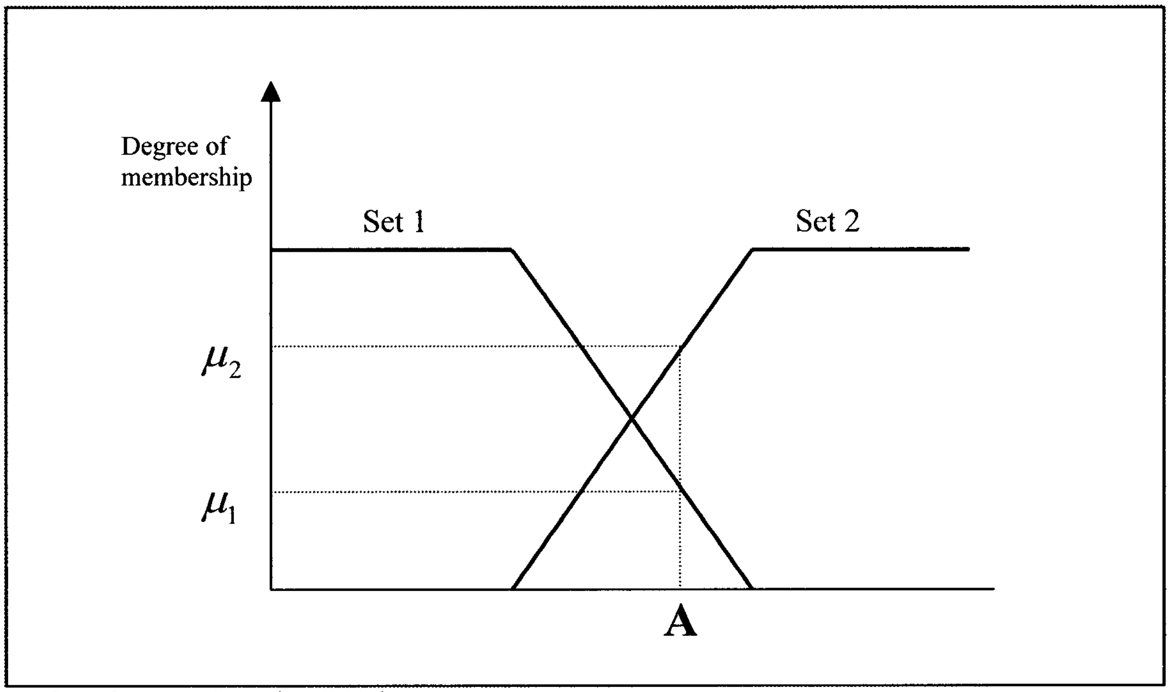


Source: Lo (2001)

Figure 10. Knowledge-based agent system

4.3.3 Fuzzy Logic

Fuzzy logic is seen as a relaxed superset of Boolean logic where concepts of partial truths are incorporated into reasoning. This means that the boundaries in traditional Boolean sets are relaxed and the degree of membership of an element in a set is expressed in a function μ. This implies that a particular entity A could be a member of two subsets with different degrees of membership $\mu_1(A)$ and $\mu_2(A)$. Figure 11 shows the membership of A in set 1 and set 2.



Adapted: Han and Kamber (2001)

Figure 11. Fuzzy logic representation

Fuzzy logic allows for imprecision to be represented in negotiation. Therefore, agents can manipulate uncertainty in the decision-making process. Wasfy and Hosni (1998) proposed the usage of fuzzy logic to model two-party multi-issue negotiation. Strategic profiles of negotiators are depicted in fuzzy logic representation to show that different concession levels are possible within a given strategy. Researchers at CSIRO in Australia bring together utility analysis and fuzzy logic to construct fully automated agents named FeNAs to negotiate on multi-issues (Alem, Kowalczk, et al: 2000). Offer evaluation and counter-offer generation are accomplished through fuzzy constraint-based reasoning during negotiation and solutions are derived as a common area between the buyer's and seller's sets. These works provide a premise to guide agents in handling decisions in various complex bargaining situations.

Although there are other heuristic techniques, like genetic algorithm and neural networks that give reasoning capabilities to agents, knowledge-based and fuzzy logic are the most appropriate algorithms in dealing with symbolic knowledge extraction in real-time environment (Azvine, Azarmi, et al; 1997). Symbolic knowledge extraction is essential to the decision-making processes in negotiation since exchanges between parties contain structural and communicative components, which cannot be modeled by mere numbers.

This literature review serves to demonstrate the importance of EM to the growth of EC and the need for negotiations between market participants. Researchers such as Kersten (2002), Maes (1998) and Negroponte (1997) have found that agents are well suited to support human decision-making. Negotiation is a complex intellectual and social activity that takes many forms (parallel, sequential, bilateral, multilateral, single-issue, multi-issue), requiring an agent with basic, adaptive and negotiation traits. In order to embody these characteristics, several techniques borrowed from Artificial Intelligence, Mathematics and MCDA show promise. The creation of an intelligent agent to support negotiations in EM is undoubtedly a step in furthering the dimensions of EC. Thus, the goal of this thesis is to build an EM with agent to support negotiations. This is devised into Section 5 where the research objective is defined, Section 6 where the methodological framework is explained, Section 7 where the system development is examines and Section 8 evaluates the system.

5 Reseach Objective

The discussion in the previous sections has no doubt revealed that electronic commerce offers tremendous opportunities for various types of businesses (Section 2.1). As more buyers and sellers transact in the marketplace, intermediary electronic markets are needed to provide an adequate unbiased platform for participants to meet and negotiate over goods and services (Section 2.2). In addition, EM must support negotiation as one of the primary business functions in order for buyers and sellers to discover mutually beneficial price mechanisms.

Negotiation has been described as either a simple or complex decision process that is characterized by many factors (i.e. multi-issue, bilateral, sequential, etc.) that may prove difficult for participants with little or no training in the process (Kersten, 2002). In order to assist novice negotiators in EC, researchers have proposed the use of agent technology to help manage information and present negotiators with knowledge that is specific to their needs (Negroponte, 1997; Mae, Guttman, et al 1998; Jennings and Faratin 2001). The negotiation agent, Atin was proposed by Lo (2001) to assist users in the Inspire environment. However, very little work has been done on creating an EM for agent supported-negotiation, therefore, this thesis aims to:

Design and implement an electronic market that allows people to negotiate, as well as develop a simple agent capable of supporting market participants in their negotiation activities.

This goal can be broken down into the following objectives:

1. Devise a framework to integrate agent technology with e-negotiation in an EM structure.

2. Construct an electronic marketplace for buyers and sellers to meet and engage in bilateral, sequential and multi-issue negotiation.

3. Develop a simple agent capable of supporting market participants in their negotiation activities.

4. Conduct a preliminary usability test of both the EM and agent supporting negotiations.

The devised framework is comprised of an agent integrated to a Web-based electronic marketplace that will allow experienced and novice participants to engage in business negotiation. Buyer and sellers will be able to register products for transaction, input issues essential to negotiation, exchange offers and messages and partake in the activities vital to the stages of negotiation. In addition, with each activity the agent will provide advice and critiques to guide the user along the decision-making process. However, the user will always have the option either to follow the agent's advice or to disagree with the suggestion and proceed independently.

The EM constructed in this thesis project is called *eAgora*, based on the word *electronic* and the Greek word *Agora*, which, coined in Mesopotamia, was the first term to mean marketplace in western civilization (Sobel, 1999). The key methodological choices required innovative approaches differing from strictly traditional information-systems practices because the experimental environment spans many computers and imposes strict requirements on the programming platform. In order to accommodate Web-based and agent technologies, the methodology consists of a cross between traditional systems development, Fusebox and Agent-Oriented approaches.

Usability testing is deployed to substantiate that eAgora meets the standards and requirements determined in the scope of this thesis.

6 Methodology

Most information systems are constructed using the System Development Life Cycle (SDLC), which consists of analysis, design and implementation[2] (Whitten and Bentley, 1998). **System analysis** is the study of business and information prerequisites as well as priorities needed to construct a new system. It is not based on any systems development technology nor concerned with programming techniques. The aim is purely to identify what is required of the system (Wetherbe, 1994; Zachman, 1987). **Systems design** is the specification of technical solutions (i.e. servers, data repositories, screens, menus, reports and other tools) to the business, negotiation and agent prerequisites identified in systems analysis (Whitten and Bentley, 1998; Rob and Coronel, 2000). **Systems implementation** is the construction of the application and its delivery from production to operation. The new system must satisfy the prerequisites established in analysis in addition to the specifications determined in design (Whitten and Bentley, 1998).

However, the SDLC of Web applications requires a methodology different from those used with traditional information systems because of its distributed environment. Not only is this thesis comprised of a Web-based system, but an agent system is also involved. In order to attend to these developments, Fusebox (Peters and Papovich, 2002), Agent-Oriented (Wooldridge, Jennings et al., 1999) in addition to traditional database development methodologies (Rob and Coronel, 2000) are implemented through the SDLC.

[2] Implementation is also referred to as development by some authors.

6.1 Traditional database development methodology

Traditional database development methodology breaks down data development into: data analysis (Section 6.1.1), where data is depicted in an entity-relationship diagram; database design (Section 6.1.2), where tables, fields and relationships are represented in a database schema; and database implementation (Section 6.1.3), where the database is coded in a database management system.

6.1.1 Data analysis

Data management is a critical aspect of any information system, making it such that data modeling is a required step in the analysis. One of the most frequently used methods of representing data is the entity-relationship diagram because it depicts data in terms of entities and relationships (Bruce, 1992). The entity is an abstract symbol for a class of persons, places, objects, events or concepts for which data is to be stored. The relationship is a business connection that exists between one or more entities. Within each relationship there is a bidirectional degree of incidence from one entity to the next. For the example where "one teacher lectures one or many students", "teacher" and "students" are entities, the relationship is "lectures" and the cardinality is "one to one or many". See Figure 12 for the entity-relationship diagram.

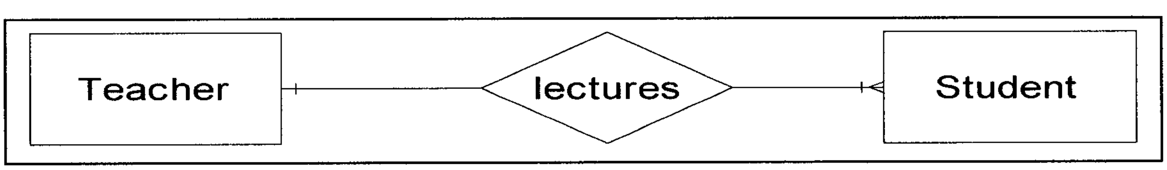


Figure 12. Entity-relationship diagram

6.1.2 Database design

Database design is concerned with representing data in a database schema, which serves as the basis for implementation (Rob and Coronel, 2000). The database schema depicts the tables and relationships necessary to create the relational database management system. It is based on the models examined in the analysis. The entities expressed in the entity-relationship diagram are used to form the tables, and the fields are derived from the data requirements specified in the analysis of processes. Based on the previous example (Figure 12), teacher may have teacher identification, teacher name and department name as fields, whereas students may have student identification, student name, student major and student average GPA as fields. This is illustrated in Figure 13. Each table also contains a primary key (PK) used as a unique identifier for each record.

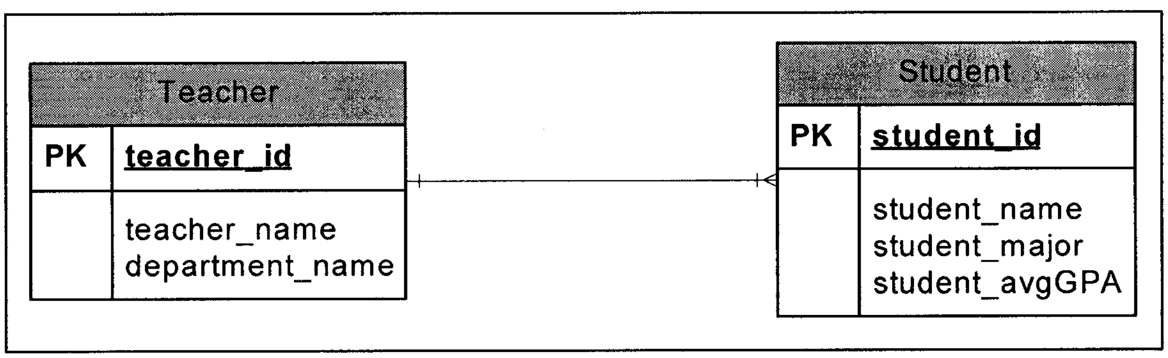


Figure 13. Database schema

6.1.3 Database implementation

The database is constructed in this part of the development process. Relational Database Management System (RDBMS) software (such as Microsoft Access, Oracle, MySQL, etc.) is used to create the tables, fields and data definition of the database, and it serves also to manage the database once the site is deployed.

6.2 Fusebox methodology

The Fusebox methodology views Web systems development in terms of components (known as fuses) that are assembled to create an entire Web page, a Fuseaction. The fuses are linked together using the Fusebox engine, metaphorically similar to the electrical fuse box in homes. Each component can function independently and can be removed from the system with only minor disturbance to the whole. The steps of Fusebox methodology according to the SDLC are the following: (1) wireframing (Section 6.2.1) and Fuseaction diagram modeling (Section 6.2.2) are used for analysis, (2) prototyping (Section 6.2.3) for design and (3) construction and coding (Section 6.2.4) as well as unit testing and deployment (Section 6.2.5) for implementation (Peters and Papovich, 2002).

6.2.1 Wireframing

Wireframing begins by the identification of key activities in the system. Based on these activities, the function and links are outlined in plain text on a Web page (Peters and Papovich, 2002). Figure 14 shows a wireframe for the display page used to capture the members' username and password as part of the login activity. Once the user provides this information, the submit button links to the checkLogin page to verify the information.

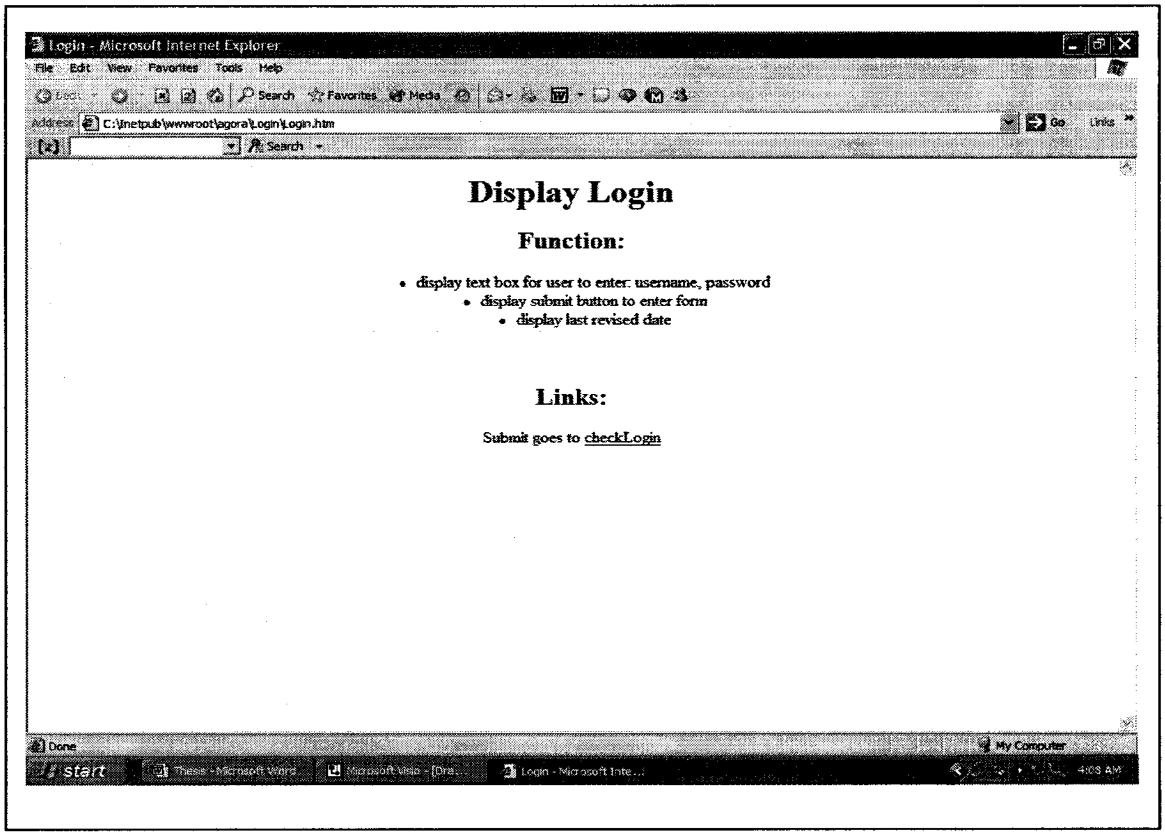


Figure 14. Wireframe

6.2.2 Fuseaction diagram modeling

Fuseaction diagrams are used to illustrate the flow of the pages. A Fuseaction diagram is composed of wireframes and the links that bind them. Fuseactions are divided into two types: form Fuseactions that display the input and output pages and action Fuseactions that manipulate data, perform queries, validate forms, etc (Fusebox, 2003). An example of a Fuseaction diagram that validates user login is shown in Figure 15. When users arrive at the site, they must enter their information in the Login page. This information is processed in the checkLogin (this action page is not displayed) to verify if they have permission to enter the site. If the information is valid, then the Welcome page is displayed, otherwise the Error page is displayed.

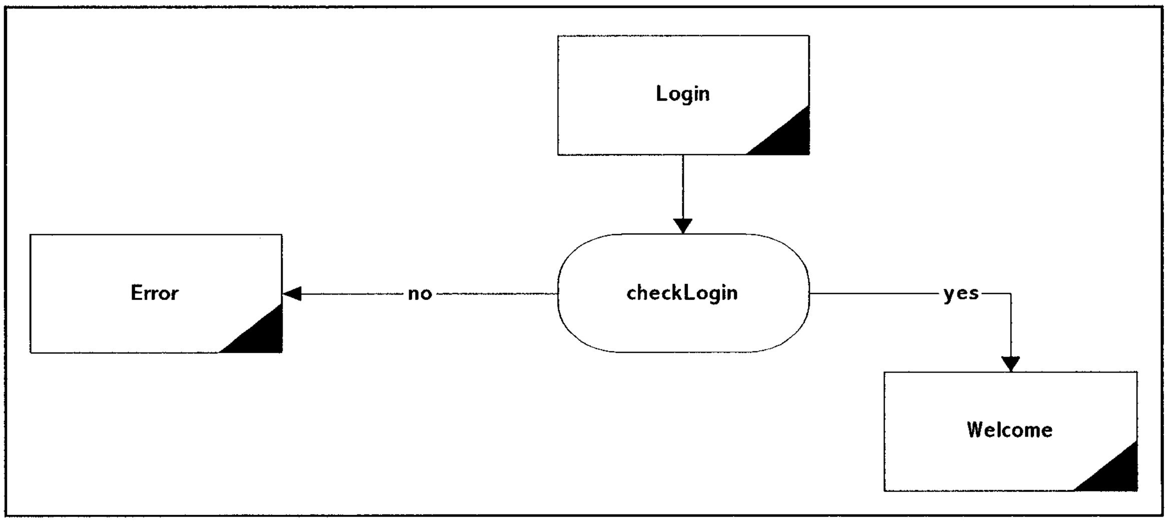


Figure 15. Fuseaction diagram for login activity

6.2.3 Prototyping

Prototyping is the creation of the HTML pages of the final application, including, colors, buttons, menus, graphics and all other aspects that appear on the forms (Peter and Papovich, 2002). The only part missing from the pages is the functionality of the Graphical User Interface (GUI) input and output controls. A prototype of the wireframe example given in Section 6.2.1 is depicted in Figure 16.

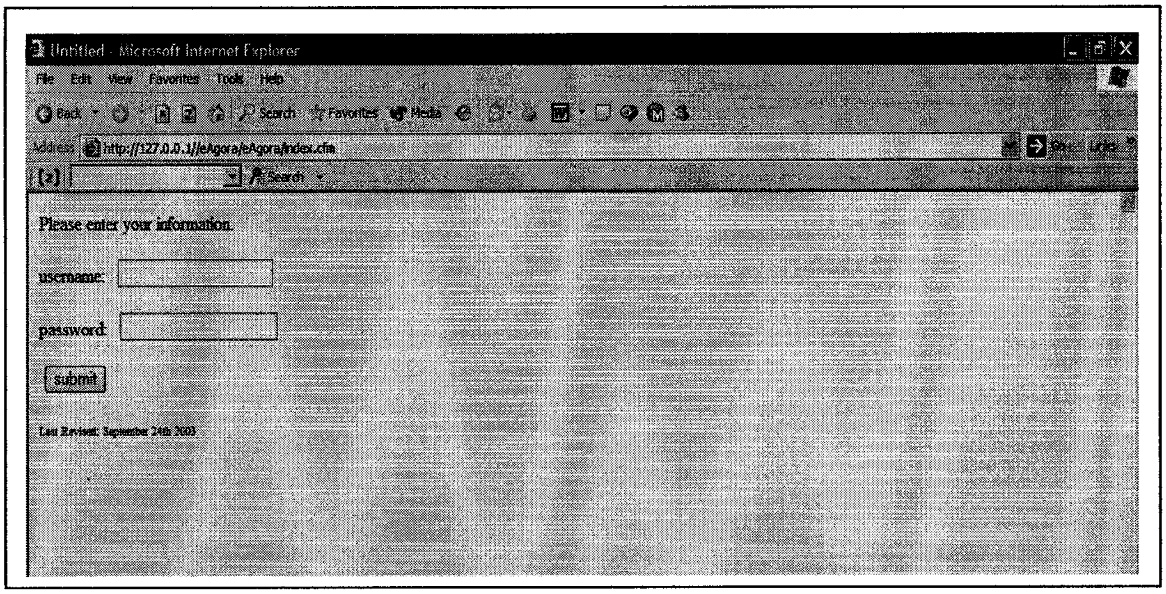


Figure 16. Snapshot of the prototype of login page

6.2.4 Coding and construction

ColdFusion language and Fusebox engine are used to construct the functionality of components for all fuses. The Fusebox engine is downloaded from Fusebox.org and installed on the server (Fusebox, 2003), after which modifications are required in the circuit, layout, setting and switch files.

The circuit file determines all circuits between Fuseboxes in the system. This involves stating the names of the Fusebox and their physical location in the system. Each Fusebox is basically a folder that contains all the fuses necessary to perform the activities demanded of the box. In order to pass from one Fusebox to another, the program just calls the name of the targeted box. This permits security functions to be applied to designated groups of folders.

The switch files are used to manage the Fuseactions in the folder such that each one folder must have its own switch file. By using the conditional switch tag in ColdFusion, fuses are assembled into a Fuseaction that the user sees as the Web page. From a physical point of view, a fuse is not necessarily a Web page, as it can be the whole or only part of a page. The advantage of this organization is that a fuse can be employed in several Fuseactions in the same box. Figure 17 explains the Fusebox decomposition.

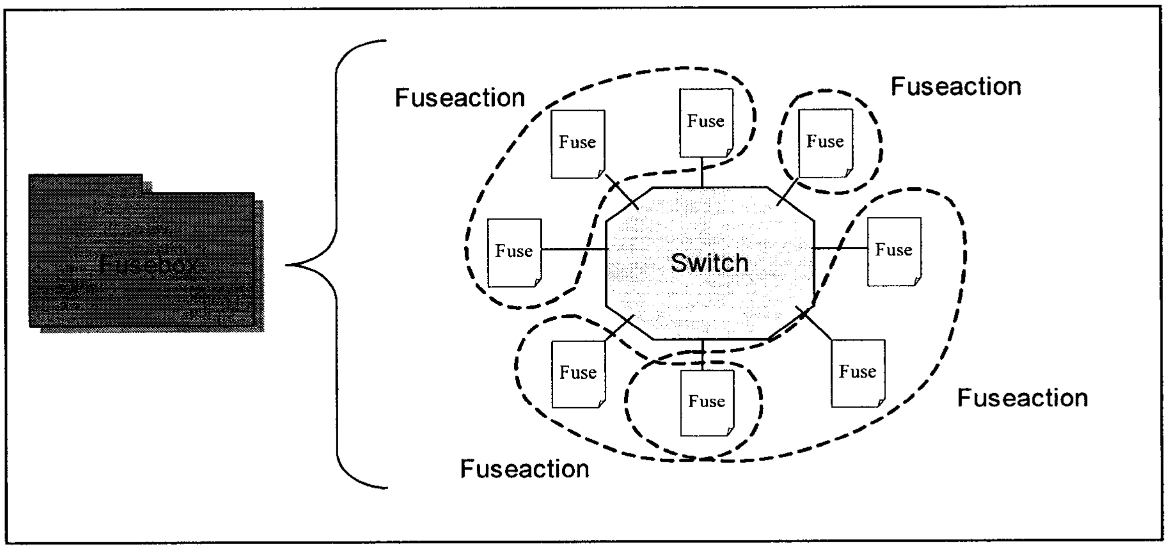


Figure 17. Fusebox decomposition

The setting file is optional in the children Fuseboxes, but required in the main box. It handles security and inheritance matters.

The layout file is optional in all circuits. However, this file is utilized to manage the outlook of the site by assembling the page components for presentation. The file provides a background, global navigation links and the screen positioning of an application.

The next step is the prototype dissection. The prototype pages created in the design phase are examined for Fuseactions and eXit FuseActions (XFA) (Peter and Papovich, 2002).

The functionalities required in a page are identified and categorized into Fuseactions. These Fuseactions are further divided into fuses that perform only one specific task. This task can be either: display HTML code (dsp), query the database (qry), layout the presentation (lay) or act on a variable (act). An example of Fuseaction and XFA identification is found in Figure 18.

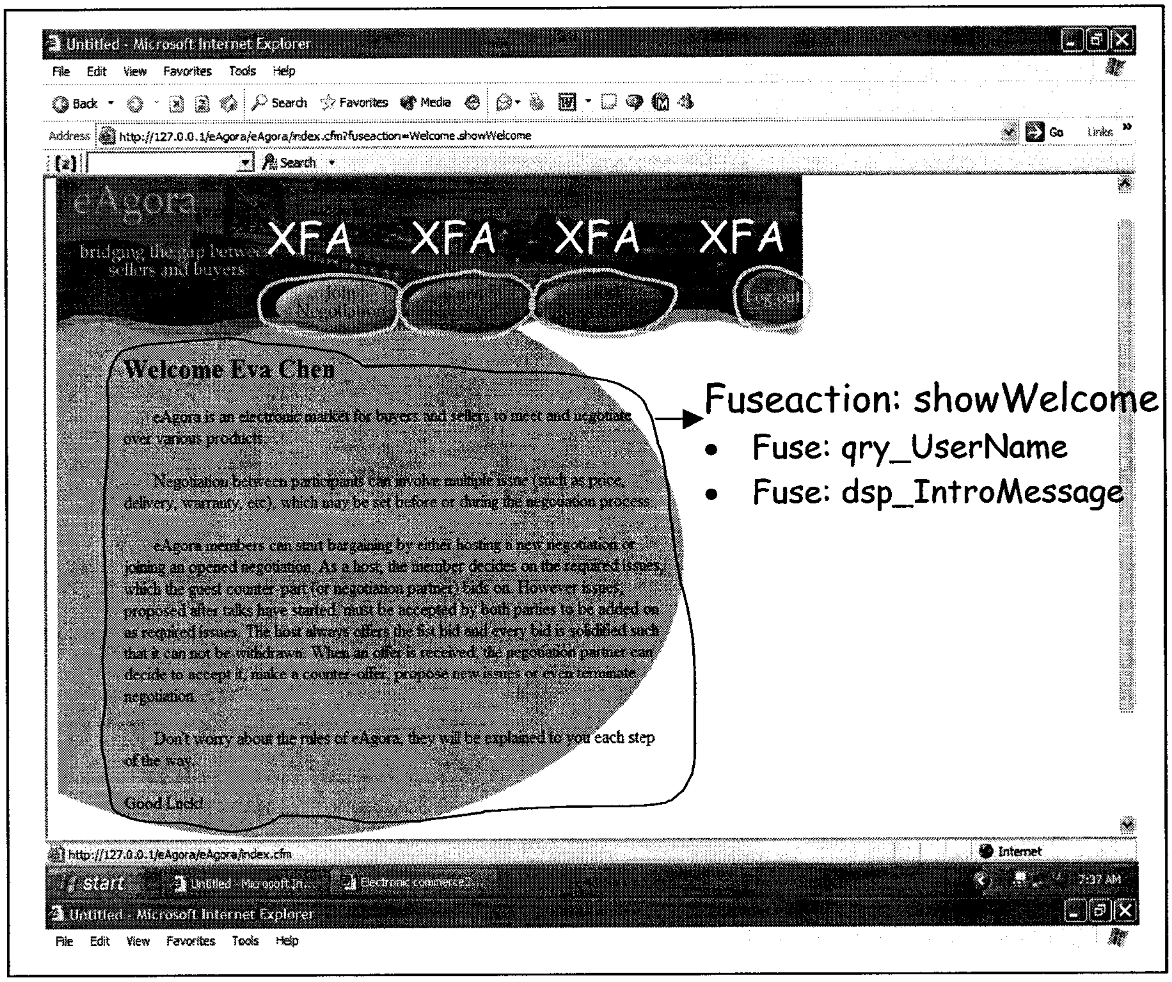


Figure 18. Fuseaction and XFA identification

An XFA is a variable used to store either a hyperlink or a functional link. The use of a variable instead of a hard-wire value allows fuses and Fuseactions to be carried anywhere in the application. Circuits become more flexible, modular and, most significantly, portable.

After the setup of the Fusebox environment and the dissection of the prototype, coding starts in ColdFusion language for each page of the site.

6.2.5 Unit testing and deployment

Unit testing consists of verifying the functionality of every page at the moment it is coded. Then testing proceeds to a group of pages that constitute an activity in the system. Once the entire application is verified to work properly, it is moved from the development server to the deployment server where it operates (Peter and Papovich 2002).

6.3 Agent-Oriented methodology

In Agent-Oriented methodology, the methods are similar to Object-Oriented methodology, except that the agent is defined beyond the scope of an object. It has an agent model (6.3.1) as well as a service model (6.3.2) and an acquaintance model (Wooldridge, Jennings et al., 1999). The acquaintance model refers to the communication links between agents. This applies to multi-agent system, which is not in the scope of this thesis.

6.3.1 Agent model

Agent analysis is based on Use Case Modeling to illustrate the agent's role in helping users in the system. By identifying the functions associated with the agent, Use Case Modeling is able to show the interaction between agent and user in a series of steps of a subsystem. Subsystems are negotiation activities that require agent assistance.

In object-oriented analysis, Use Case Modeling is one of the more accepted and successful methods for finding and identifying objects (Martin and Odell, 1992). It breaks down the entire system-functionality into many smaller statements, called **use case** from the perspective of external users, namely **actors** (Ivar et al, 1992; Lo, 2001). Figure 19 shows the subsystem for determining the age status of a user. The agent starts by asking the age of the user. Once the age is entered, the agent retrieves it, the knowledge-base is searched for the status and the result is displayed.

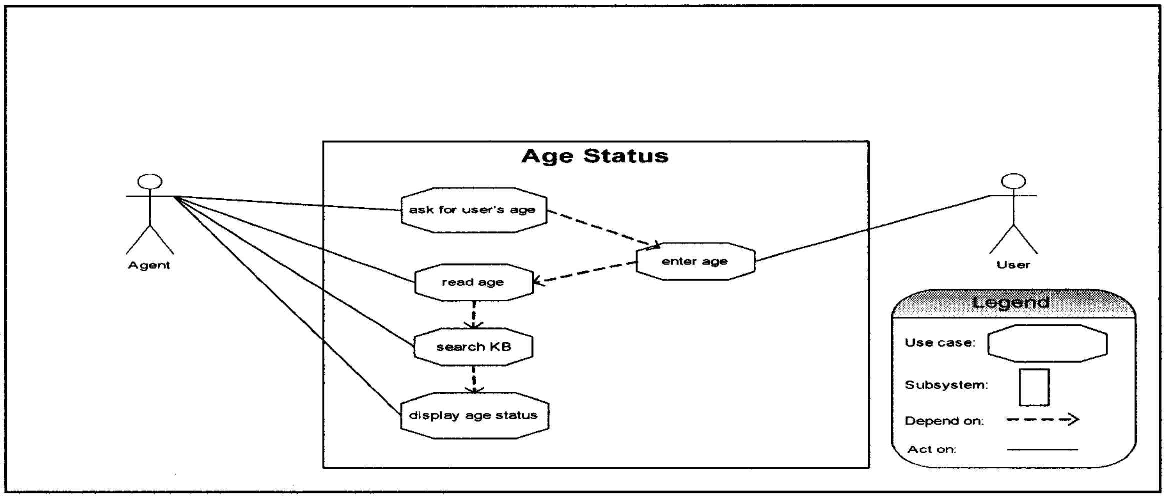


Figure 19. Use Case Model of age status

6.3.2 Service Model

The service model is essentially the knowledge-base that is made up of rational rules used by the agent to reason. Decision trees serve to design the knowledge-base. A decision tree is illustrated by a flowchart tree-like structure of nodes, each of which represents a condition. The branches in the tree denote the outcomes of the condition (Han and Kamber, 2001). As an example, Figure 20 depicts the decision tree employed by an agent to determine whether a user is considered an adult.

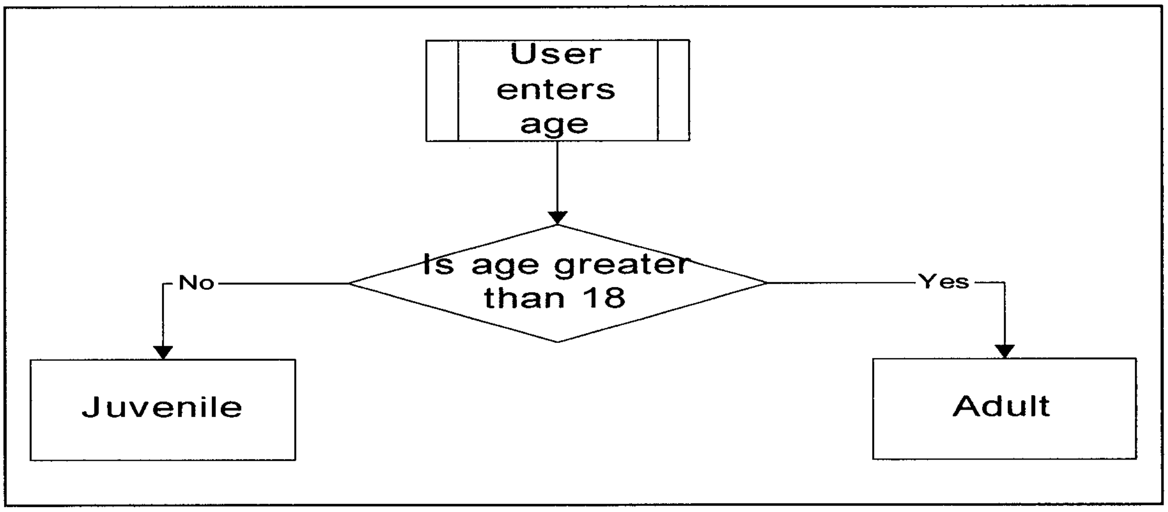


Figure 20. Decision tree to determine adulthood

The service model characterizes the design stage of the agent. The implementation of the agent can be in any object friendly Web-based language that complements the rest of the site. For example, if the site is programmed in JavaScript, then the agent can be also created in JavaScript or any other JavaScript based language (DecisionScript, JScript, etc).

6.4 Framework of merging methodologies

The framework for constructing a dynamic Web-based system that contains business, negotiation and agent activities requires the merging of the three methodologies to the SDLC. This framework focuses on three building blocks: (1) data, the raw material treated in the system, (2) EM, the business and negotiation activities and (3) agent, the intelligent entity that assists negotiators. These building blocks are treated by the three different methodologies. The overall representation of the framework and deliverables are shown in Figure 21.

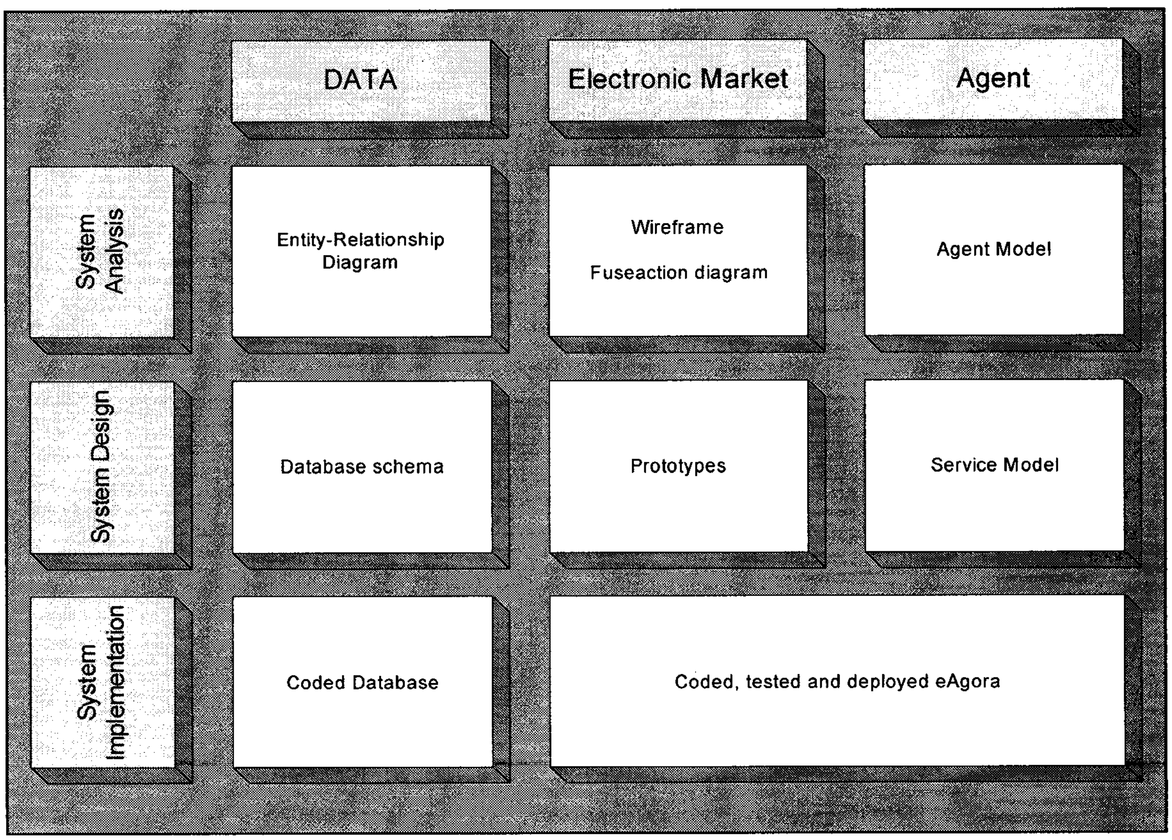


Figure 21. Methodological framework and deliverables

7 eAgora

The development of eAgora is based on the framework described in Section 6.4. Each building block is integrated to every stage of the SDLC. The analysis is explained in Section 7.1 to show the prerequisites of the system. The design stage refers to identifying solutions to the requirements (Section 7.2). The implementation of these solutions is presented in Section 7.3.

7.1 Analysis

For eAgora, the analysis consists in finding the requirements and functionalities for the data, EM and agent building blocks. Data analysis focuses on modeling data that is shared by the many pages on the site. In fact, it is the most emphasized stage of most system information methodologies, since data must be organized into a flexible and adaptive manner for all present and future system needs. The modeling technique is entity-relationship diagramming, which allows data to be represented in distinguishable groups that are connected in a logical context (Bruce, 1992).

EM analysis centers on identifying the business and negotiation activities in addition to the processes required to carry out these activities. A business function diagram is used to represent the various activities. Based on these activities, negotiation protocols are established to show the rule-based flow of negotiation activities. As for process modeling, Fusebox methodology employs wireframes and Fuseaction diagrams to represent processes as pages containing functionalities and links. This approach focuses the analyst's attention on the user's perspective of the Website, in terms of Web pages (Peter and Papovich, 2002).

Agent analysis is focused on Use Case Modeling to depict the agent's role in helping users negotiate on eAgora. Use Case Modeling emphasizes the interaction between agent and negotiator in a series of steps in a subsystem (Martin and Odell, 1992; Lo, 2001).

System analysis is an important part of any methodology because it serves as the basis for building information systems. A proper analysis can reduce design, implementation and maintenance time (Whitten and Bentley, 1998).

7.1.1 Data model

The entity-relationship diagram for eAgora focuses on merely identifying the basic data components required to support e-negotiations in the EM setting. The entities consist of user, buyer, seller, negotiation issues and temporary issues. The overall representation is described in Figure 22. The users are individuals with access to the site, who may become either a buyer or seller engaging in zero, one or many negotiations. Negotiations contain all data related to the product and negotiation activities. It may have zero, one or many issues precluding price because price is a necessary part of any business transaction on eAgora. Issues are items discussed as part of the offer exchange process. They can be delivery date, warranty, color, etc. During the process of exchange, buyers and sellers can suggest zero, one or many additional issues to be added to the negotiation. These are called temporary issues, since they must be agreed upon by the counter-part in order to be made an issue.

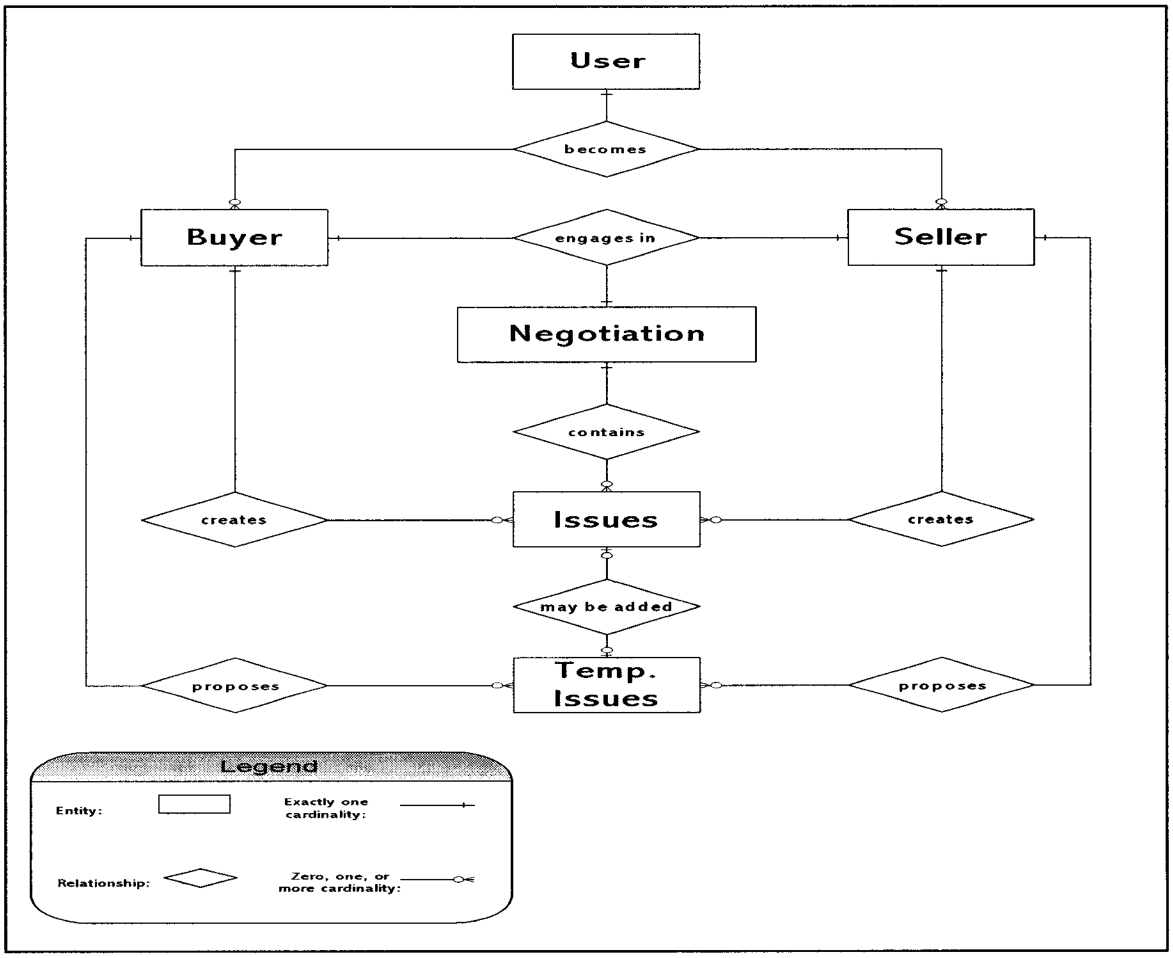


Figure 22. Entity-relationship diagram for eAgora

7.1.2 Electronic market model

The EM modeling begins by identifying activates necessary to achieve a successful negotiation described in the stages of business negotiation (Section 3.3). The activities are grouped into a business function diagram to show the overall components of the EM. Using these activates, the negotiation protocol is established to depict the rules of engagement for eAgora. For example, when a buyer makes an offer, he or she must wait for the seller to act before another offer can be made. Once the protocol is instituted, wireframes and Fuseaction diagrams are drawn to show the detailed logic of the site under Fusebox methodology.

The business function diagram represents ongoing activities that contain common processes necessary to support the Website. eAgora is composed of five business activities (Welcome, Login, Host, Current and Phase) and thirty negotiation activities (fifteen for the buyer and fifteen for the seller). See Figure 23. The business activities allow the user to mange their negotiations, whereas negotiation activities represent the procedures that occur during negotiation. **Welcome** is the main menu for users to access the different business activities in the system. **Login** permits users to enter eAgora by submitting a password and user identification. **Host** lets users start a new negotiation that is opened for others to join. **Join** allows users to enter in a new negotiation hosted by another. **Current** lists all negotiations a user is engaged in as well as the negotiation status. **Phase** controls the negotiation activities available to the user, meaning that it administers the protocols. The descriptions of the negotiation activities are shown in Table 2.

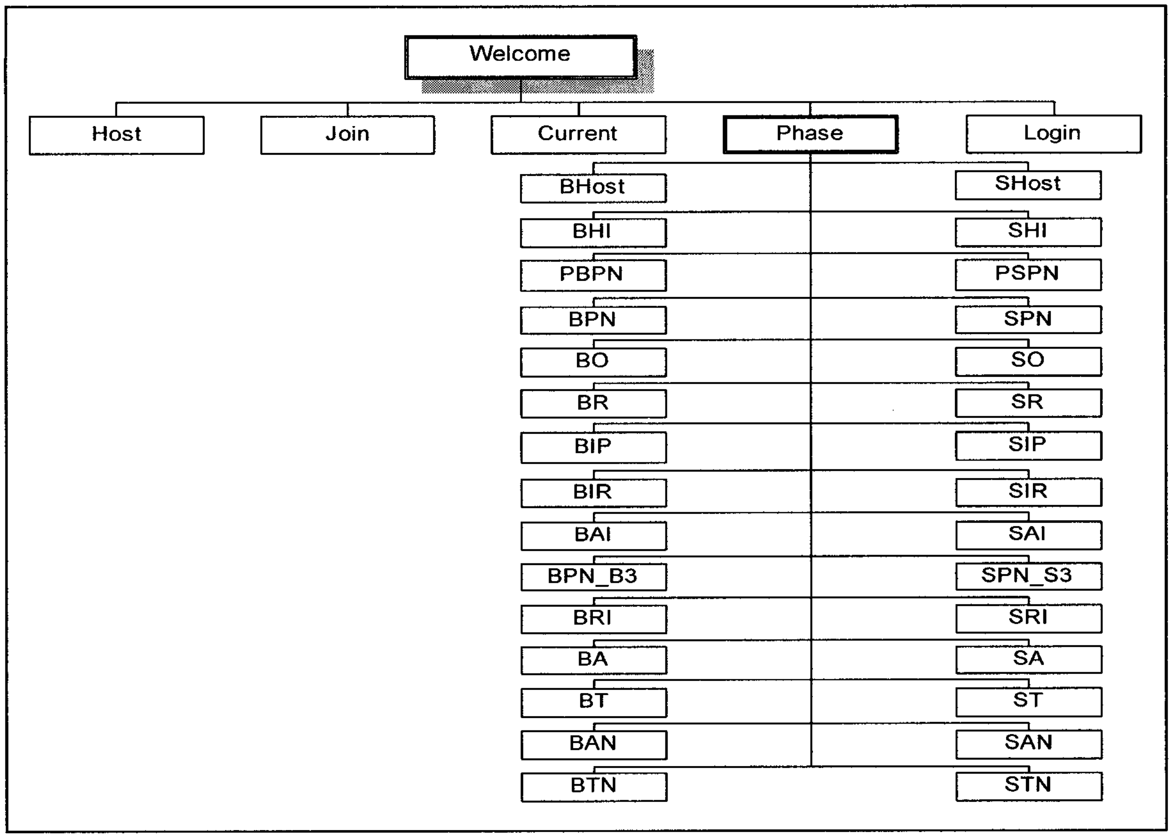


Figure 23. Business function diagram

Buyer	Seller	Description
BHost	SHost	User completes host form with product and negotiation information
BHI	SHI	Host sets up issues and options and rates them when using an agent
PBPN	PSPN	Adversary joins negotiation and accepts issues and options set by host
BPN	SPN	Adversary rates issues and options if using agent
BO	SO	Negotiator makes offer to counter-part
BR	SR	Negotiator receives offer by counter-part
BIP	SIP	Negotiator proposes new issues
BIR	SIR	Negotiator receives proposed issues
BAI	SAI	Negotiator accepts proposed issues to be added to negotiation
BPN_B3	SPN_S3	Negotiator re-evaluates issue and option ratings due to new issues
BRI	SRI	Negotiator rejects proposed issues
BA	SA	Negotiator accepts offer
BT	ST	Negotiator terminates exchange
BAN	SAN	Negotiator signs off because offer has been accepted
BTN	STN	Negotiator signs off because adversary has terminated negotiation

Table 2. Description of negotiation activities

Once negotiation activities are established, two strategic protocols for eAgora are determined based on suggestions by Norman (1994), Holsapple (1996) Jennings and Faratin (2001), Oliveira (1999), Ströbel (2001) and Winoto (2002) in Section 4.2.3. Two sets of rules are needed because either a buyer or a seller can host negotiations, therefore requiring two different initial sequences of activities. Figure 24 shows the eAgora protocol for the case when the buyer is the host of the negotiation. The seller-host communication pattern is found in Appendix A.

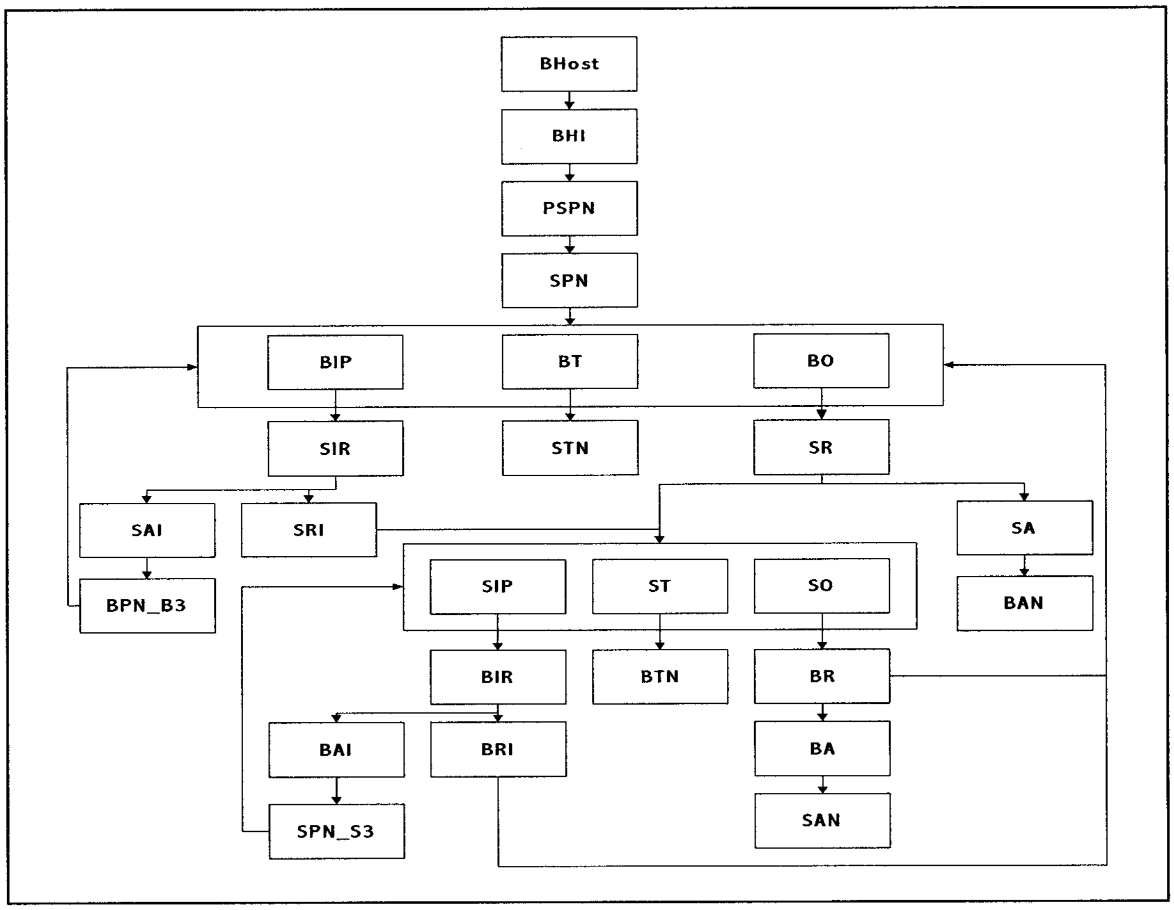


Figure 24. Buyer negotiation protocol

In order to define the requirements and functionality of eAgora, wireframing and Fuseaction diagrams are used to identify the processes involved with each activity. A wireframe is a plain-text depiction of the functionality involved in each Fuseaction, also known as a page or template of the Website. For example, Figure 25 shows that the Welcome page serves as the home page where the user names are displayed along with an introduction to eAgora. It also provides links to other fuses such as Current, Host and Join. Wireframes are simple representations that preclude any technical terms and programming codes. This method aims to help the analyst discover the business processes and not the Website design.

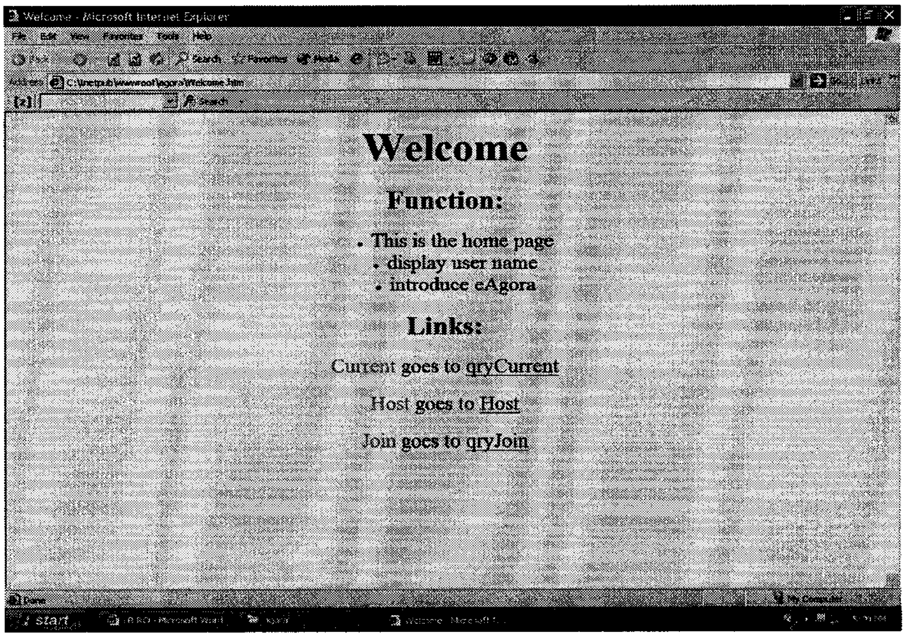


Figure 25. Wireframe for Welcome page

In addition to wireframing, Fuseaction diagrams are also part of the analysis. Each diagram is formed from of wireframes and the links that bind them. Fuseactions are separated into two types: form and action pages. Figure 26 is an example of a diagram that shows the processes necessary for the five business activities.

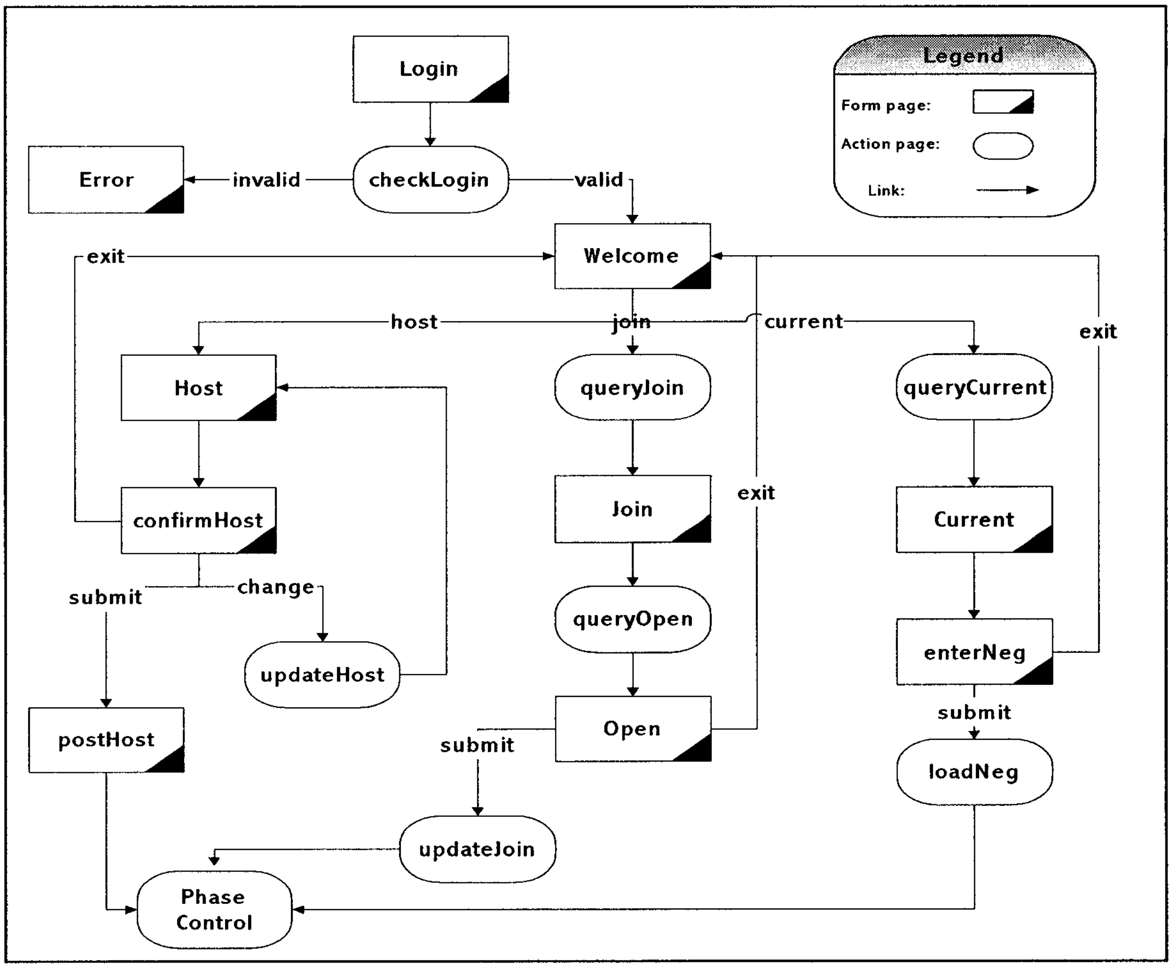


Figure 26. Fuseaction diagram of business activities

The **Login** activity allows only users with access permission to enter eAgora, and it necessitates two form pages (Login and error) and one action page (checkLogin). The first form page, Login, asks for user identification and password. checkLogin then verifies whether the information provided is valid. An invalid input leads users to the Error page, but a valid entry directs them to Welcome.

Welcome activity only has one Fuseaction, a display page called Welcome, where the user may navigate to the three other business activities: **Current, Host** and **Join**.

Host is comprised of three form pages (Host, confirmHost and postHost) and one action page (updateHost). The Host page is a form for users to enter information necessary to commence a new negotiation. The input is re-displayed for verification in confirmHost. If the information needs to be changed, updateHost takes it and situates it back on the Host page for editing. However, if the user proceeds to submit in confirmHost, then postHost tells the user that a negotiation is opened for someone to sign up and start negotiation activities based on the information posted.

When a participant chooses to join a new negotiation from Welcome, a query is made to the database for opened negotiations (queryJoin), and the search results are shown in Join. The user may select to view a specific negotiation, which requires a query for the detail of this specified negotiation (queryOpen) and its display in Open. After examining the details, the user may decide to enter into negotiation (updateJoin) that calls for the database to be updated with the user as the counter-part. Once the counter-part is declared, the participant may enter into negotiation activities. All three actions (queryJoin, queryOpen and updateJoin) and two forms (Join and Open) make up the processes for the **Join** activity.

Current is first composed of a query page to retrieve all negotiation that the user is involved in (queryCurrent), and then the results are displayed in the Current page. The participant may decide to view an explicit negotiation, which is shown in enterNeg. If the user decides to further engage in the activities of that negotiation, loadNeg retrieves the status of the negotiation and allows the user to enter into negotiation activities depending on the protocol. For all other Fuseaction diagrams in the EM analysis see Appendix B.

Fuseaction diagram modeling and wireframing are different from the traditional data flow diagram modeling because they broaden the analyst's perspective, from simply identifying process and information flow to a Web outlook of what needs to be manipulated and displayed from page-to-page. This reduces the design and development time by having a clear understanding of the requirements and functionalities for every page.

7.1.3 Agent model

In this thesis, Use Case Modeling is employed to represent the agent model. The actors are the negotiators (either buyer or seller) and agent on the Website. The use cases are the functionalities related to the behavioral series of steps necessary to complete a single negotiation activity. Kersten and Lo (2001) have found that negotiators often need assistance in these activities: (1) Pre-negotiation, where the agent helps negotiators structure the problem by eliciting their preferences, constructing a utility function and critiquing erroneous decisions, (2) Offer formulation, where the agent provides timely advice regarding possible counter-offers, concessions and tactics, (3) Reception of the counter-offer, where the agent critiques the adversary's offer and suggest possible actions and (4) Offer acceptance, where the agent critiques the negotiator's decision to accept an offer.

Thus the agent model in Agent-Oriented methodology is simply the representation of these activities in subsystems that depict the agent's role in eAgora. Figure 27 presents the use cases in the **Pre-negotiation** activity. The agent first retrieves the negotiation information (product detail, issues information, etc.) from the database. It then asks the negotiator for pre-negotiation settings (PNS). PNS are value ratings, reservation levels and strategy that the negotiator sets concerning price, issues and options. After the negotiator submits the values for the PNS, the agent takes these values, updates the database, searches the knowledge-base and checks if there are any appropriate warnings to provide. If this condition is met (need for warning), the agent displays the cautionary message. The warnings are generally prompted by violations that the user makes in rating the PNS. The definition of the proper PNS values is important for structuring the negotiation problem.

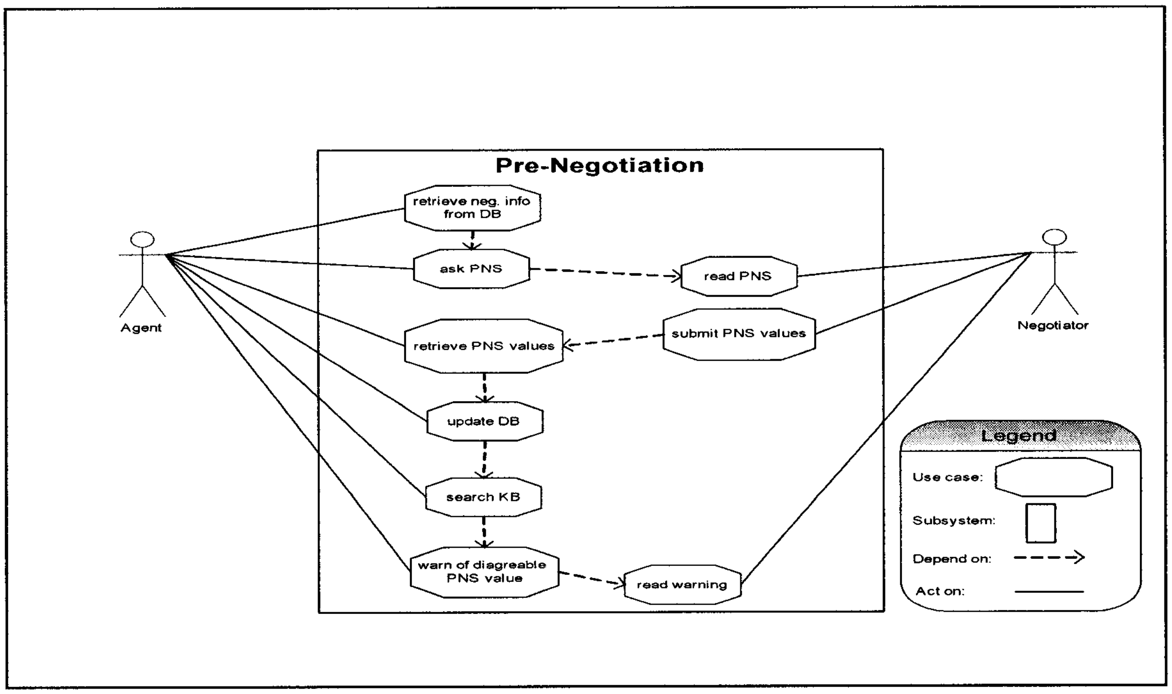


Figure 27. Use Case Model for pre-negotiation

The **Offer formulation** requires the agent to retrieve negotiation information from the database, and then the knowledge-base is searched for possible offer packages to suggest to the negotiator. The negotiator may pick one of the suggestions given or propose his (her) own offer package. The agent reads the offer, updates the database, searches the knowledge base, critiques the offer and sees if it is appropriate for the negotiator. If the proposal violates the limits or strategies determined in Pre-negotiation, then the agent warns the user. Offer formulation is depicted in Figure 28.

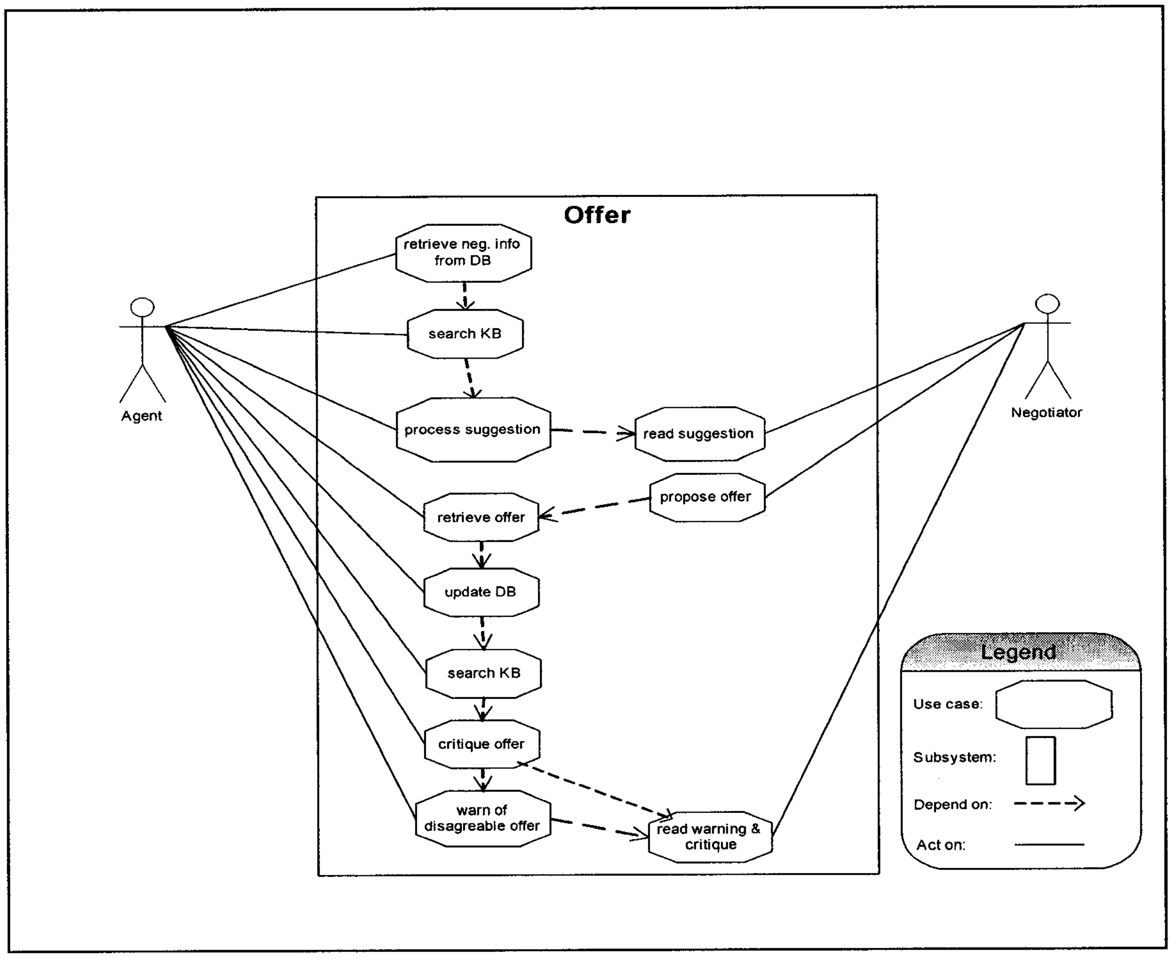


Figure 28. Use Case Model of offer formulation

Figure 29 represents the **Reception of the counter-offer** by the negotiator. The agent starts by reading the counter-offer made by the opponent. It then searches the knowledge-base for the appropriate advice, critiques the counter-offer and provides the negotiator with possible actions to take.

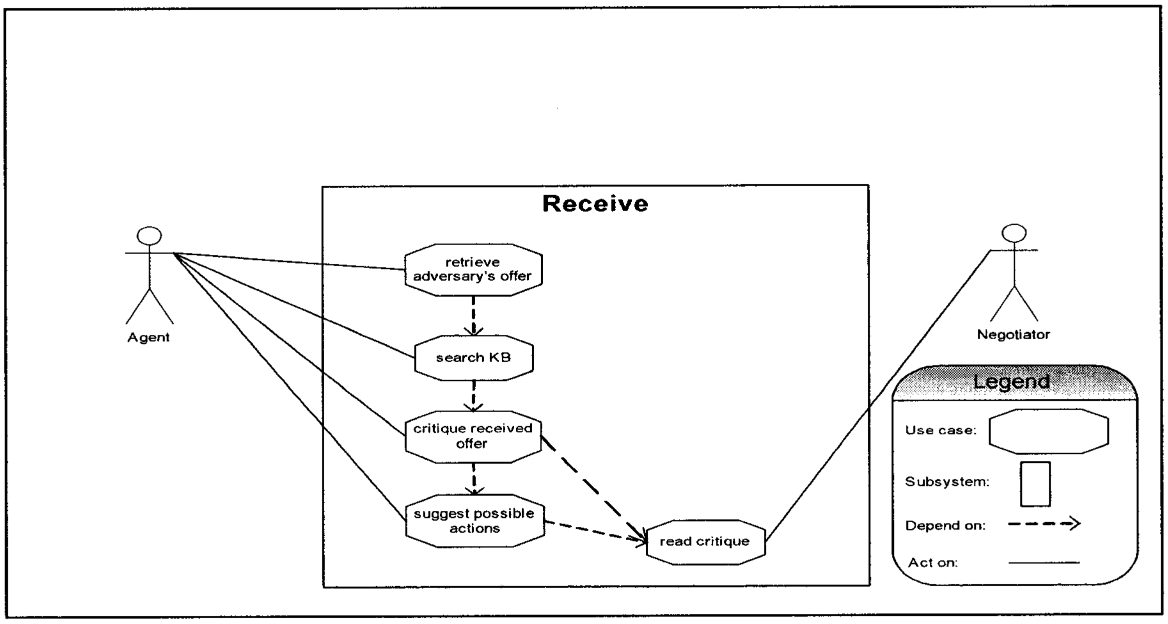


Figure 29. Use Case Model of counter-offer reception

The **Offer acceptance** is a critical stage in negotiation. In order to prevent users from agreeing to an unfavorable deal, the agent is needed to monitor actions in this stage. The agent must caution negotiators if they decide to accept an unfavorable deal. This is first triggered by the negotiator's action to accept an offer. The agent then retrieves this counter-offer and a search is performed in the knowledge-base for any necessary warnings. If the offer conflicts with the rules of the knowledge-base, then the agent shows a warning message. See Figure 30 for these events.

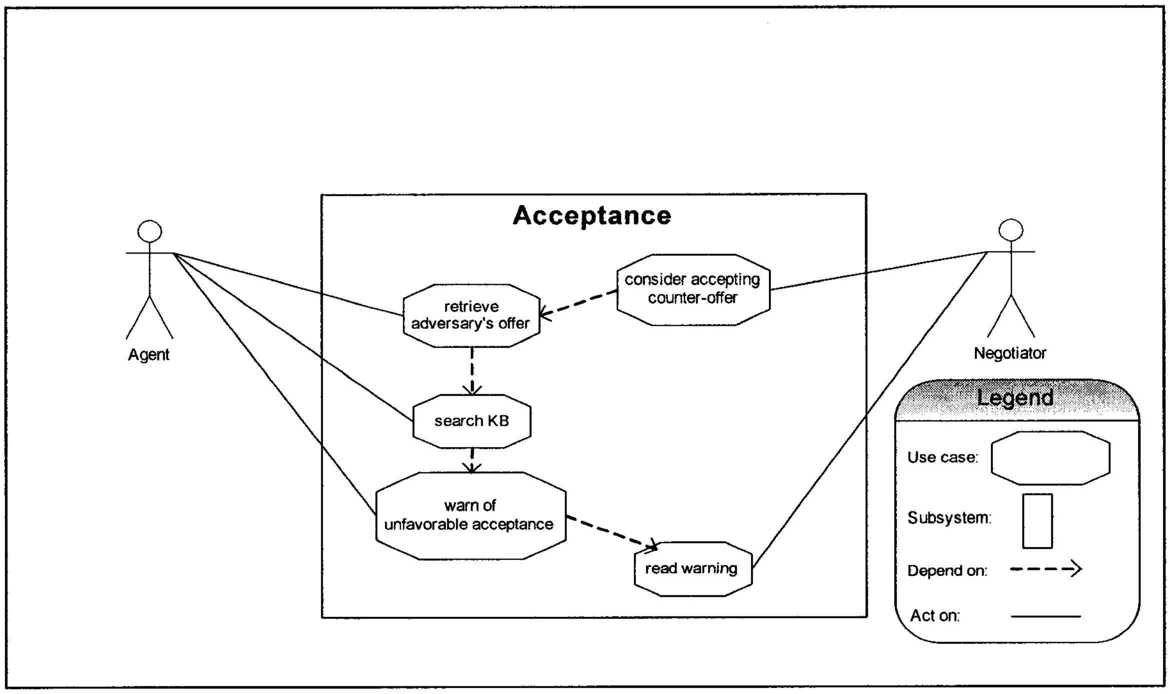


Figure 30. Use Case Model of offer acceptance

Once the agent analysis is completed, additional wireframes and Fuseaction diagrams are needed to describe the pages used to handle the agent's input and output screens (see Appendix B for all Fuseaction diagrams).

The analysis of eAgora was a lengthy endeavor that required many back and forth examinations of requirements and functionalities. However, the result sets the essential outline for the creation of the site.

7.2 Design

Each building block of eAgora requires specific designs that act as the blueprints for systems implementation. The database design involves using a database schema to illustrate the relationships, tables and fields necessary to structure data (Section 7.2.1). Prototyping is the specification of the display screens viewed by the user for both the EM and agent Fuseaction (Section 7.2.2). The knowledge base is the agent's intelligence and it is created from decision trees and fuzzy logic graphics (Section 7.2.3).

Before any designing is to commence, the systems architecture needs to be defined. eAgora consists of a multi-tiered architecture comprised of four parts: client tier, Web server, application server and database server (Rob and Coronel, 2000; Hewitt, 2002). This is depicted in Figure 31.

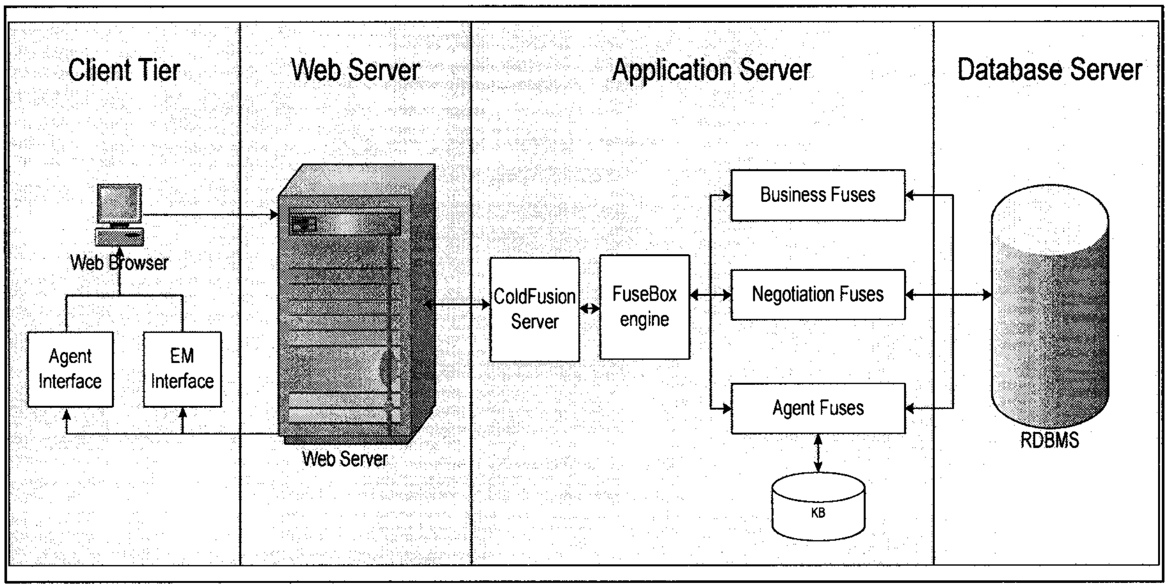


Figure 31. Systems architecture for eAgora

The **client tier** represents not only the client's Web browser, but also the interfaces that interact with the user. Both the EM and agent have forms, which are pages that display input and output screens.

The **Web server** in this case is a Microsoft IIS (Internet Information Services) 4.0 running in Windows NT. The server handles all the HTTP (HyperText Transfer Protocol) pages requested by the user. When a script page is asked for, it passes this page to the application server for processing, and receives HTML (HyperText Markup Language) formatted pages in return that are then sent to the client.

The **application server** is mainly a ColdFusion 4 server that is linked to the database, knowledge-base and Fusebox engine. ColdFusion is based on a server-side markup language known as CFML (ColdFusion Markup Language), which is used to create ColdFusion application pages called scripts. A script contains a combination of HTML and JavaScript code that allows access to the database from the Web front end. Connection to the database server is performed through ODBC (Open DataBase Connectivity).

The Fusebox engine is a code management system that controls the flow of the application. It organizes code into components that are called fuse and it joins these fuses together by a circuit to create a Fusebox. This engine also allows for multiple Fuseboxes to be connected in a circuit to build one big Fusebox system. The description and download of the engine's core files (Fusebox 3) are available at http://fusebox.org. Further explanation on Fusebox can be found in Section 6.2.4.

The agent's brainpower is in fact the knowledge-base, which is evolved by agent fuses. It consists of rules that govern the agent's behavior. One important design decision is to not use an inference engine to run eAgora's knowledge-base because most of the intelligence requires simple rules, which can be easily executed by the Fusebox engine through agent fuses. Another reason is the extra run-time needed when employing an inference engine. If the knowledge-base is to expand, a Web-based inference engine can be effortlessly added to the system, since the knowledge-base is designed separately from the rest of the agent's functions.

The **database server** consists of a RDBMS that is responsible for storing data. This system contains data in tables, which are related to one another through unique identifiers. RDBMS is a popular choice for most Web-based systems because the relationships between the tables are imposed only by the developer and not by constraints defined in the database, as in flat file database systems. The RDBMS for eAgora is Microsoft Access 2000.

7.2.1 Database design

The database schema for eAgora is drawn with the tables and relationships necessary to create the relational database management system. It is based on the data, EM and agent models explained in the analysis. The entities expressed in the entity-relationship diagram are used to form the tables, and the fields are derived from the data requirements specified in the wireframing process. For example, the "issue" table is generated from the "issue" entity and the fields (such as issue name, description and options) are obtained from the wireframe describing the form a user fills out during negotiation setup. Moreover, from the agent model additional tables were included to the schema to store information related to the agent's activities. The relationships are the same as those characterized in the entity-relationship diagram.

Figure 32 is the resulting database schema, which uses fourteen tables to manage the data required for eAgora. Every table contains a primary key (PK) that provides each record with unique identifier and some have foreign keys (FK) to connect records within different tables.

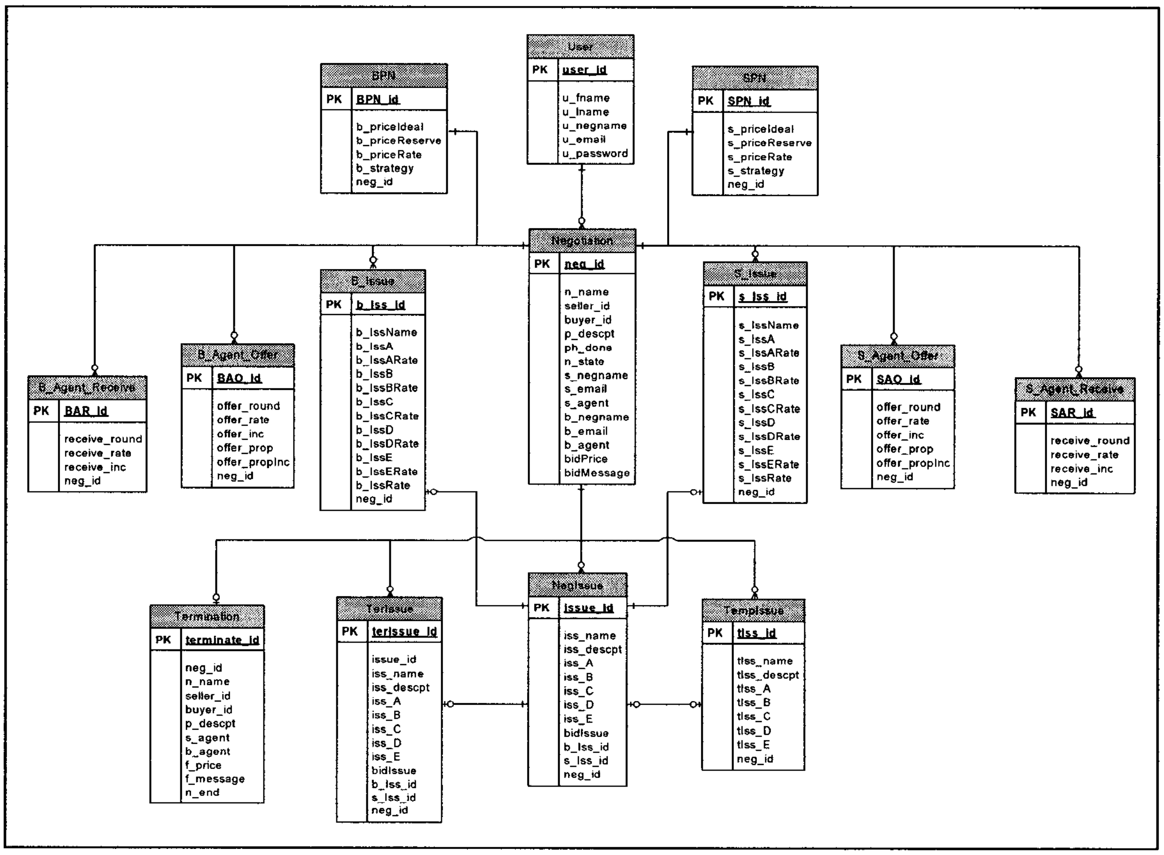


Figure 32. Database schema for eAgora

The **User** table contains information on all participants in the system. The user log on with their u_negname and u_password. The PK is user_id, which serves both as a unique identifier and to recognize the negotiations involving the user. Other information such as user's first name, last name and email address are also recorded as u_fname, u_lname and u_email, respectively.

The **Negotiation** table stores product and negotiation information. In addition the one-to-one relationship between the seller, buyer and negotiation entities are normalized to this one table. Its PK is neg_id, which uniquely identifies the various negotiations in the system. The negotiation name and product description are contained in n_name and p_descrpt, respectively. The seller (seller_id, s_negname, s_email) and buyer (buyer_id, b_negname, b_email) fields are derived from the negotiator's user_id, u_negname, u_email. The s_agent and b_agent are binary fields that are either "enabled" or "disabled" depending on whether the negotiator employs the agent or not. The negotiation activity is marked by ph_done, which describes the last activity completed in negotiation. The negotiation status (n_state) is either "1" or "0", meaning that negotiations may be either "opened" or "closed" for others to join. The bidPrice and bidMessage are fields used to record the offer price and offer message.

NegIssue is used to hold data related to the issues. There is a PK (issue_id) and three FK (s_Iss_id, b_Iss_id and neg_id). The FKs are used to link this table to that of B_Issue, S_Issue and Negotiation, which in turn connects the records from this table to the others. The issue name and description are stored in iss_name and iss_descpt. Each issue is allowed a minimum of two and a maximum of five options that are stored in iss_A (for the first option), iss_B (for the second), iss_C (for the third), iss_D (for the forth) and iss_E (for the fifth). For example, the issue delivery date may have three choices such as: two days, one week or two weeks. These options are registered per se; two days in iss_A, one week in iss_B and two weeks in iss_C. This classification allows the agent and negotiator to structurally analyze offer packages that may contain many issues with many options. The field bidIssue contains the option, which is selected as part of the offer package being proposed.

The table **TempIssue** registers information about proposed issues that a negotiator makes to the counter-part. tIss_id is the PK used to identify the record, and neg_id is the FK connecting the record to the appropriate negotiation. The proposed issue name, description and options are stored in tIss_name, tIss_descpt, tIss_A, tIss_B, tIss_C, tIss_D and tIss_E, accordingly.

In order to reduce the load on the table Negotiation, **Termination** is arranged to hold all records of terminated negotiations, such that each negotiation is represented in a zero or one relation to termination. The PK is terminate_id and the FK, neg_id, is employed in this table to connect terminated negotiations with their issues kept in TerIssue table. The records in n_name, p_descpt, seller_id, buyer_id, s_agent and b_agent in Negotiation are moved to the same fields in Termination. However, the final price and message are contained in f_price and f_message. Moreover, the reason for termination is recorded in n_end, as either "success" or "terminated" based on whether negotiations are ended successfully (i.e. an agreement is reached) or not.

The **TerIssue** table describes the issues of terminated negotiations. Every terminated issue is derived from NegIssue, and each terminated negotiation may have zero, one or many terminated issues. This table contains all the same fields as in NegIssue, except the PK, terIssue_id.

The following tables are based on the agent's Use Case Models and are drawn in the database schema in order to support agent activities.

The **BPN** table stores the buyers' information (elicited by the agent) on price range and strategy. Negotiation is a one-to-one relationship with BPN because there is only one BPN record for every negotiation. Normalization (merging of Negotiation and BPN to one table) of is not performed since the information contained in BPN is highly sensitive to the buyer. The ideal and reserve prices in addition to price rate are recorded in b_priceIdeal, b_priceReserve, and b_priceRate. b_strategy is the strategy employed by the agent whether it be accommodating, compromising, competitive or cooperative. The PK is BPN_id and neg_id is the FK linking the records to the correct negotiation. **SPN** is a reflection of BPN for the seller.

B_Issue contains the issue ratings for issues and options. The zero to one relationship between B_Issue and Issue is due to the fact that if there is no agent employed by the buyer, then there is no need for the issues and options to be rated. b_Iss_id is the PK and neg_id is the FK. The issues name, description rating and options are provided in b_IssName, b_Iss_descpt, b_IssRate, b_IssA, b_IssB, b_IssC, b_IssD and b_IssC, accordingly. Furthermore, each option has a rating, which is kept in b_IssARate, b_IssBRate, b_IssCRate, b_IssDRate or b_IssERate. A similar table, **S_Issue** is employed for the sellers' information.

B_Agent_Offer contains offer information. BAO_id is the PK and neg_id, the FK, is the link to the negotiation table. The offer round, rate and rate increment are recorded as offer_round, offer_rate and offer_inc. However, in every round there is a proposed value rate (offer_prop) that characterizes the offer packages suggested by the agent. The difference between offer_prop and the negotiator's offer rate (offer_rate) is offer_proInc. The same fields are found in **S_Agent_Offer**, except that the PK is SAO_id.

The **B_Agent_Receive** includes information on the offers received by the negotiator. The PK is BAR_id and neg_id is the FK. In order to show the round, rate and rate increment, the received offer is described by receive_round, receive_rate and receive_inc,. With the exception of SAR_id as the PK, the **S_Agent_Receive** has the exact fields.

7.2.2 Prototyping

Prototypes are designed using both Fireworks 4 and Dreamweaver UltraDev 4 from Macromedia. These tools allow pages to be built from a graphical perspective. Prototyping is performed on all EM and agent form fuses by taking the functions on the wireframes and representing those using GUI elements (text boxes, radio buttons, drop-down lists, sliders, buttons, tables, etc.). Figure 33 is a snapshot of a prototype. It illustrates the transformation of the wireframe for Welcome (from Figure 25) to the prototype. In prototyping, additional functionalities are discovered; such as the need for a logout process for the application.

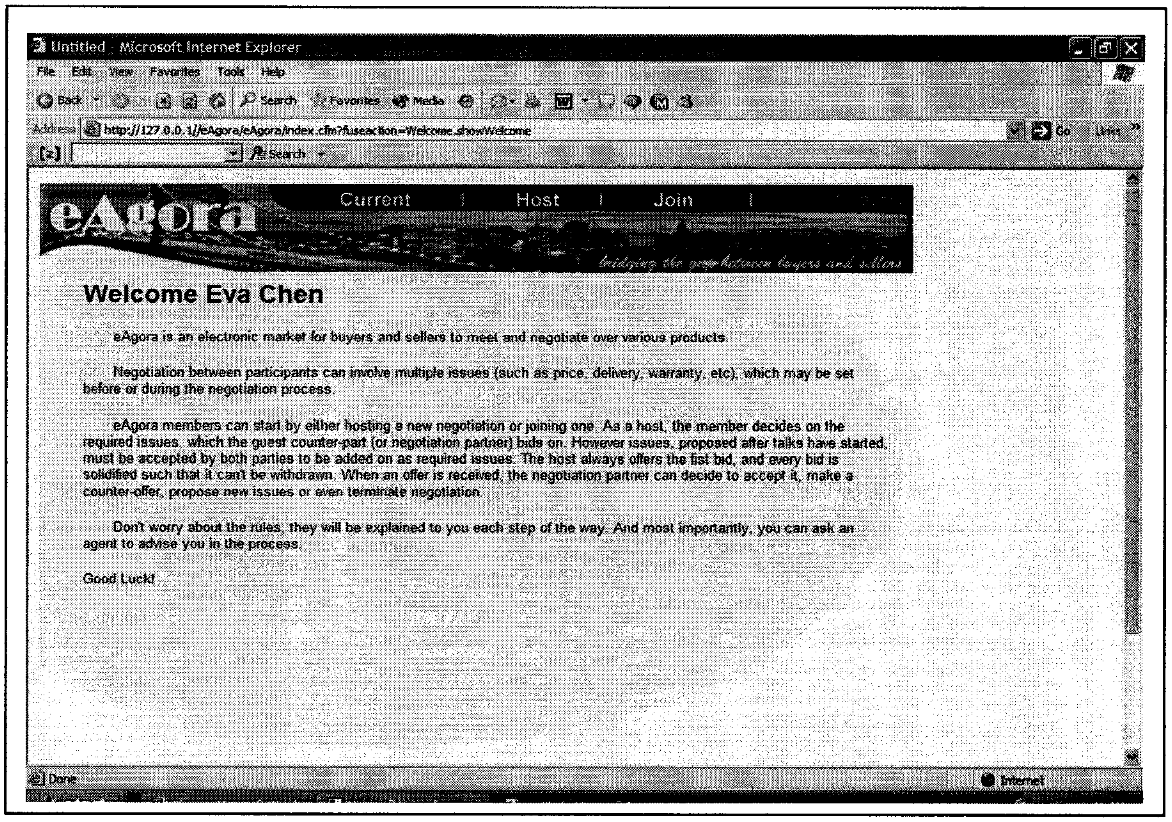


Figure 33. Snapshot of the prototype for Welcome page

One main concern in designing the prototypes involves using general human engineering guidelines (Whitten and Bentley, 1998). This prompts the employment of swing weight method from MCDA research (Section 4.3.1). Sliders are implemented in agent fuses regarding PNS. Figure 34 is an example, where rating values for price and delivery date are elicited from the user. Swing weights present the negotiator with a comfortable and visual environment to assign weights (Belton and Stewart 2001).

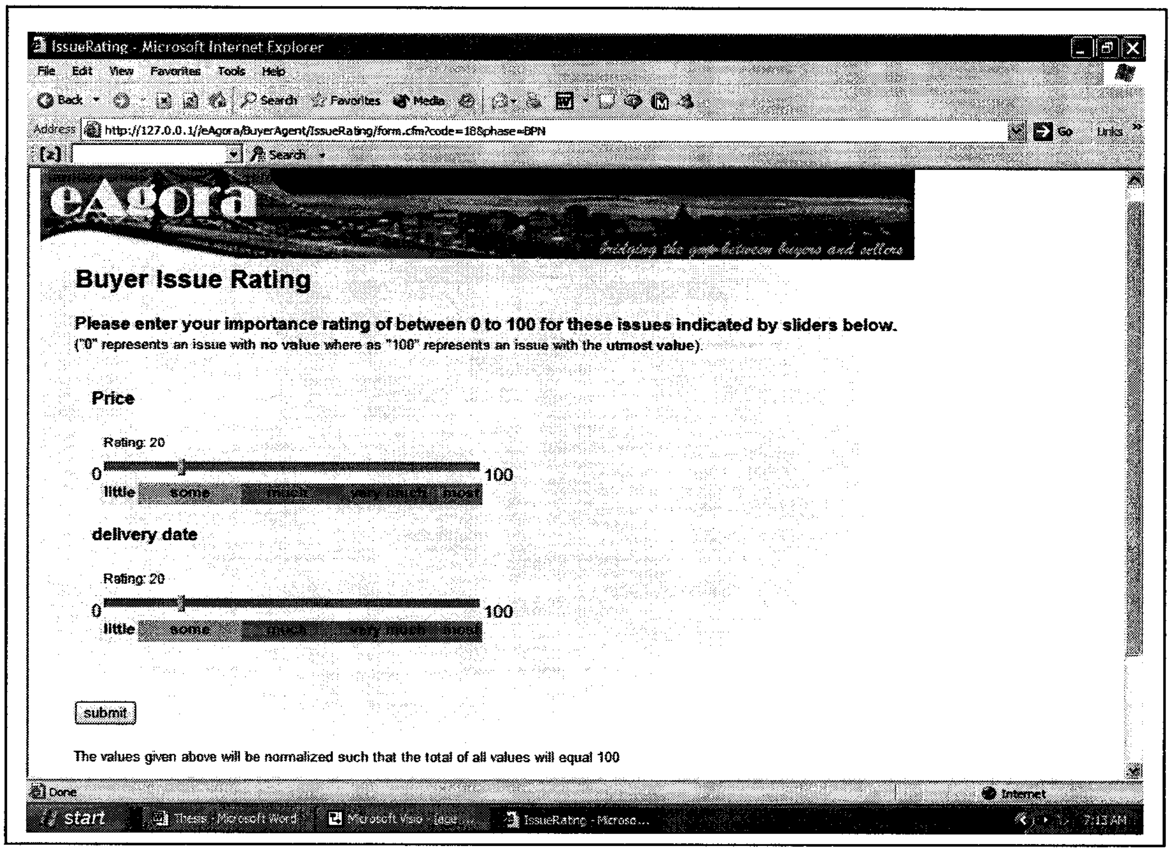


Figure 34. Snapshot of the prototype of PNS rating page

In terms of a practical perspective, prototyping is a very powerful tool because it allows the designer to show the client the appearance of every page before coding takes place. At this stage, any changes requested by the client can be performed without major modifications to the implementation of the site (Peter and Papovich, 2002).

7.2.3 Knowledge-base design

The service model for agent design is the knowledge-base, which consists of the rule that gives intelligence to the agent (Wooldridge Jennings et al., 1999). The purpose of the knowledge base is to provide warnings and suggestions to the user. Warnings are messages that the agent sends out to the user when triggered by an erroneous action from the user.

In the current implementation of eAgora, rules used for warnings are simple and do not require chaining. That is, at any point in time, only two rules are selected that correspond to the user's particular input; one rule is used to compare input with an expected value stored in the knowledge-base, and another is used to check if the input is erroneous. The generic rule-base for warnings is a simple branch decision tree, illustrated in Figure 35. An example in the first case is: a buyer decides to make a greater concession than the value proposed by the agent. This action strays from the strategy stated in PNS and, most importantly, it provides the seller with a greater offer than necessary, putting the buyer in a disadvantage situation. An instance of the second case is: a seller accepts an offer that is below the reservation level. Unchecked, this mistake could mean that the seller is trading below cost. Three activities use warnings: Pre-negotiation, Offer formation and Offer acceptance. Snapshots of the messages are described in Appendix C.

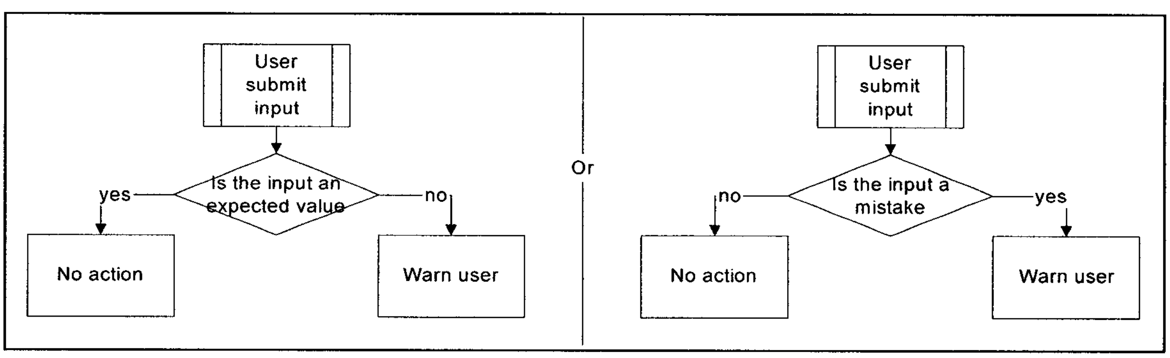


Figure 35. Generic decision tree for warning generation

Suggestions require the agent to conduct a more complex assessment. They occur in two activities: Offer formation and Reception of the counter-offer. In each case, a decision tree is needed to show rules that govern the agent's actions.

The Reception of the counter-offer is an activity that requires a simple tree to dictate the critique for suggestion. The following rule base explanation for message processes 1, 2, 3, 4, 5, 6, 7 and 8 are in reference to Figure 36.

The knowledge-base is initiated by the retrieval of the adversary's offer (pre-defined process), then "receive round" is determined to indicate whether it is the first offer made by the counter-part. If indeed it is such, then the agent calculates the rate based on the value ratings the user provided in PNS. This follows the value-based approach described in Section 4.3.1. If the rate is above the user's acceptable preference level (i.e. rate greater than zero on the user's value scale), then the agent states that the offer is acceptable, but given that it is the first counter-offer, the agent also suggests that the negotiator could try for a better deal (see message process 1). However, if the rate is less than or equal to zero for the first offer, then the negotiator should reject the offer and propose a new offer to the counter-part, as shown in message process 2.

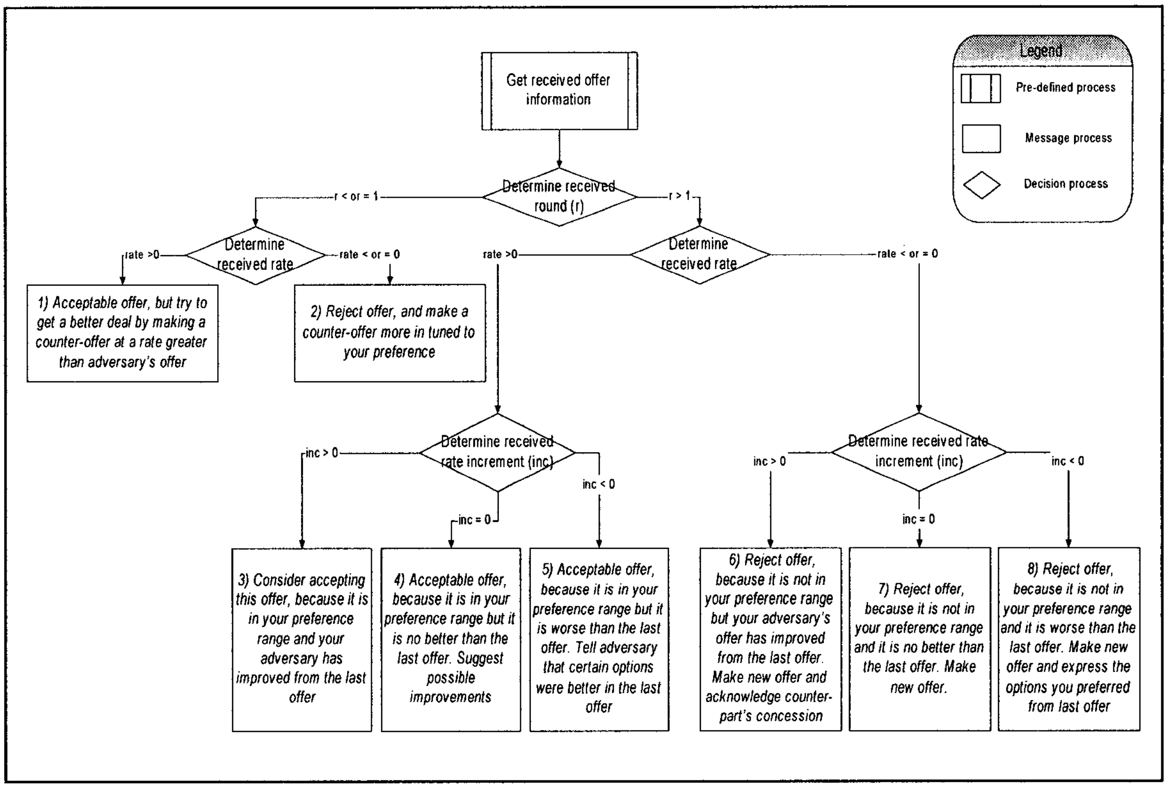


Figure 36. Decision tree for counter-offer reception

If the adversary made at least one offer, the rate is computed along with the received rate increment (Inc), meaning that the difference between the rate of the present offer and that of the previous offer. If the rate is above zero and Inc is greater than zero, then the negotiator should consider accepting the counter-offer because it is above their preference level and the adversary has made a concession towards their needs, as shown in message 3. Yet if the rate is greater than zero and the Inc is equal to zero, the deal is still acceptable but there was no improvement from the last offer. The negotiator can ask that the counter-part make some positive concession in their favor (message 4). Message process 5 also depicts a situation where the rate is positive, but the adversary has worsened the offer from the last. The agent recognizes that the offer is acceptable, but advises the negotiator to tell the adversary that certain options were more appreciated in the last offer.

When the rate is equal or less than zero, the counter-offer cannot be accepted. Message processes 6, 7 and 8 describe this condition. However, message process 6 shows that the adversary has made a positive concession and the negotiator should acknowledge this when making a counter-offer. Message process 5 shows a situation where the adversary has not made improvement from the last offer. Therefore, the user should simply make a new offer to emphasize their preferences. Message process 7 is given when the present offer is worse than the last. Not only should the negotiator make a new offer, they should also stress the points that were more favorable in the last proposal.

The offer formation is a complicated activity for the agent. It employs fuzzy logic reasoning to determine what possible offer packages it can suggest to the user. This area of mathematics allows the agent to represent imprecise variables affecting the range of concession. Fuzzy logic not only provides the agent with intelligence, but it also allows the agent to reactively recommend progressive bidding advice. Figure 37 depicts the decision tree used in offer formation. Before any rules are executed in the knowledge-base, the retrieval of PNS values (a pre-defined process) is required. The first node in the tree is to determine the offer round. It tells the agent whether the user is in the situation of making the first offer (i.e. round equal to zero), the first comeback offer (i.e. round equal to one) or any successive offers (i.e. round greater than one).

In the case of the first offer, the agent proposes the best offer package to the user, meaning that it contains the ideal price and the most preferred option for every issue. The user is always given the choice to follow the suggestion or act independently from the agent.

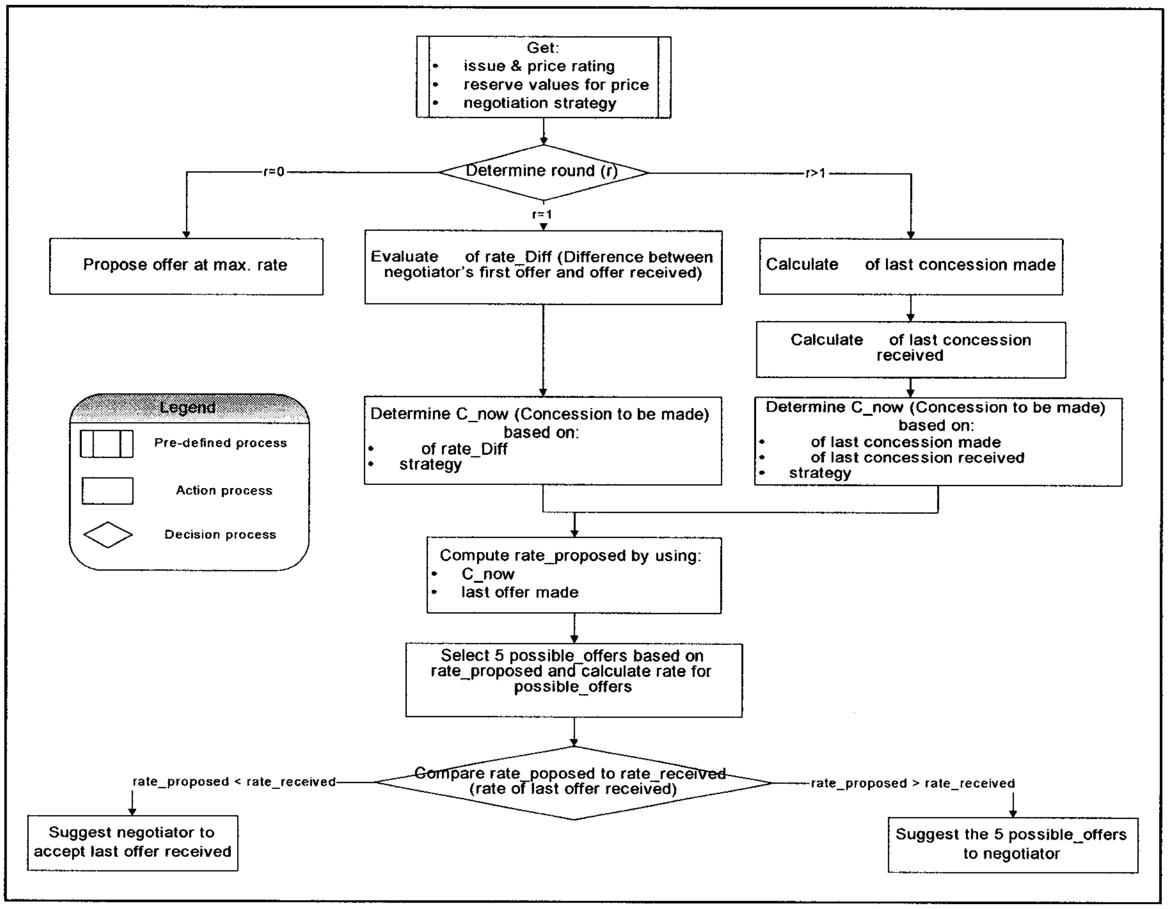


Figure 37. Decision tree for offer formation

If the round number is equal to one, then the membership (μ) of rate_diff (i.e. difference in rates between the negotiator's previous offer and the last counter-offer) is determined in order to calculate the concession needed for the present offer (C_now). For example, if rate_Diff is 0.10, then μ is definitely small (see Figure 38). This means that the negotiator's previous offer rate is very close to their last counter-offer. Therefore, the concession should be small, but the exact value is dependent on the strategy.

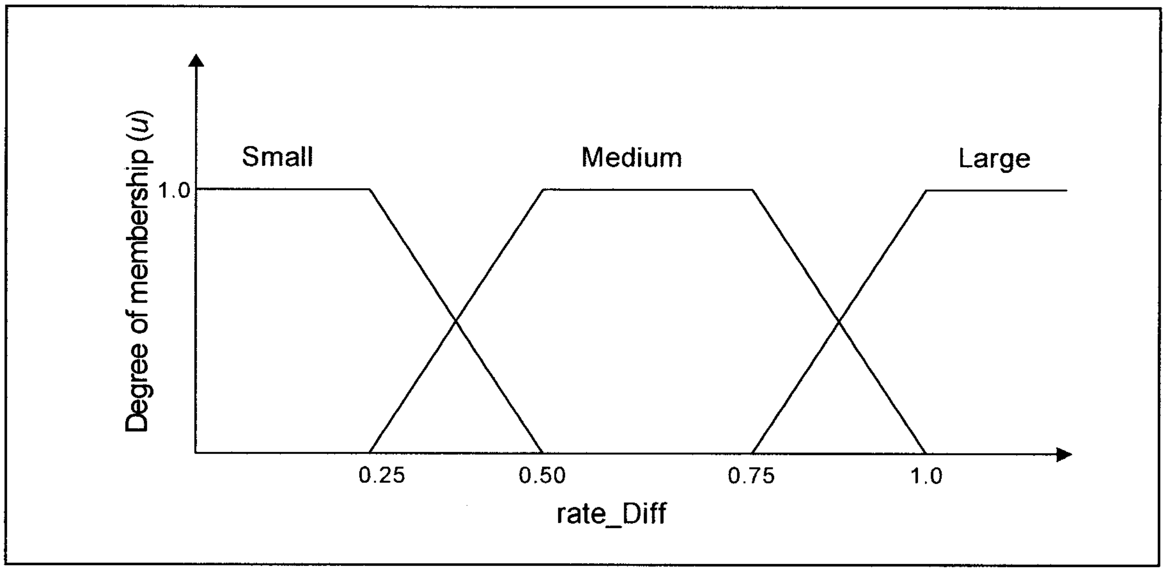


Figure 38. Fuzzy logic graph for rate_Diff

If round number is greater than one, then C_now is computed based on the membership of rate of concession for both the last offer and counter-offer received.

Once C_now is found, it is used to generate a new rate (rate_proposed) that is the basis for creating five offer packages. Before the five packages are suggested to the user, the agent also calculates if rate_proposed is better than the rate of the last offer received (rate_received). Only when rate_proposed is better than rate_received are the five offers recommended to the user. For the detail accounts of the offer formation rules, examples and step-wise calculations see Appendix D.

The overall design of the system is the most important stage to building eAgora. It sets the course for systems programming, meaning that any ambiguity or mistakes in this phase can greatly extend the implementation time.

7.3 Implementation

The first steps to implementing eAgora involve creating a database in the RDBMS (Section 7.3.1). The database serves to store all information relating to the business and negotiation activities, which consists of Web pages that are connected together through a circuit to form a Fusebox. Thus, to set the foundation for coding both the EM and Agent, the next steps are the installation of the Fusebox engine to run the Fuseboxes and prototype dissection (Section 6.2.4). Once Fuseactions and XFA are determined, coding commences on the individual fuse (Section 7.3.2). This component-based approach simplifies programming and reduces code per ColdFusion page. Unit testing is performed the moment a Fusebox is completed and later on the integration of Fuseboxes. The final step is the deployment of eAgora to the server, http://mis.concordia.ca (Section 7.3.3).

7.3.1 Database construction

Database construction is a relatively simple process using the database schema generated in Design. Microsoft Access 2000 is the RDBMS software used to build the tables, fields and relationships. The software's designer interface allows tables and data definition to be created from a graphical perspective. Data definition is especially important because an improper description can prevent values from being added to the database (Rob and Coronel, 2000).

The finished database is uploaded to the Web server into a folder that is mapped to the ColdFusion server using a Data Source Name (DSN). When called in the program, the

DSN connects ColdFusion applications to the database, given that access permission is granted.

7.3.2 Coding

Once the Fuseactions, XFA and fuses are declared, as well as the installation of Fusebox core files, programming can begin in CFML (Mohnike, 2000; Hewitt, 2002). Coding starts with setting up the circuits and folders for EM and agent activities. Afterwards, the switch files are copied to the folders, fuses are coded and XFA are incorporated to link Fuseactions. The greatest challenge to coding eAgora lies in the fact that negotiations can have infinite issues. This means that a simple variable is not sufficient to handle issues. Moreover, it presents a difficult problem for the agent when generating suggestions for offer packages. Therefore, the solution is to use arrays and embedded variables to capture any possible number of issues used in negotiation. eAgora is programmed to manage infinite issues, but due to the lengthy execution time required to manipulate a large array, the agent limits the negotiator to a maximum of five issues. Five issues can translate to over 300,000 offer packages[3] for the agent to analyze. See Appendix C for snapshots of eAgora and the accompanying CD for the code.

7.3.3 Unit testing and Deployment

Unit testing involves first verifying that each fuse functions properly. The procedure is then moved up to the Fuseactions and sequentially the Fuseboxes. Unit testing allows

[3] 5 issues to the power of 5 options multiply by a 100 point distribution of price equals 312,500 possibilities

implementation problems to be isolated and easily solved. Most essentially, Fusebox integration is facilitated knowing that the unit logic is correct.

Integration is concerned with bring together the code to create the entire site. If all prior steps have been carried out correctly, the only bugs remaining are with merging Fuseboxes. The organization of the Fusebox environment makes unit testing and integration of business and negotiation activities a structured process, leaving only minor glitches for overall testing.

Deployment is asserting the proper execution of the application in the production environment. Since eAgora is programmed with XFA, very few changes occurred in moving it from the development platform to the University server. Most alterations affect removing or updating information in the database. The resulting eAgora is found at: http://mis.concordia.ca/students/eva/eAgora/eAgora/index.cfm .

The Fusebox milieu is ideal for the implementation of this thesis project because of the many business and negotiation activities involved. The engine manages the application flow such that programming efforts are focused mainly on functionality and not on passing between activities within the system. Furthermore, in the future, eAgora can be easily expanded and modified by adding, updating or removing Fuseboxes.

The overall development of eAgora is based on the stages of the SDLC in conjunction with traditional database development, Fusebox and Agent-Oriented methodologies. These three approaches are complementary in generating a model-driven framework that is rational and transparent.

8 Evaluation and Discussion

The final testing of eAgora consisted of a usability test based on a convenience sample, such that the results cannot be generalized. People from various backgrounds and age-groups were invited to use eAgora, with and without the agent, to verify the actual functionality and concepts of the system in a hands-on approach (Whitten and Bentley, 1998). The main focus was to determine whether the design and implementation of eAgora generated an EM ideal for Web-based negotiation. Furthermore, the agent's features were also examined to confirm that it succeeded in supporting negotiators. Section 8.1 describes the manner in which the usability tests are conducted, and Section 8.2 presents the results of the testing.

8.1 Usability testing

The entire evaluation comprised of a three-part questionnaire (pre-test, post-test without agent and post-test with agent), and two negotiation scenarios (the first without an agent and the second with an agent). See Appendix E for the questionnaire based on Lo's work (2001). The questions were read and explained to the users before they answered the questionnaire.

Twelve individuals divided into six pairs volunteered to test eAgora. One volunteer in a pair served as a buyer and the other as a seller to evaluate the system. They started by completing the pre-test part of the questionnaire to determine their expertise level on negotiation and Internet usage. A demonstration of eAgora was given to show all the features and functionalities of the site, during which user comments and questions were recorded. The demonstration served to teach the volunteers how use the site and the negotiation process. The volunteers were then asked to select a product that they are familiar with and comfortable negotiating on. Once a product was settled on, they were directed into different rooms to start negotiations.

The first negotiation scenario employed no agent. It began with one user hosting a new negotiation on the product and deciding on issues (each with minimum two to maximum five options) important for the transaction. For example, the issue delivery date may have as options: 2 days, 5 days or 10 days. The users were asked to negotiate with a minimum of three and a maximum of five issues, which could be determined at the hosting stage or during negotiations. The second user then joined this negotiation. They commenced exchanging offers and messages while a monitor remained by their side to observe any problem and to answer any technical questions about the eAgora. After talks ended (either because an agreement was reached or one party walked away), the volunteers were asked to respond to the second part of the questionnaire. This measured how well eAgora (without an agent) serves to support Web-based negotiations.

The second scenario involved negotiations using an agent. Again, the volunteers were asked to select a product for negotiation. They were advised to choose a different product in order to start new talks, where previous knowledge of each other's preference is insignificant. They proceeded exactly as the first scenario except that they informed the agent on: their preference values for each issue and option, the price range and negotiation strategy. Based on these pre-negotiation settings, the agent evaluated each counter-offer received and suggested possible offers to propose to the opponent. Once negotiations in the second scenario were completed, the final part of the questionnaire was filled out to show how well the agent performed from the volunteers' perspectives.

8.2 Results

The responses provided by the users were collected and calculated to present the overall picture of eAgora's usability. See Appendix F for the table of results. The average respondents use the Internet around 17 hours a week and are familiar with buying product over the Web. However, only two of these had experience with Web-based negotiation.

The general feedback from the user showed that 92% of the respondents are in favor of employing eAgora to buy or sell products over the Web. The average user claimed that they were very satisfied with the usability of site. They rated their experience as between very good and great. One user expressed the following:

> "*eAgora is like the classified section of the newspaper, but much more. You can negotiate with the seller or buyer right away, and have a deal ready before you go and meet them. This can save you time and money*"

The respondents liked the fact that eAgora is easy to use and issues can be added during negotiation. As an improvement, they suggested that a help function is needed to explain the terminologies found on the site. In addition, eAgora could have different protocols allowing different approaches to negotiation. For example, the issues proposed during negotiation could be added right away as required issues, instead of waiting for the adversary's agreement. They also mentioned that a picture upload function is required for the site. This would give the buyer a visual representation of the product.

In terms of the agent, 83% of the participants responded that the agent provided useful advice and suggestions during negotiations. Everyone felt that they were in control of the negotiation with the agent present. In addition, 17% more participants reached an agreement using the agent during negotiation. They rated their satisfaction level with the agent as between good and very good. Most individuals (83%) said that they would use the agent in future negotiations on eAgora. Here are three comments from different volunteers:

1. "*The agent was very useful because it gave me several suggestions on what to offer my opponent, and it helped me understand the value of the offers that I got.*"

2. "*I used the agent's suggestions as a guideline to make my offers.*"

3. "*The agent prevented me from accepting a bad deal. It showed me what good and bad offers are.*"

Some respondents expressed that the current features of the agent are limited. The agent could improve by allowing negotiators to adjust their pre-negotiation settings during talks. It could also show the users different scenarios to help them select a strategy for negotiation. The agent may want to adopt an entity that users could relate to, similar to the paper clip used by Microsoft Office applications.

Usability testing is an important part of any system's evaluation. The feedback provided by the users is essential to upgrading the existing functionalities and creating new features for eAgora.

9 Conclusion

The literature review demonstrated that as EC grows, negotiation will no doubt be an essential activity in EM. Moreover, multi-issue negotiation will play an important part in allowing for mutually beneficial transactions between buyers and sellers. The influx of negotiators with various levels of expertise will require an agent that supports their activities in the EM. In order to properly aid these negotiators, the agent must encompass basic, adaptive and negotiation traits. It must also remain subservient to the users such that they always feel in control of the process.

The contributions of this thesis are: (1) the creation of an innovative EM that allows participants to negotiate multiple issues, (2) the incorporation of an agent that provides users with warnings of critical errors, timely advice on offers received and suggestion on possible offers to propose, (3) the development of a novel framework for integrating Web-based and agent technologies and, (4) the formulation of a platform for future studies in the areas of multi-issue e-negotiation, EM and agent technology.

The methodological choices to construct a framework for developing the EM with supporting agent consist of utilizing the three stages of the SDLC (analysis, design and implementation) to apply the three methodologies (traditional database development, Fusebox and Agent-Oriented) in relation to the three building blocks (data, EM and agent). The resulting system, eAgora, is evaluated through usability testing, which showed that it is well received. 91% of the participants said that they would use eAgora to sell and buy products, and 83% found the agent useful in negotiations.

9.1 Limitation of this thesis

The limitations of this thesis project are due mostly to time and budget constraints, such that restrictions are placed on the functional and technical aspects of eAgora, concerning both the EM and agent.

Two functional flaws, that respondents in the evaluation found are the existence of only one protocol for the buyer and one for the seller and the lack of any help or tutorial features to support usage of the site. If eAgora is to progress to public testing, then a tutorial and help features will be necessary to teach and assist users. However, the option of providing numerous protocols may be confusing for participants. Instead, the present protocol could be re-designed to one that is more intuitive.

The database for eAgora is presently managed using Microsoft Access. As more users are given permission to the site, a better RDBMS is needed to sustain the traffic. Another technical shortcoming of the system is that the Fusebox engine does not allow for dynamic slider bars, which are required for the swing weight method. In order to circumvent this, the pre-negotiation settings are programmed outside of the Fusebox environment, which deviates from proper Fusebox practice. The developers of Fusebox announced that a new version (Fusebox 4) is coming out in late 2003. Hopefully this problem is corrected, or otherwise the solution is to employ an advanced JavaScript programmer to build the slider in JavaScript within Fusebox.

The obvious limitation with the agent is that negotiations are restricted to five issues. Either a more powerful server or a better approach to generating the suggestions for possible offers can correct this. The agent is also limited by the present knowledge-base. If the agent is to become more intelligent a large knowledge-base is needed, in addition to an inference engine to handle the execution of rules. Another shortcoming of the agent is that pre-negotiation settings cannot be changed during talks. A simple link can be added to the existing system to rectify this problem.

Another limitation with this study is that neither a laboratory nor field experiment was performed on eAgora. Even though a usability test was conducted, it does not have the scientific vigor of an experiment.

9.2 Future research

In order to improve on this work, a full-scale experiment and enhancement to the agent are possible areas for future research. A laboratory experiment involving control and treatment groups on eAgora with and without agent can be used to measure users' satisfaction, agent's performance (in terms of effectiveness and efficiency), etc. The agent can be expanded to allow for product search, user and opponent profiling, and strategy building.

eAgora is developed as an e-negotiation platform that supports transactions between buyers and seller, allows the teaching of e-negotiation and permits the study of social interactions as well as behavioral aspects of multi-criteria problem solving between multiple parties. Therefore, eAgora has the potential to further the dimensions of EC by providing a commercial market and by serving as an educational tool.

References

Alem, L., R. Kowalczyk, et al. (2000). "Recent Advances in Negotiation Agents." The International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet, SSGRR 2000, L'Aquilla, Italy.

Applegate, L. M. and C. W. Holsapple (1996). "Electronic Commerce: Building Blocks and New Business Opportunity." Journal of Organizational Computing and Electronic commerce 6(1): 1-10.

Azvine, B., N. Azarmi, et al. (1997). "An Introduction to Soft Computing: A tool for Building intelligent Systems." Software Agents and Soft Computing. H. S. Nwana and N. Azarmi, Springer-Verlag: 191-210

Bakos, Y. (1998). "The Emerging Role of Electronic Marketplaces on the Internet." Comm. ACM 41(8): 35-42.

Bazerman, M. H. and M. Neale (1992). Negotiating rationally. New York, Free Press.

Beam, C. and A. Segev (1996). "Automated Negotiation: A survey and the state of Art." Hass School of Business (Berkeley CA).

Benyoucef, M., H. Alj, et al. (2001). "Combined Negotiation in E-Commerce: Concepts and Architecture." Journal of Electronic Commerce Research 1(3): 277-299.

Belton, V. and T.J. Stewart (2001). "Multiple Criteria Decision Analysis: An Integrated Approach." Kluwer Academic Publishers (Boston, MA)

Bloch, M. (1996). "On the Road of Electronic Commerce." www.stern.nyu.edu/~mbloch/doc/roadtoec/ec.htm.

Bruce, T.A. (1992). "Designing Quality Database with IDEF1X Information Models." Dorsett House Publishing (New York)

Bui, T., J. Yen, et al. (2001). "A Multi-attribute Negotiation Support System with Market Signaling for Electronic Markets." Group Decision and Negotiation 10: 515-537.

Chavez, A., D. Dreilinger, et al. (1997). "A Real-life Experience in Creating an Agent Market." Application of Intelligent Agents and Multi-Agent Technology PAAM'97 (London UK).

Choi, S. Y., D. O. Stahl, et al. (1997). The Economics of Electronic Commerce. Indianapolis, Macmillan Technical Publishing.

Clay, k., K. R. Krishnan, et al. (2001). "Prices and Price dispersion on the Web: Evidence from the On-line Book Industry." National Bureau of Economic Research working Paper #8271, May.

Collins, J., B. Youngdahl, et al. (1998). "A Market Architecture for Multi-Agent Contracting." Proc. of the 2nd Int. Conference on Autonomous Agents.

Croson, R. T. (1999). "Look at Me When You Say That: An electronic negotiation simulation." Simulation and Gaming 30(1): 23-37.

Foroughi, A. (1995). "A Survey of Use of Computer Support for Negotiation." Journal of Applied Business Research(Spring 1995): 121-134.

Fusebox, (2003). http://fusebox.org

Han, J. and M. Kamber (2001), "Data Mining: Concepts and techniques". Morgan Kaufman (San Francisco, CA)

Hartman, A., J. Sifonis, et al. (2000). "Net Ready". USA, McGraw-Hill.

Hewitt, E. (2002). "Core ColdFusion 5.0." Prentice Hall (Upper Saddle River, NJ)

Holsapple, C. W. (1996). "Implications of Negotiation Theory for Research and Development of Negotiation Support Systems." Group Decision and Negotiation 6(3): 255-274.

Hsu, C.-N. and C. A. Knoblock (1993). "Reformulating Query Plans for Multi-database Systems." Proc. of the 2nd Int. Conference on Information and Knowledge Management (Washington D.C.).

Ivar, J., M. Christerson, et al. (1992). "Object-Oriented Software Engineering: A Use Case Driven Approach." Wesley (Wokingham, England).

Jennings, N. R. and J. R. Campos (1997). "Towards a Social-level Characterization of Socially Responsible Agents." IEE Transactions on Software Engineering 144(1): 11-25.

Jennings, N. R., K. Sycara, et al. (1998). "A Roadmap of Agent Research and Development." Autonomous Agent and Multi-Agent Systems 1: 275-306.

Jennings N. R. and M. Wooldridge (1998). "Applications of Intelligent Agents." Agent Technology: Fondation, Applications and Markets. Springer-Verlag

Jennings, N. R. and P. Faratin (2001). "Automated Negotiation: Prospects, Methods and Challenges." Group Decision and Negotiation 10: 199-215.

Kambil, P. F. and D. Wilson (1999). "Transforming the marketspace with all-in-one markets." Int. J. Electronic Commerce 3(4): 11-28.

Katz, J. E. and P. A. Aspden (1997). "Nation of Strangers?" Comm. ACM 40(12): 81.

Keeney, R. and H. Raiffa (1976). "Decision with Multiple Objectives" Wiley, New York.

Kersten, G. E. (2002). "E-negotiation Engineering: Review of the Emerging Field." Working Paper.

Kersten, G. E. and G. Lo (2001). "Negotiation Support Systems and Software agents in E-business Negotiations." 1st International Conference on Electronic Business (Hong Kong).

Kersten, G. E. and S. J. Noronha (1998). "Negotiation and the Internet: Users' Expectations and Acceptance." http://interneg.org/interneg/research/papers/1998/01.pdf.

Kersten, G. E. and S. J. Noronha (1999). "Negotiations in Electronic Commerce: Methodological Misconceptions and a Resolution." http://interneg.org/interneg/research/papers/1999/02.pdf.

Lai, H. and T.-C. Yang (2000). "A System Architecture for Intelligent Browsing on the Web." Decision Support Systems 28: 219-239

Lander, S. and V. Lesser (1993). "Understanding the Role of Negotiation in Distributed Search Among Heterogeneous Agents." Proc. of the 13th Int. Joint Conference on Artificial Intelligence (France).

Lee, K. J., Y. S. Chang, et al. (2000). "Time-bound Negotiation Framework for Electronic Commerce Agent." Decision Support Systems 28: 319-331.

Lee, O. (1998). "Negotiation and Contract though E-mail." Electronic Markets 8(3): 17-20.

Lewicki, R., D. Saunders, et al. (1997). "Essentials of Negotiation." McGraw-Hill (USA).

Li, H., C. Huang, et al. (2002). "Design and Implementation of Business Objects for Automated Business Negotiations." Group Decision and Negotiation 11: 23-44.

Lo, Gordon (2001). "Active Negotiation Support with a Software Agent." Master of Business Administration Thesis, Carleton University (Ottawa, Canada).

Lui, J. (2001). Autonomous Agents and Multi-Agent Systems. Singapore, World Scientific.

Maes, P., R. Guttman, et al. (1994). "Agents That Reduce Work and Information Overload." Comm. ACM 27(7).

Maes, P., R. Guttman, et al. (1998). "Agents That Buy and Sell." Comm. ACM 42(3): 81-91.

Malone, T. W., J. Yates, et al. (1987). "Electronic Markets and Electronic Hierarchies." Comm. ACM 30(6): 81-91.

Martin, J. and J. Odell. (1992). "Object-Oriented Analysis and Design." Prentice Hall (Englewood Cliffs, NJ).

Moknike, C. (2000). "SAMS Teach Yourself Allaire ColdFusion." Sams Publishing (Indianapolis, Indiana)

Moukas, A., R. Guttman, et al. (1998). "Agent-Mediated Electronic Commerce: An MIT Media Laboratory perspective." Proc. of the Int. Conference on Electronic Commerce (Korea).

Negroponte, N. (1997). "Agents: from Direct Manipulation to Delegation." Software Agents. J. Bradshaw, M., MIT Press: 57-66

Norman, D. (1994). "How People Interact with Agents." Comm. ACM 37(7).

Nunamaker, J. F., A. R. Dennis, et al. (1991). "Information Technology for Negotiating Groups: Generating options for mutual gain." Management Science(October 1991).

O'hara, G. M. P. and N. R. Jennings (1996). Foundations of Distributed Artificial Intelligence. USA, John Wiley and Sons.

Oliveira, E. (1999). "A Multi-criteria Negotiation on Multi-Agent Systems." Proc. of the 1st Int. Workshop of Central and Eastern Europe on Multi-Agent Systems (Russia).

Oliver, J. R. (1997). "A Machine- Learning Approach to Automated Negotiation and Prospects of Electronic Commerce." Journal of Management Information Systems 13(3): 83-112.

Parkin, M. and R. Blade (1997). "Microeconomics 3rd edition." Addison Wesley (Canada).

Peters, J. and N. Papovich (2002). "Fusebox: Developing ColdFusion Applications." New Riders (Indianapolis, Indiana).

Raiffa, H. (1998). The Art and Science of Negotiation. Cambridge, Harvard University Press.

Rob, P. and C. Coronel (2000). "Database Systems: Design, implementation and management 4th edition." Course Technology Thomson Learning (Cambridge, MA).

Rosenschein, J. S. and G. Zlotkin (1994). Rules of Encounter. USA, MIT Press.

Rubinstein, A. (1982). "Perfect Equilibrium in a Bargaining Model." Econometrica 50(1): 97-110.

Sandholm, T. W. (1999). "Automated Negotiation: The best for all concerned." Comm. ACM 42(3): 84-85.

Sandholm, T. W. and N. Vulkan (1999). "Bargaining with Deadline." Proc. National Conference on Artificial Intelligence (Florida).

Sarkar, M. B., B. Butler, et al. (1995). "Intermediaries and Cybermediaries: A Continuing Role for Mediating Players in the Electronic Marketplace." Comp.-Mediated Comm. 1(3).

Scott-Morton, M. and T. J. Allen (1994). Information Technology and the Corporation of the 1990's. New York, Oxford University Press.

Sobel, R. (2000). "The Pursuit of Wealth: The Incredible Story of Money Throughout the Ages." McGraw-Hill (NewYork, NY)

Stewart, T.J. (1992). "A Critical Survey on Status of Multiple Criteria Decision Making Theory and Practice." OMEGA Int. J. of Management Science. 20(5): 569-589

Ströbel, M. (1999). "Effects of Electronic Markets on Negotiation Process: Evaluating Protocol Suitability." IBM, Zurich Research Laboratory (Switzerland).

Ströbel, M. (2001). "Design of Roles and Protocols for Electronic Negotiation." IBM, Zurich Research Laboratory (Switzerland).

Swatman, P. M. C. and P. A. Swatman (1992). "EDI System Integration: A defenition and Literature Survey." The Information Society 8(3): 169-205.

Sycara, K., S. Roth, et al. (1991). "Distributed Constraint Heuristic Search." IEE Transactions on System, Man, and Cybernetics 21: 1446-1461.

Tapscott, D. and A. Caston (1993). Paradigm Shift: The New Promise of Information Technology. New York, McGraw-Hill.

Teich, J., H. Wallenius, et al. (1999). "Multiple Issue Auction and Market Algorithms for the World Wide Web." Decision Support Systems 26: 49-66.

Thompson, L. (1998). The Mind and Heart on the Negotiator. New Jersey, Prentice Hall.

Thompson, L. and J. Nadler (2002). "Negotiating via Information Technology: Theory and application." Journal of Social Issues 58(1): 109-124.

Turban, E., J. Lee, et al. (2000). "Electronic Commerce: A managerial perspective." Prentice Hall (New Jersey).

Vahidov, R. and R. Elrod (1999). "Incorporating Critiquing and Argumentation in DSS." Decision Support Systems 26: 249-258.

Warkentin, M., R. Bapna, et al. (2000). "The Role of Mass Customization in Enhancing supply Cahin Relationships in B2C E-Commerce Markets." Journal of Electronic Commerce Research 1(2).

Wasfy, A. and Y. Hosni (1998). "Two-Party Negotiation Modeling: An Integrated Fuzzy Logic Approach." Group Decision and Negotiation 7: 491-518.

Weiss, G. (1998). Multi-Agent Systems. USA, MIT Press.

Wetherbe, J. (1994). "Systems Analysis and Design: Best Practices, 4th ed." West Publishing (St. Paul, MN).

Whitten, J. and L. Bentley (1998). "Systems Analysis and Design Methods 4th ed." McGraw-Hill (Boston, MA)

Winoto, P. et al. (2002). "An Extended Alternative-Offer Bargaining Protocol for Automated Negotiation inMulti-agent Systems." to appear in: Proc. of the 10th Int. Conference on Cooperative Information Systems (USA, November 2002).

Wooldridge, M., N. R. Jennings et al. (1999). "A Methodology for Agent-Oriented Analysis and Design." Autonomous Agents (Seattle, WA).

Zeng, D. and K. Sycara (1998). "Bayesian Learning in Negotiation." Int. J. Human-Computer Studies 48: 125-141.

Zachman, J. (1987). "A Framework for Information System Architecture." IBM Systems Journal 26(3).

Zwass, V. (1996). "Electronic commerce: structures and issues." Int. J. Electronic Commerce 1(1): 3-23.

Appendix A: Seller Negotiation Protocol

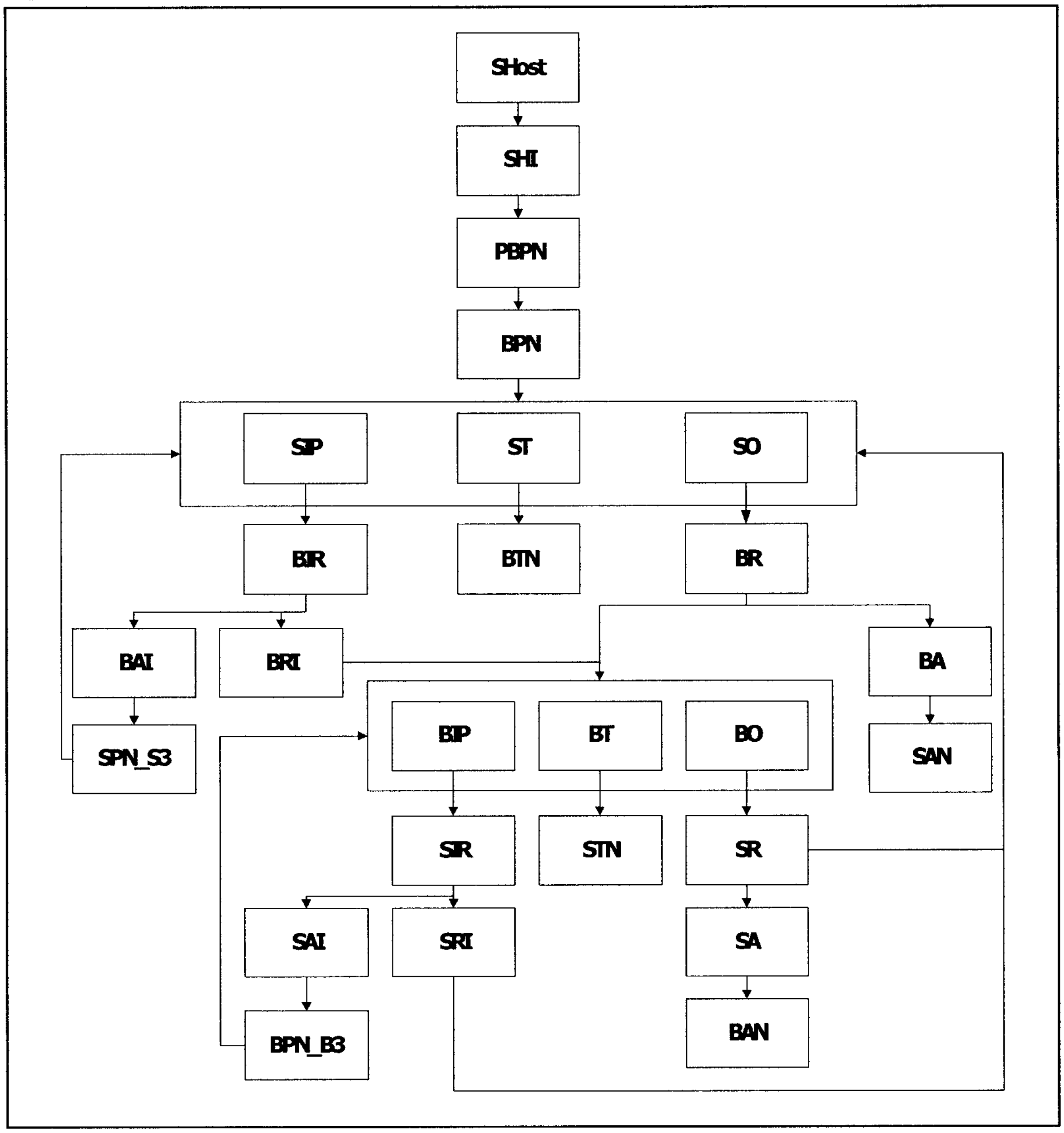


Figure 39. Seller negotiation protocol

Appendix B: eAgora Fuseaction Diagrams

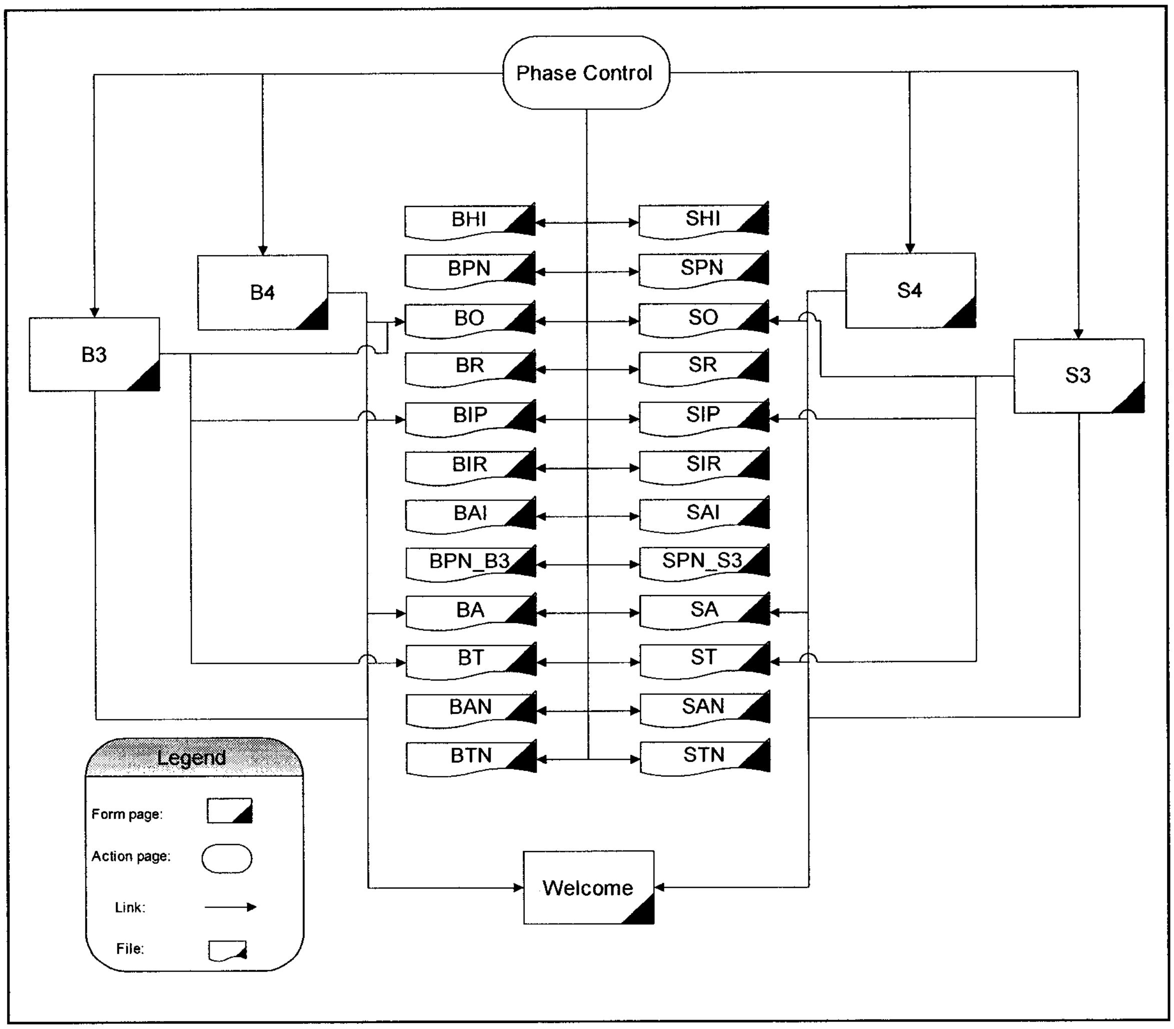


Figure 40. Fuseaction diagram for control phase[4]

The Fuseaction diagrams for negotiation activities are the same for both buyer and sellers. For the sake of simplicity, the following diagrams are represented in terms of the buyer.

[4] B4: Buyer is given 4 possible actions to take (BO, BIP, BA and BT). B3: Buyer is given 3 possible actions to take (BO, BIP and BT). S4: Seller is given 4 possible actions to take (SO, SIP, SA and ST). S3: Seller is given 3 possible actions to take (SO, SIP and ST).

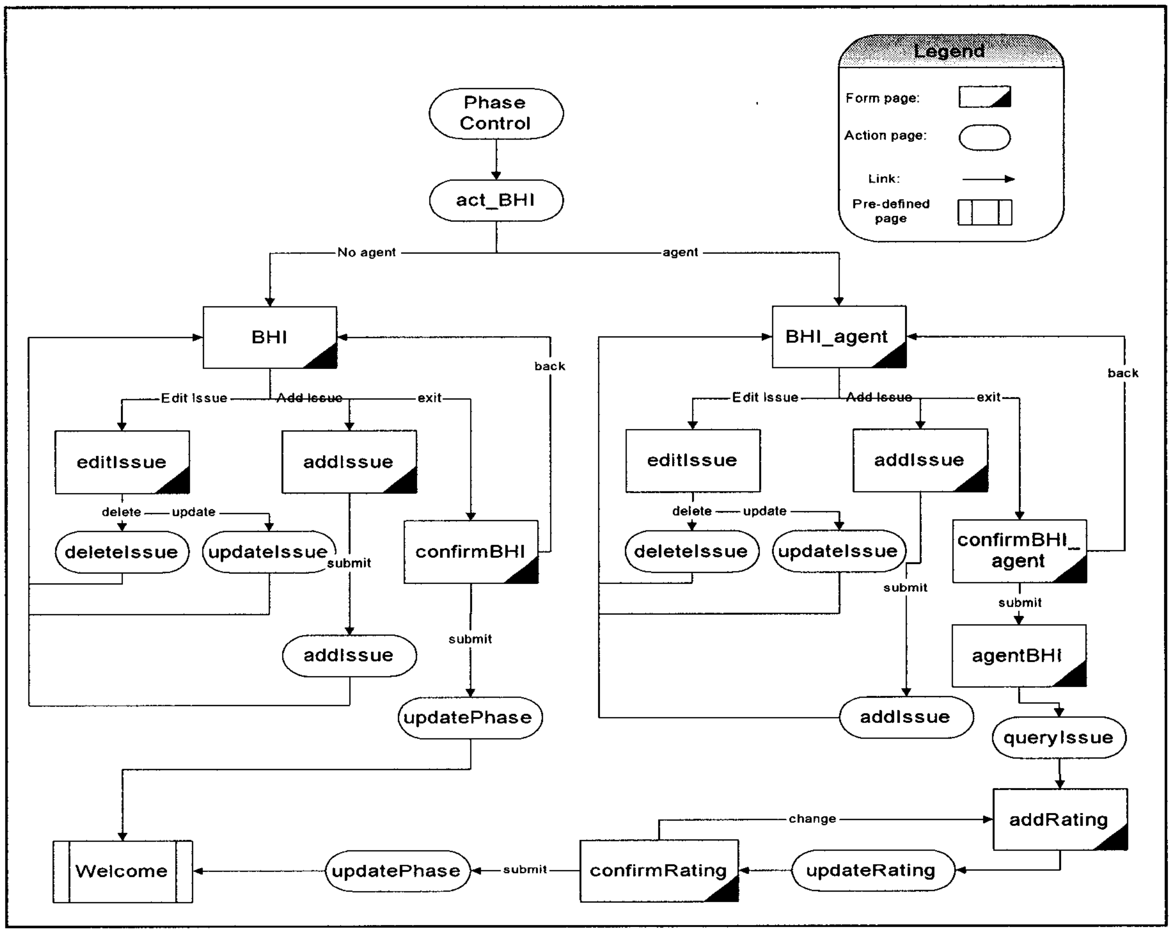


Figure 41. Fuseaction diagram for BHI

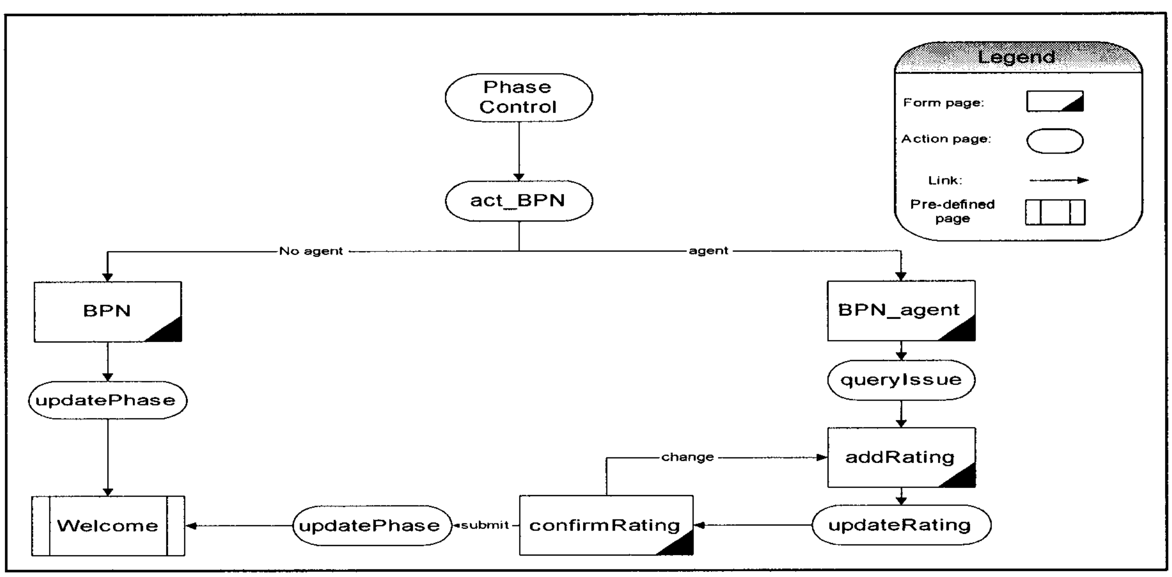


Figure 42. Fuseaction diagram for BPN

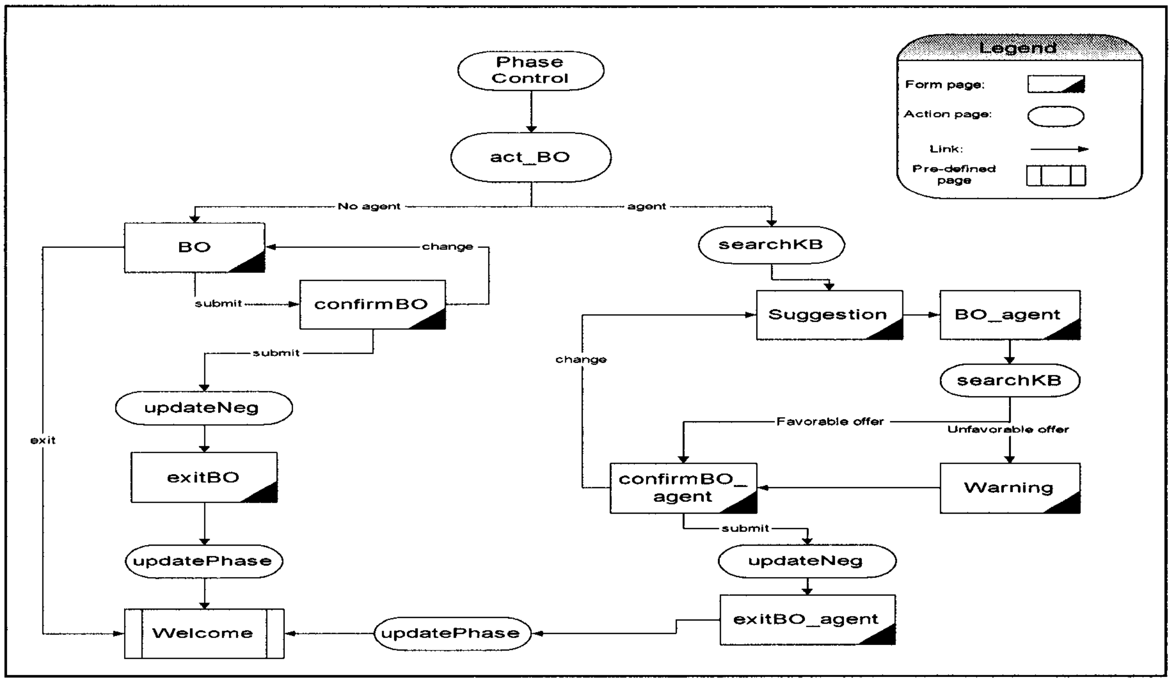


Figure 43. Fuseaction diagram for BO

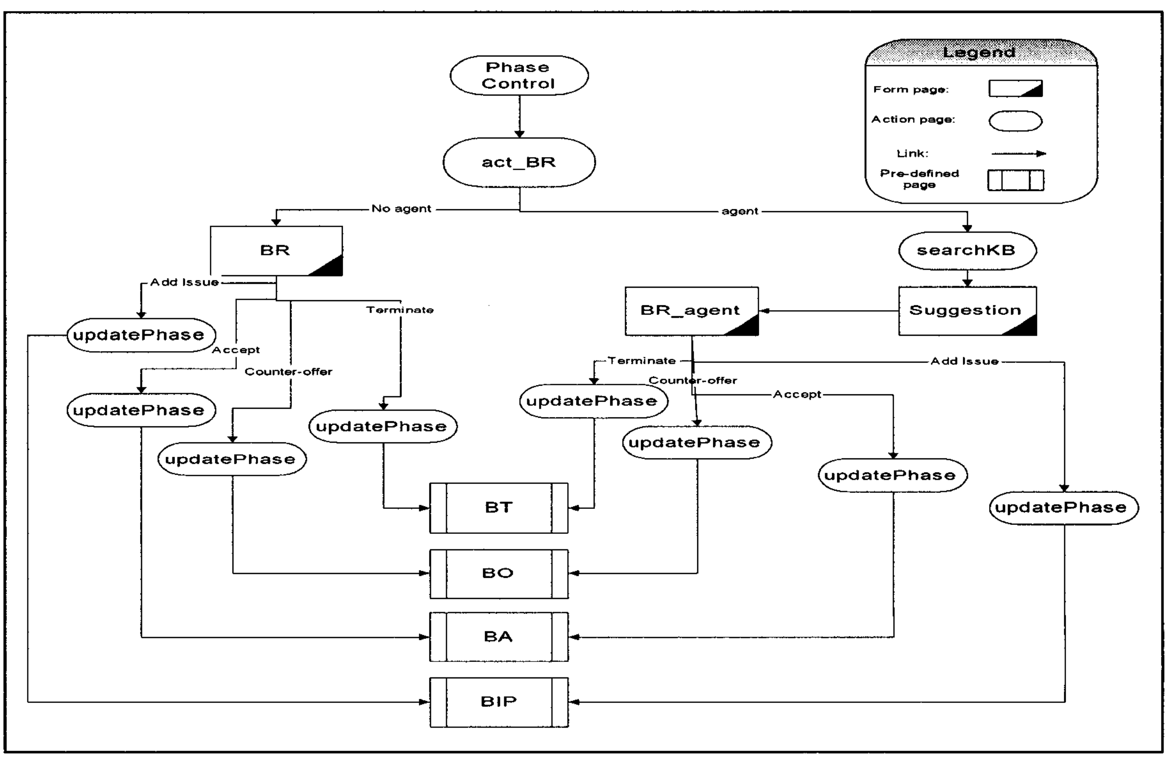


Figure 44. Fuseaction diagram for BR

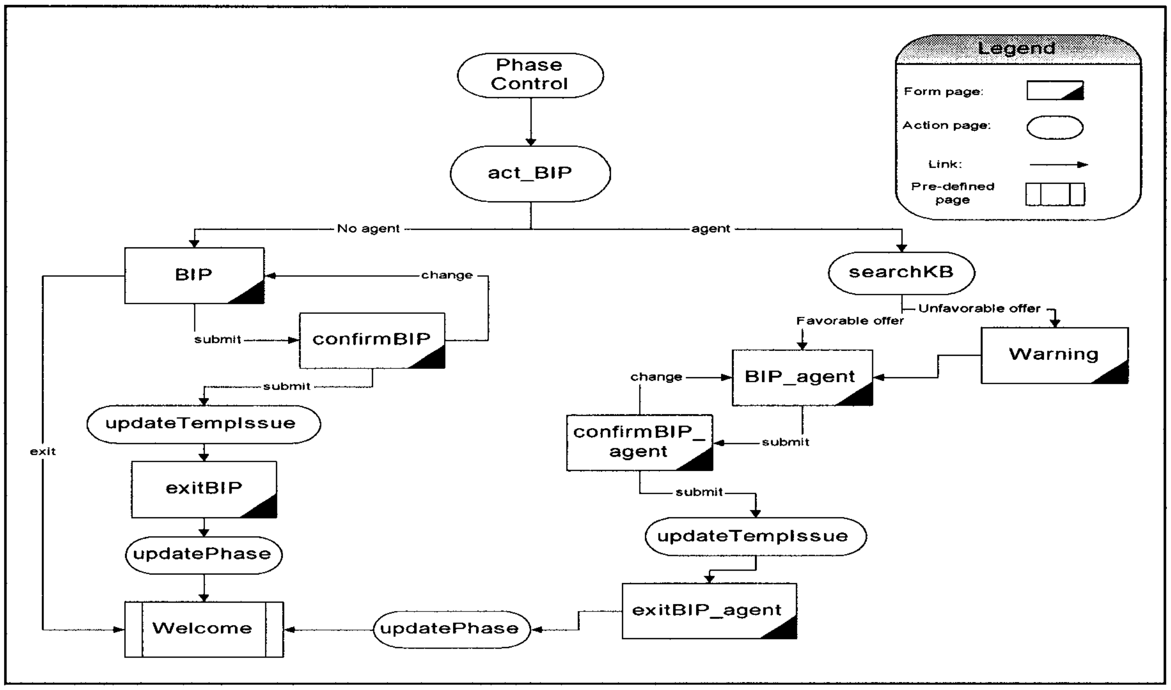


Figure 45. Fuseaction diagram for BIP

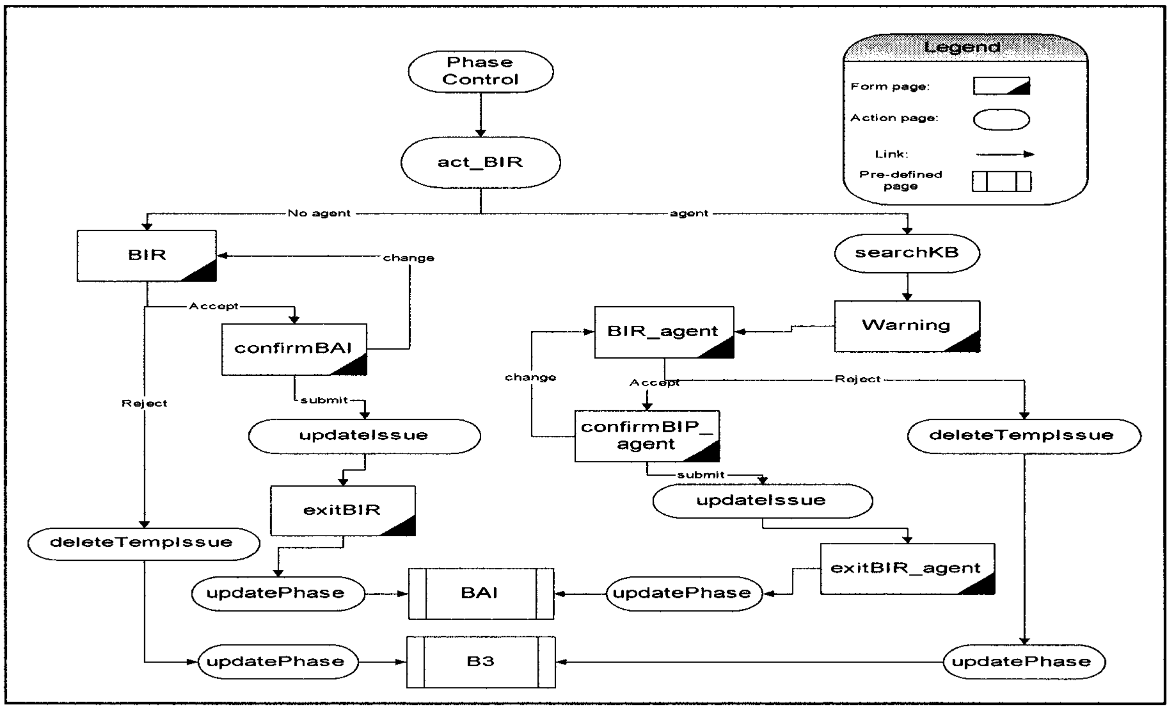


Figure 46. Fuseaction diagram for BIR

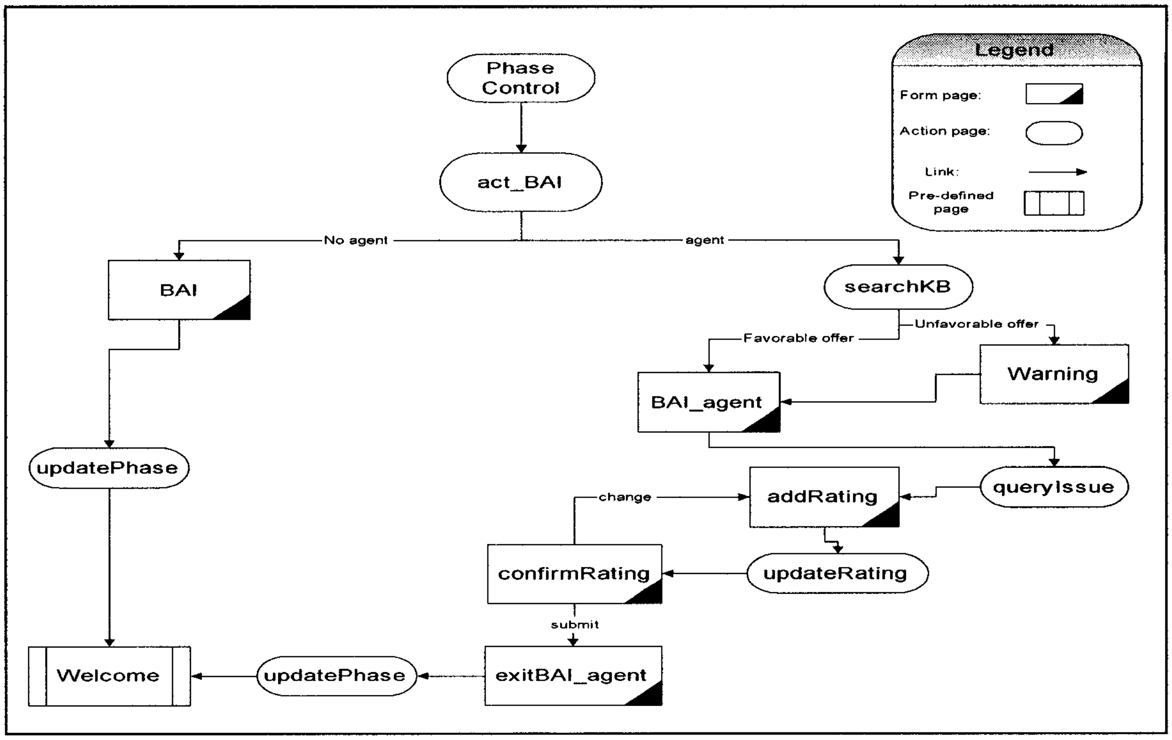


Figure 47. Fuseaction diagram for BAI

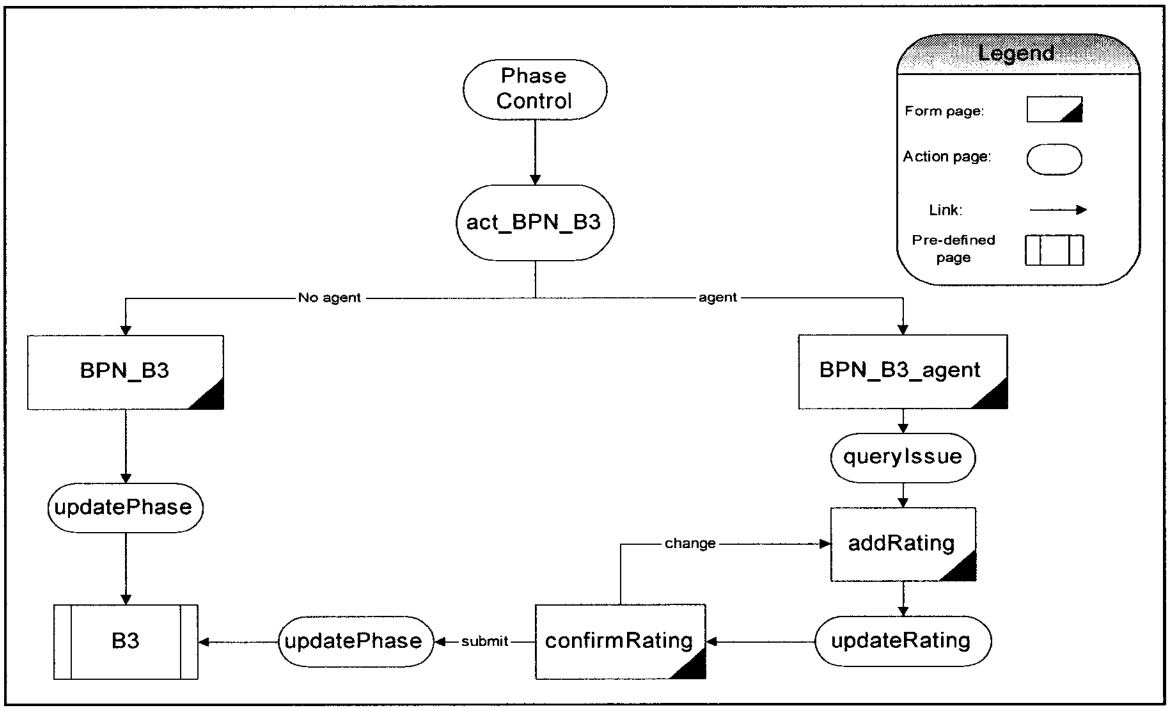


Figure 48. Fuseaction diagram for BPN_B3

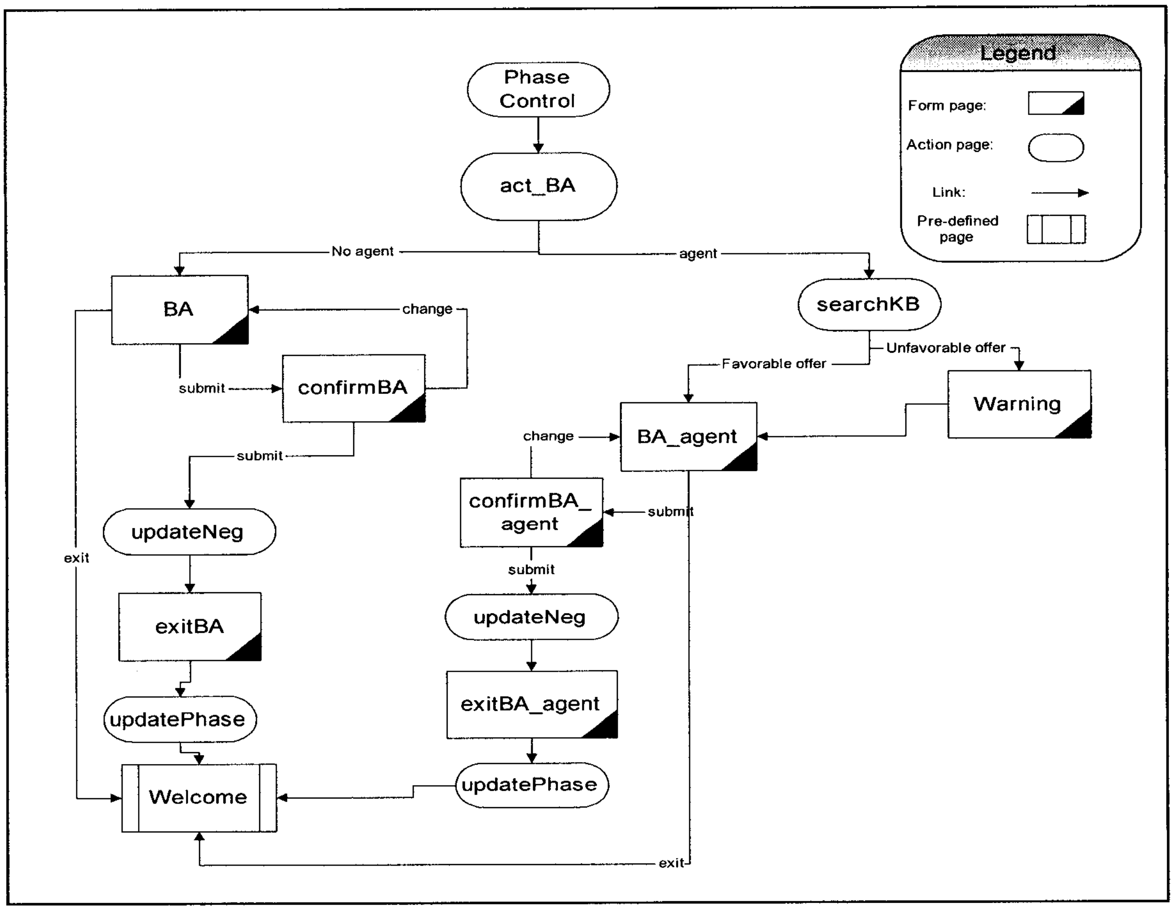


Figure 49. Fuseaction diagram for BA

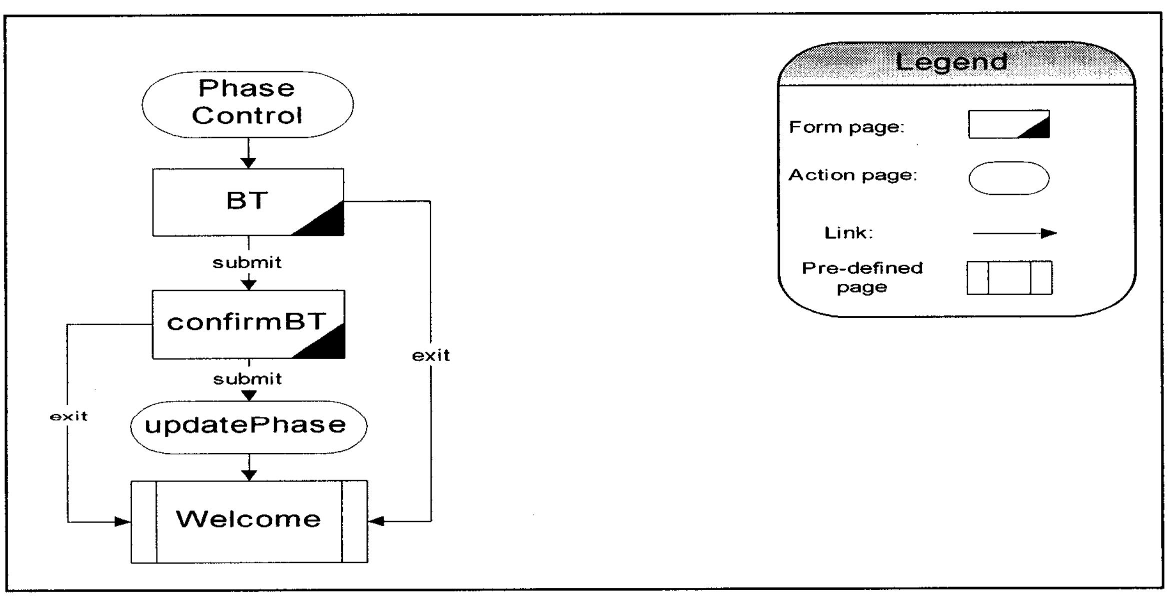


Figure 50. Fuseaction diagram for BT

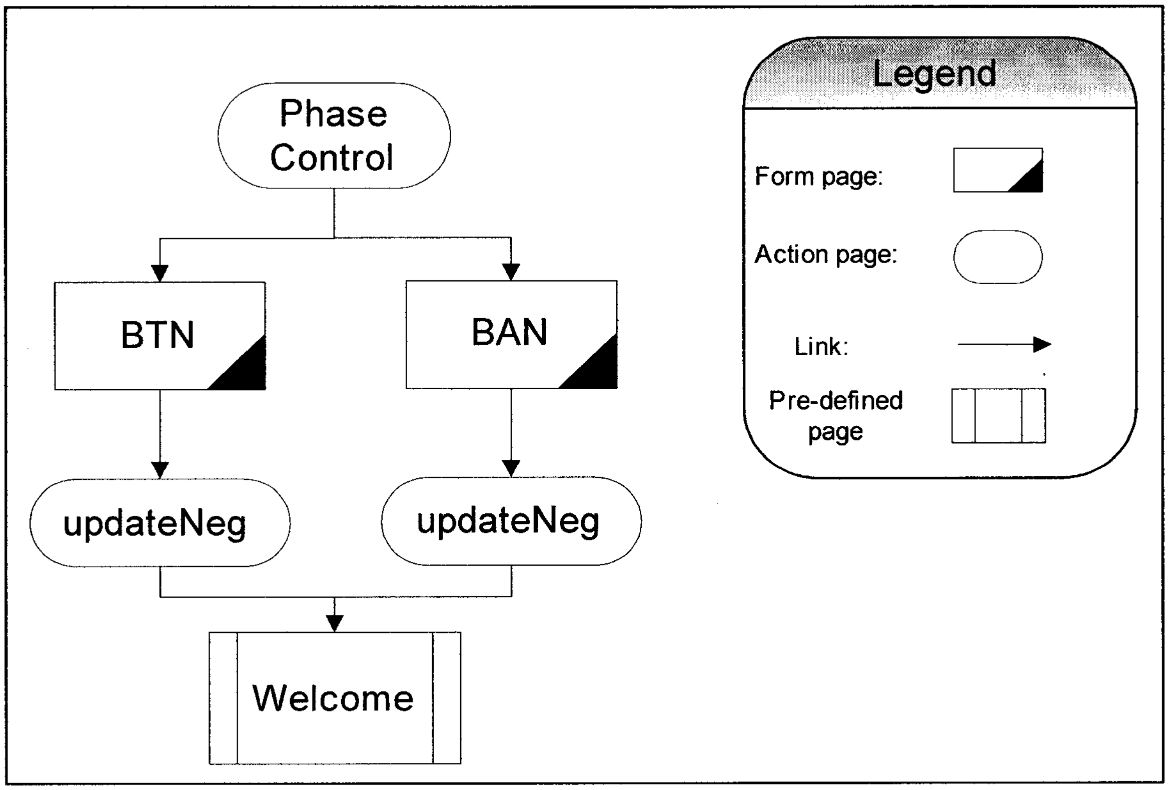


Figure 51. Fuseaction diagram for BAN and BTN

Appendix C: Snapshots of eAgora

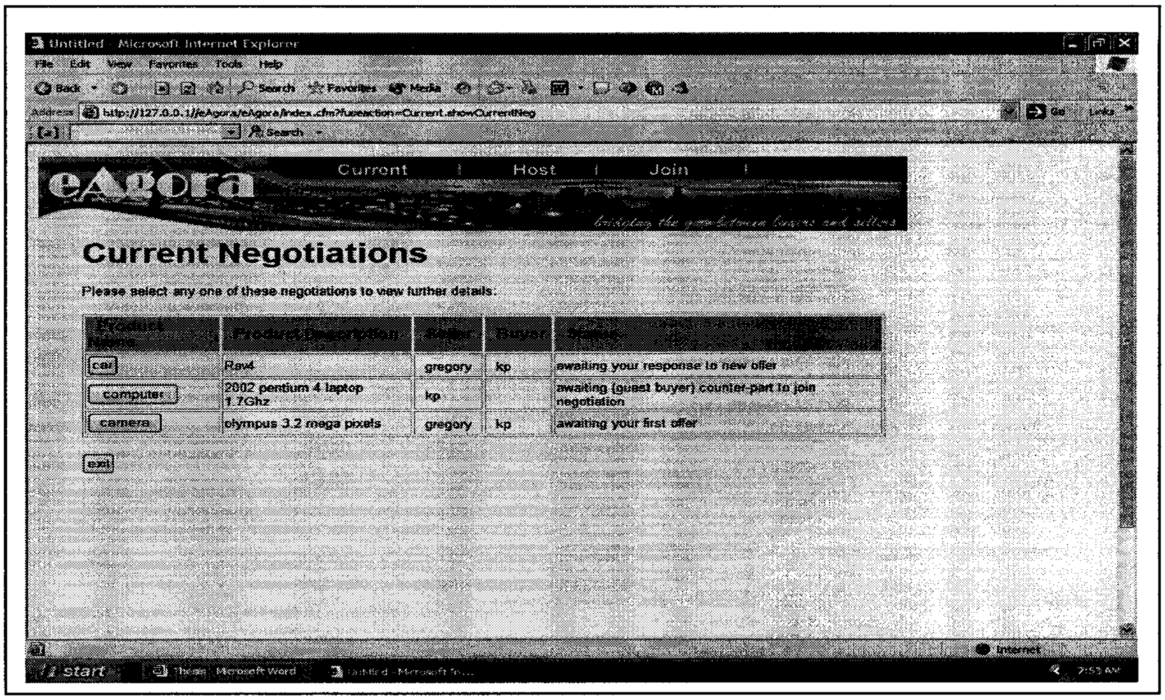


Figure 52. Current negotiations of one user

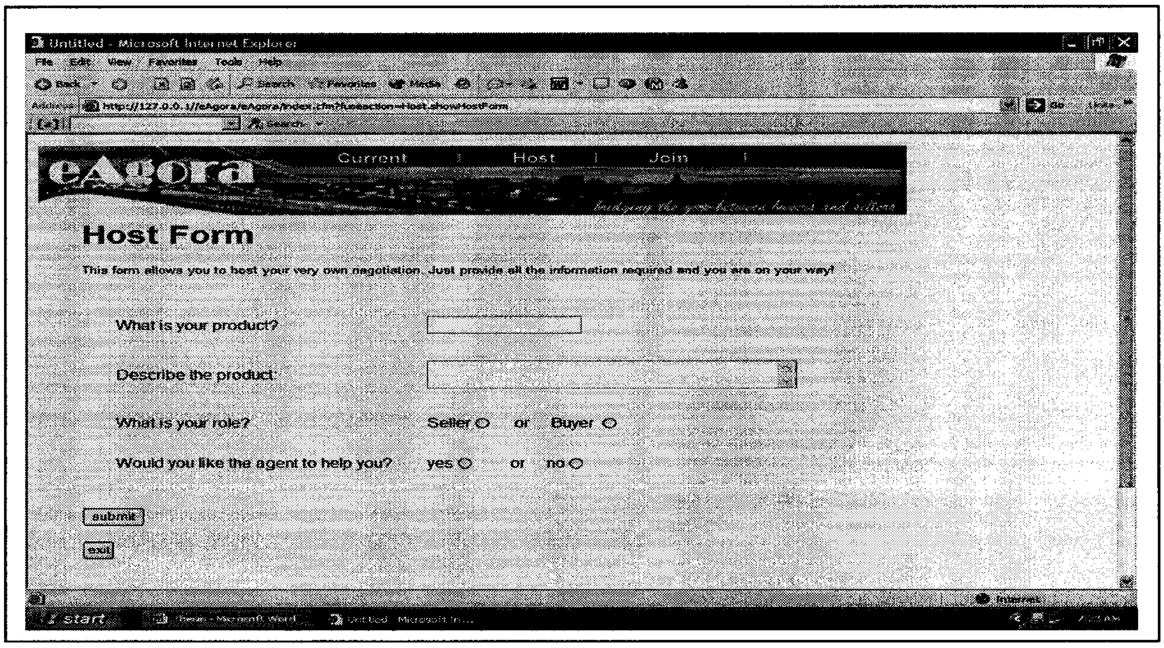


Figure 53. Hosting form

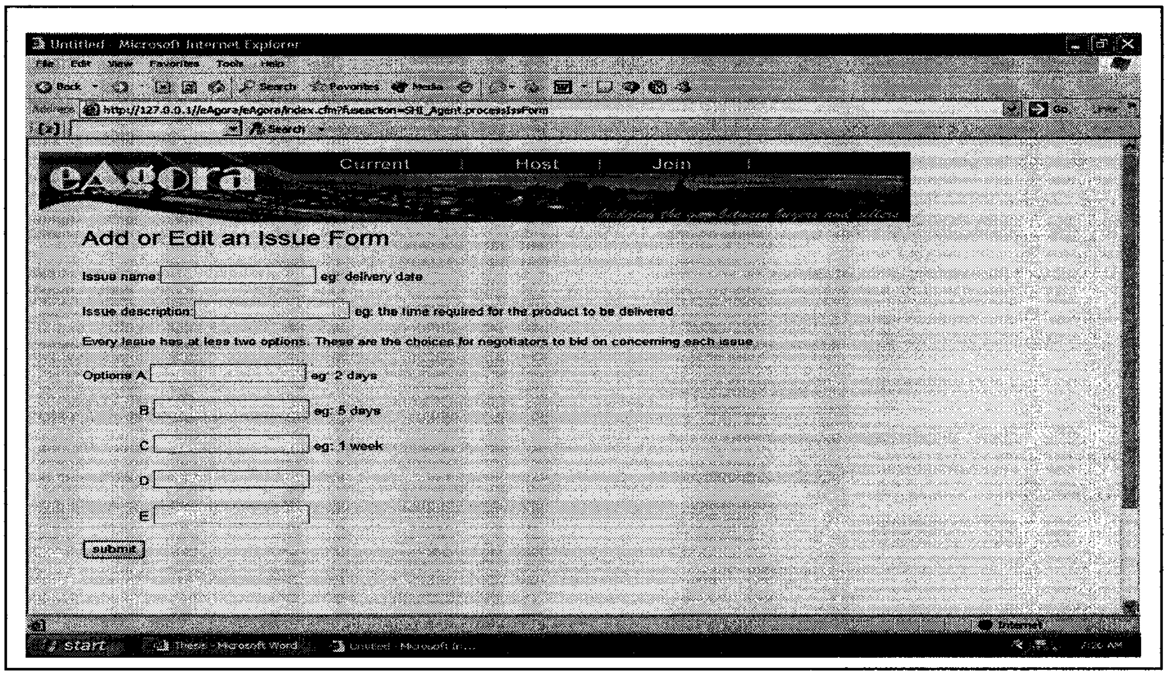


Figure 54. Issue creation form

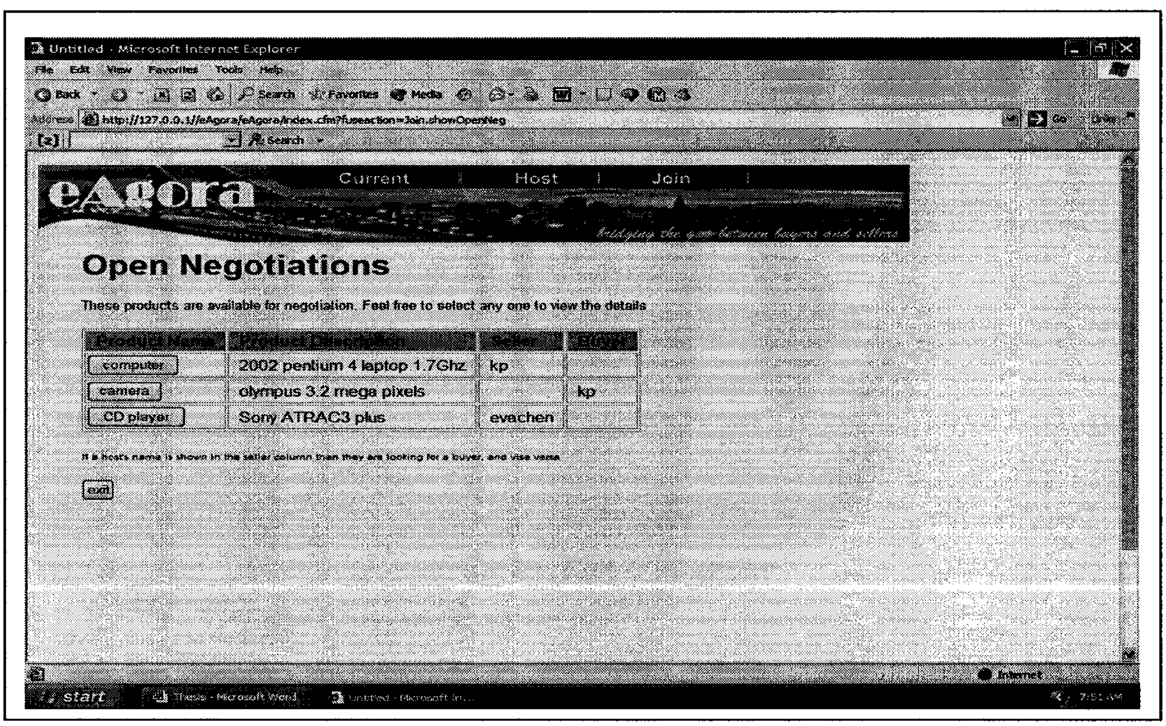


Figure 55. Opened negotiations available to join

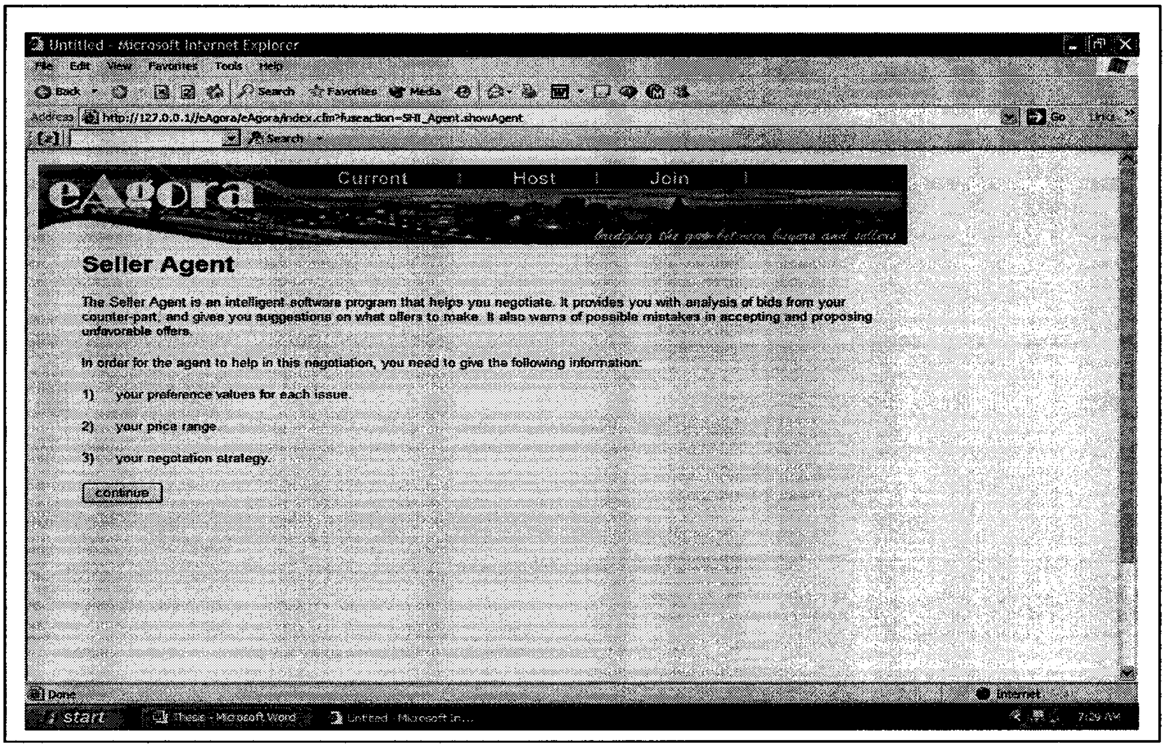


Figure 56. Seller Pre-negotiation: Introduction to agent

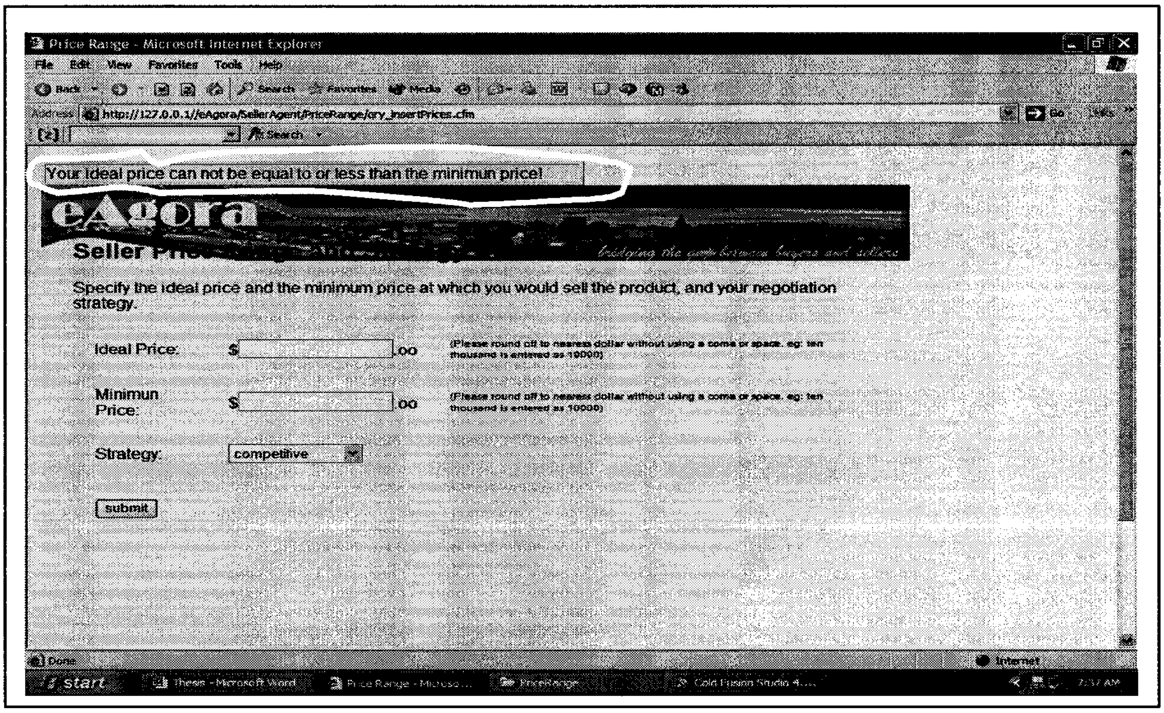


Figure 57. Warning of pre-negotiation setting error

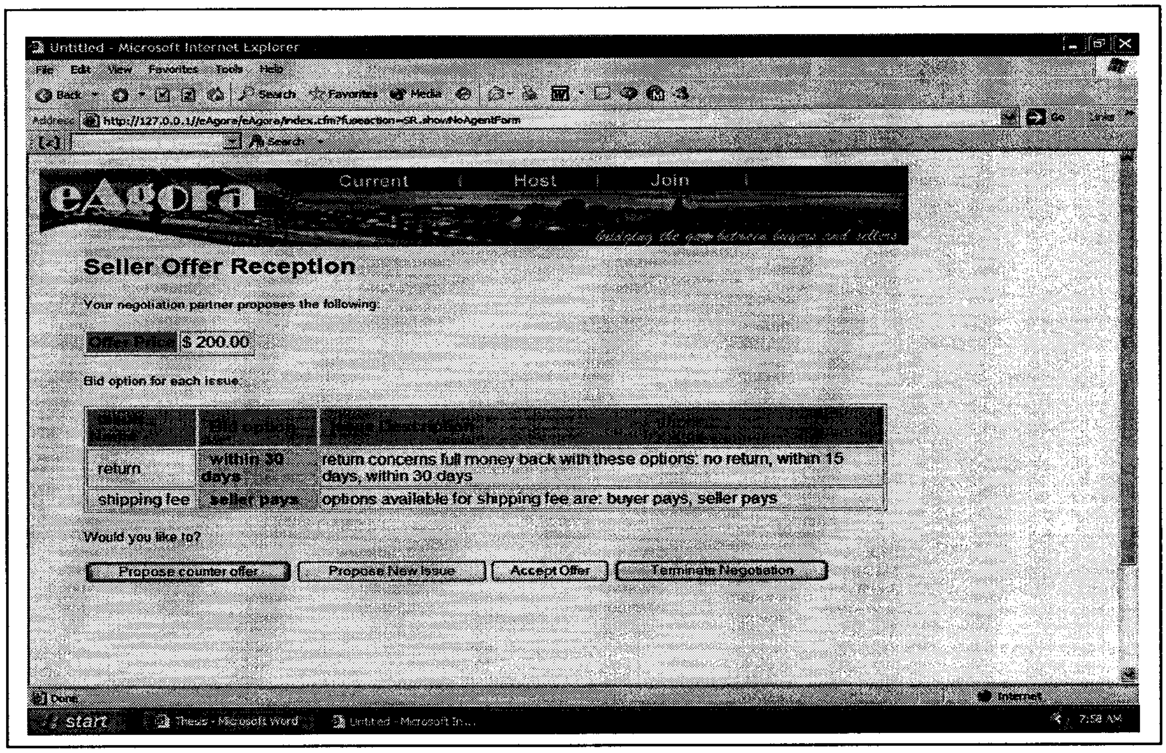


Figure 58. Counter-offer reception without agent

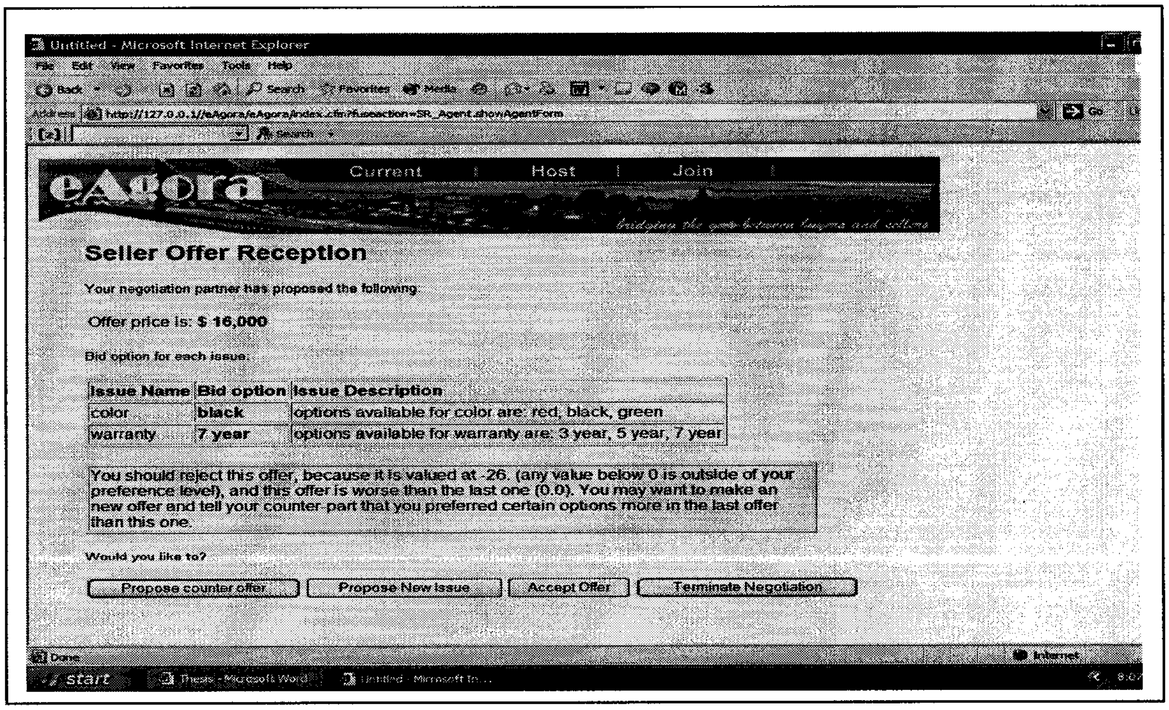


Figure 59. Counter-offer reception with agent

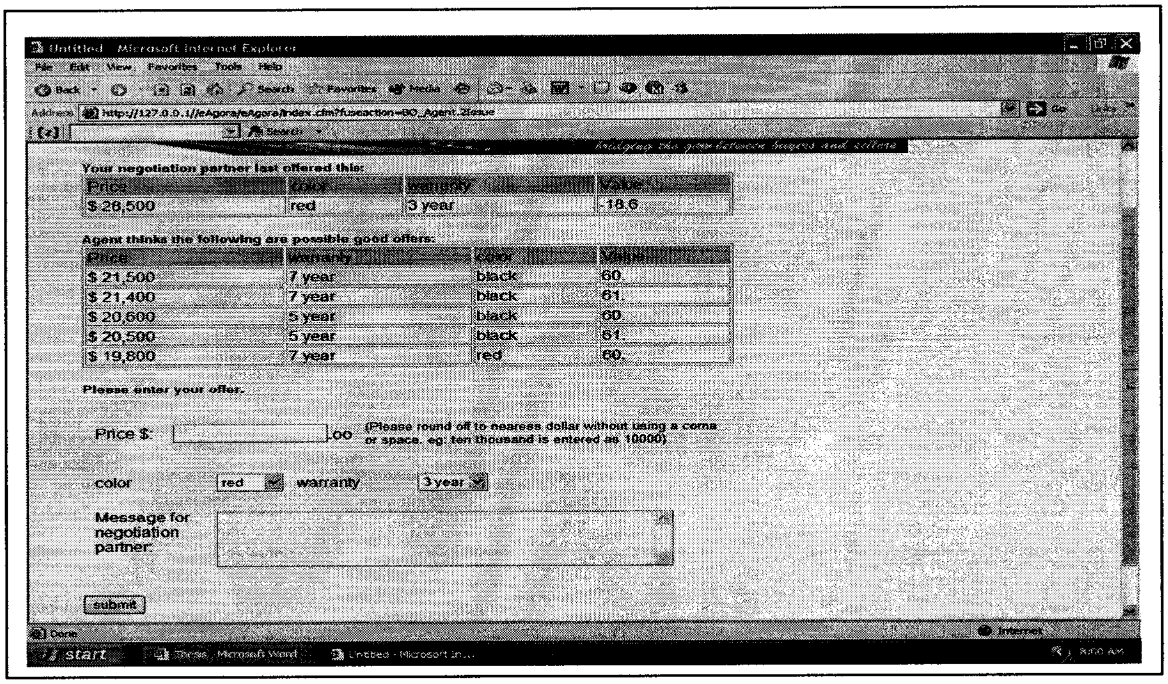


Figure 60. Offer formulation with agent

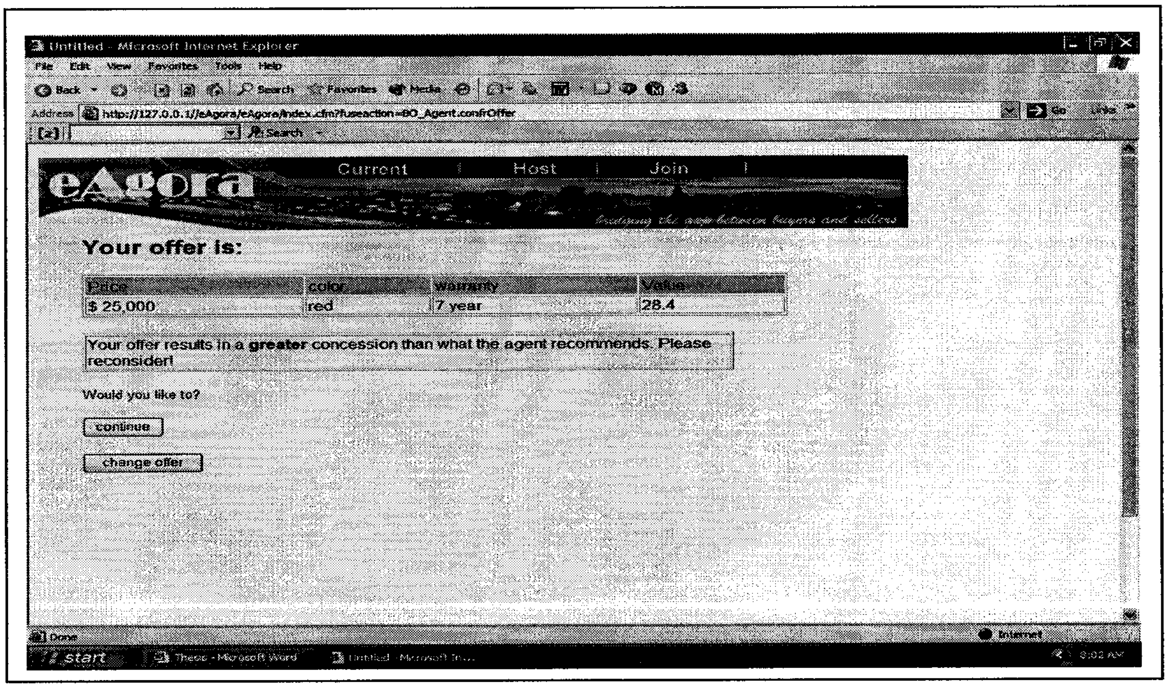


Figure 61. Warning in offer proposal

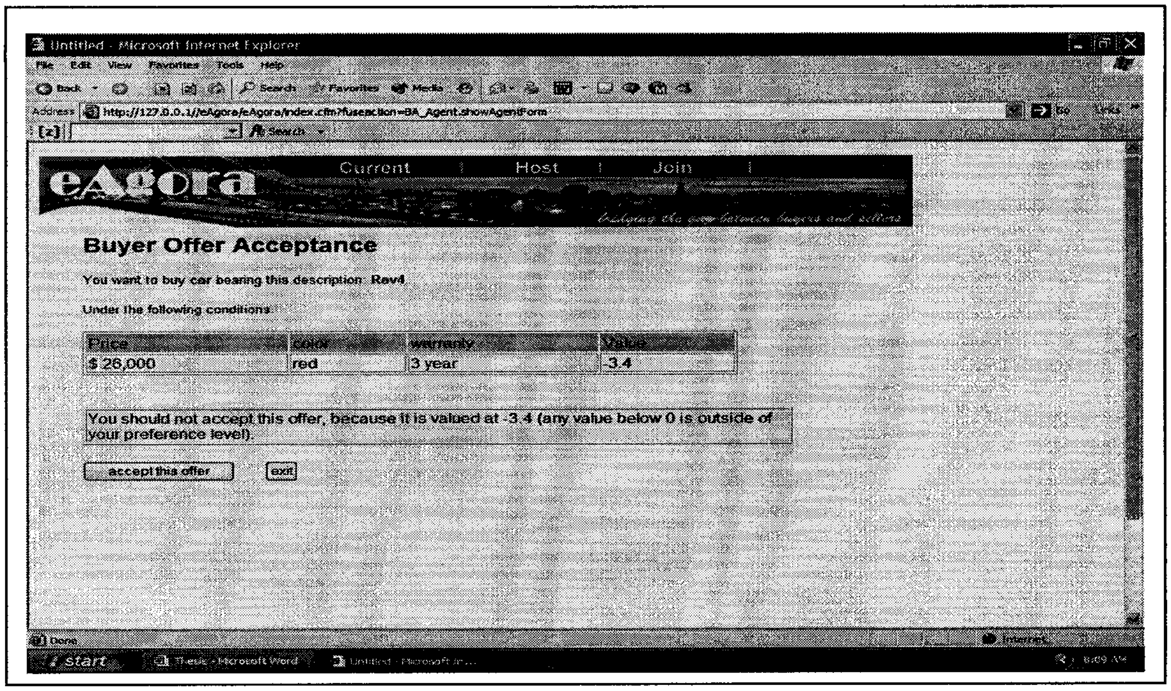


Figure 62. Warning of accepting an offer that violates pre-negotiation settings

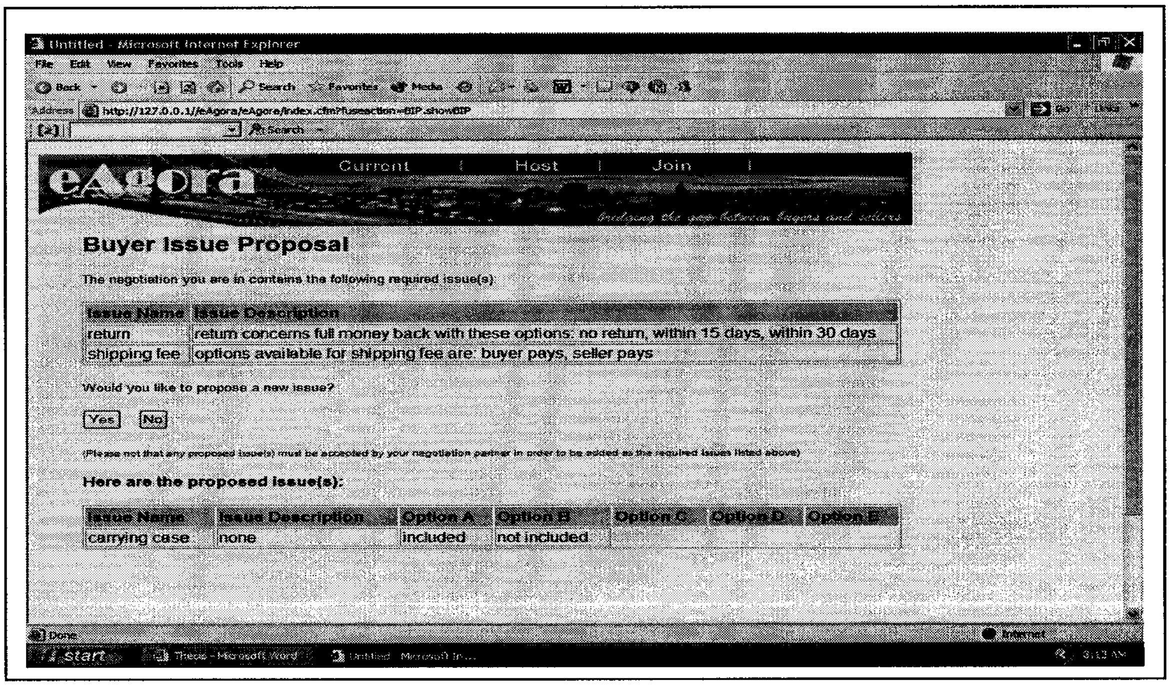


Figure 63. Propose issue during negotiation

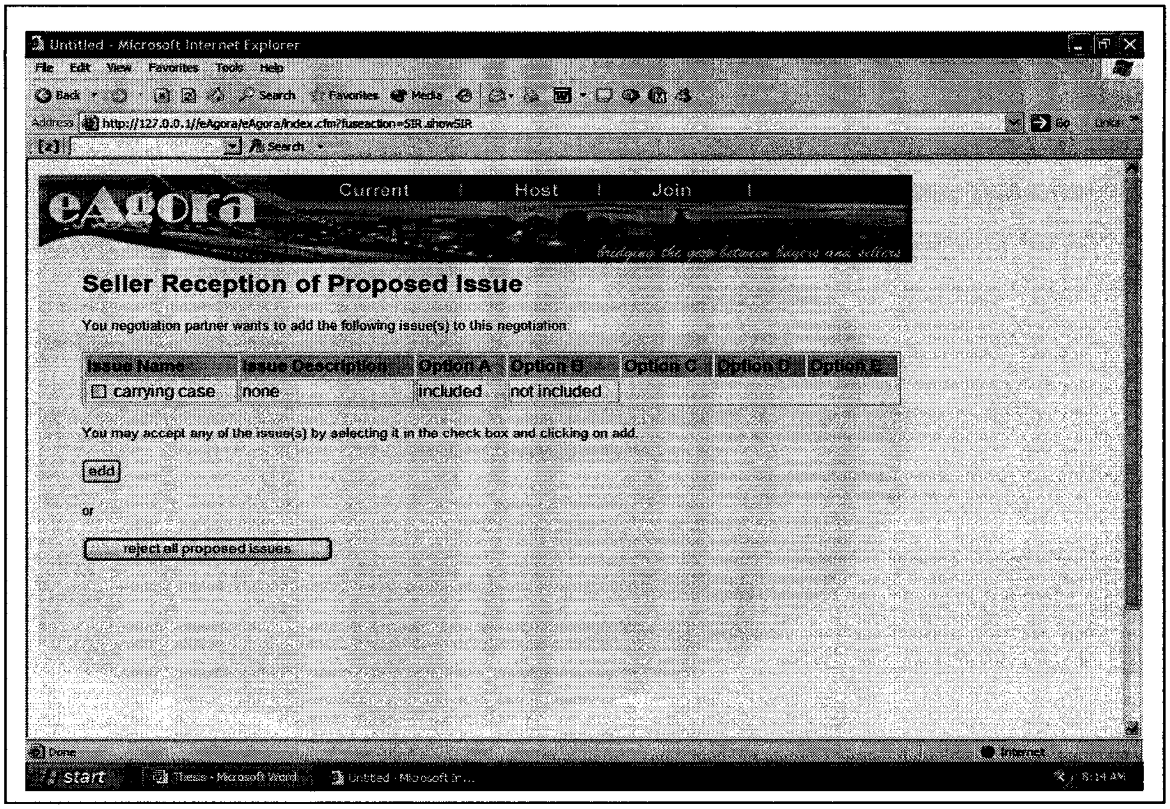


Figure 64. Proposed issue(s) reception

Appendix D: Rules for Offer Formation

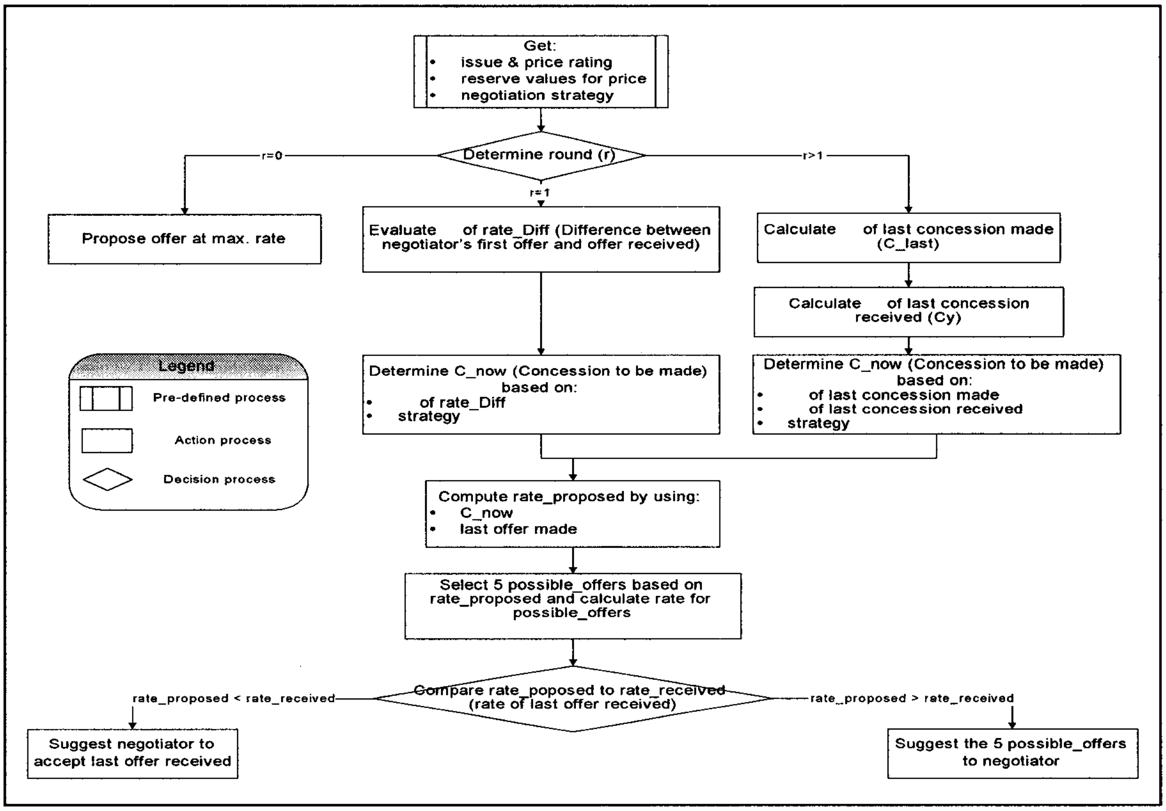


Figure 65. Rules for offer formation

Get variables

Example for Buyer:

Strategy: Competitive

Price: 70%

Price ideal: $100 = 100%
Price reserve: $200 = 0%

Delivery date: 20%

5 days: 0%
3 days: 50%
2 days: 100%

Return in 30 days: 10%

No: 0%
Yes: 100%

$$\left.\begin{aligned}&U_x = Rate\_price[(price_x - price\_reserve)/(price\_ideal - price\_reserve)] + \\&Rate\_dd(rate\_5days, rate\_3days, rate\_2days) + \\&Rate\_return(rate\_no, rate\_yes)\end{aligned}\right\} \begin{matrix} x1 \\ x2 \\ x3 \end{matrix}$$

U*x*	**Price ($)**	**DD (days)**	**Return**
100	100	2	Yes
...	...	...	...
0	200	5	No

Table 3. Matrix of possible offer packages

X={ x1, x2, x3}

Offer at r=0

U_max = U*x* at 100%

U*x*	**Price ($)**	**DD (days)**	**Return**
100	100	2	Yes

Table 4. Buyer's offer at round=0

Offer at r=1

Calculate U*y* (based on opponent's offer ***Y={y1, y2, y3}***)

U*y*	**Price ($)**	**DD (days)**	**Return**
0	210	5	No

Table 5. Seller's counter-offer

rate_Diff = U_max – U*y*

Determine size of rate_Diff (i.e. degree of membership (μ))

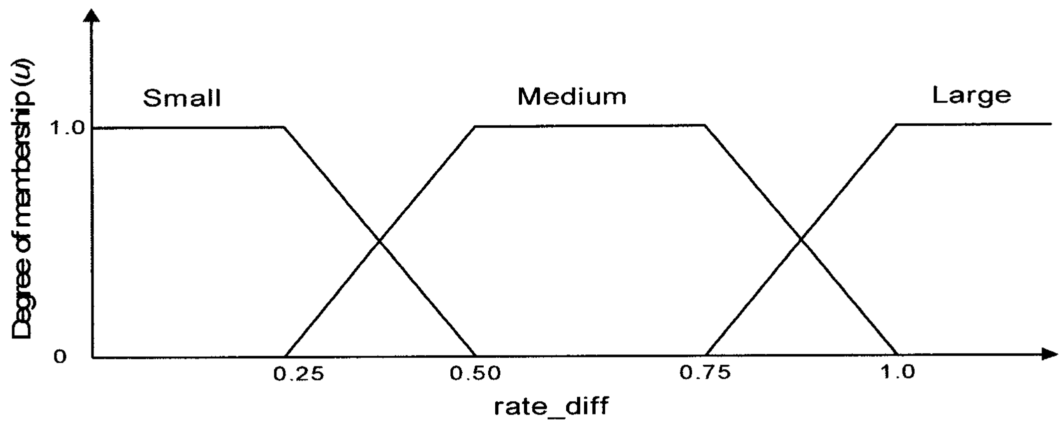


Figure 66. Fuzzy logic graph for rate_diff

If $0 < \text{rate\_Diff} < 0.25$, Then:

- μ_small = 1
- μ_medium = 0
- μ_large = 0

If $0.50 < \text{rate\_Diff} < 0.75$, Then:

- μ_small = 0
- μ_medium = 1
- μ_large = 0

If rate_Diff =1, Then:

- μ_small = 0
- μ_medium = 0
- μ_large = 1

If $0.25 < \text{rate\_Diff} < 0.50$, Then:

- μ_small = (0.50 – rate_Diff)/ (0.50 - 0.25)
- μ_medium = (rate_Diff – 0.25)/ (0.50 – 0.25)
- μ_large = 0

If $0.50 < \text{rate\_Diff} < 0.75$, Then:

- μ_small = 0
- μ_medium = (1 – rate_Diff)/ (1 – 0.75)
- μ_large = (rate_Diff – 0.75)/ (1 – 0.75)

Determine the concession to be make (C_now) based on the strategy chosen

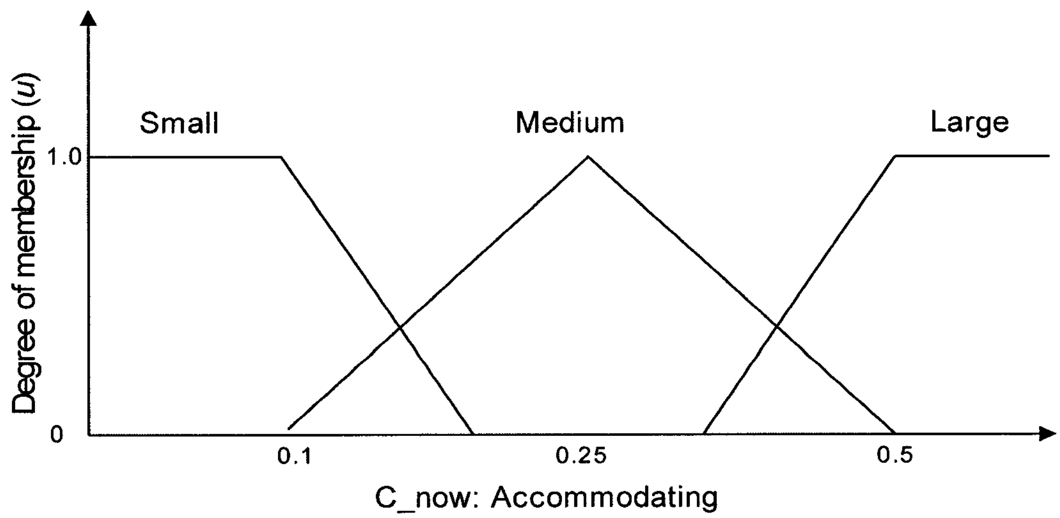


Figure 67. Fuzzy logic graph for C_now: Accommodating

C_now for the different strategies:

$$\text{Accommodating} = \frac{0.1(\mu\_small) + 0.25(\mu\_medium) + 0.5(\mu\_large)}{\mu\_small + \mu\_medium + \mu\_large}$$

$$\text{Compromising} = \frac{0.05(\mu\_small) + 0.2(\mu\_medium) + 0.4(\mu\_large)}{\mu\_small + \mu\_medium + \mu\_large}$$

$$\text{Competitive} = \frac{0.03(\mu\_small) + 0.15(\mu\_medium) + 0.3(\mu\_large)}{\mu\_small + \mu\_medium + \mu\_large}$$

Calculate the proposed rate for suggesting offer packages

rate_proposed = U_last– C_now (at r=1, U_last (i.e. rate of last offer) =U_max)

Find five best offer packages at rate_proposed

Suggest these packages on if rate_proposed is greater than rate_received

Offer at r>1

C_last = C_now in the last round (at r-1)

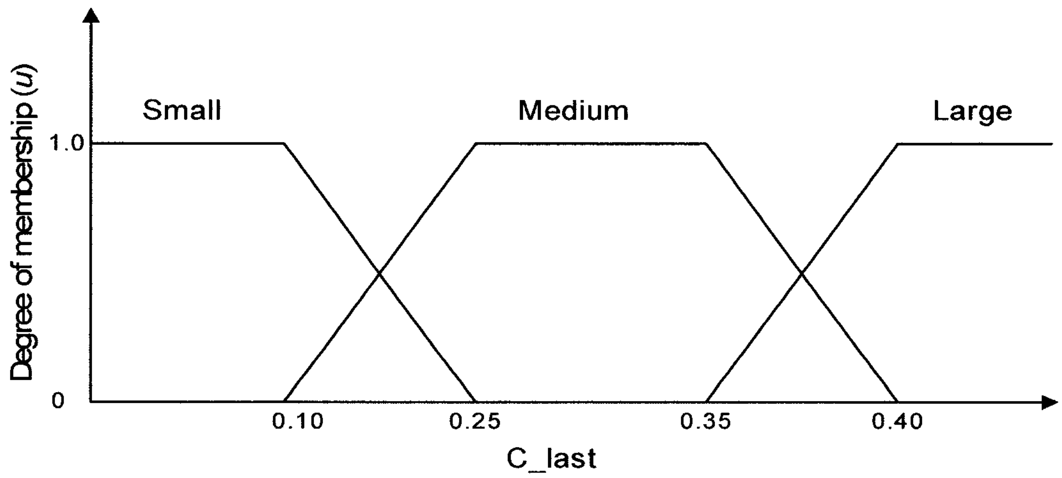


Figure 68. Fuzzy logic graph for C_last

If 0 < C_last < 0.10, Then:

- μ_small = 1
- μ_medium = 0
- μ_large = 0

If 0.25 < C_last < 0.35, Then:

- μ_small = 0
- μ_medium = 1
- μ_large = 0

If C_last >0.40, Then:

- μ_small = 0
- μ_medium = 0
- μ_large = 1

If 0.10< C_last < 0.25, Then:

- μ_small = (0.25 – C_last)/ (0.25 - 0.10)
- μ_medium = (C_last – 0.10)/ (0.25 – 0.10)
- μ_large = 0

(if μ_small < μ_medium, then μ_medium =0)
(if μ_small > μ_medium, then μ_small =0)

If 0.35< C_last < 0.40, Then:

- μ_small = 0
- μ_medium = (0.40 – C_last)/ (0.40 - 0.35)
- μ_large = (C_last – 0.35)/ (0.40 – 0.35)

(if μ_medium< μ_large, then μ_large =0)
(if μ_medium> μ_large, then μ_medium =0)

Determine the concession made by opponent

Cy = (Uy at r) – (Uy at r-1)

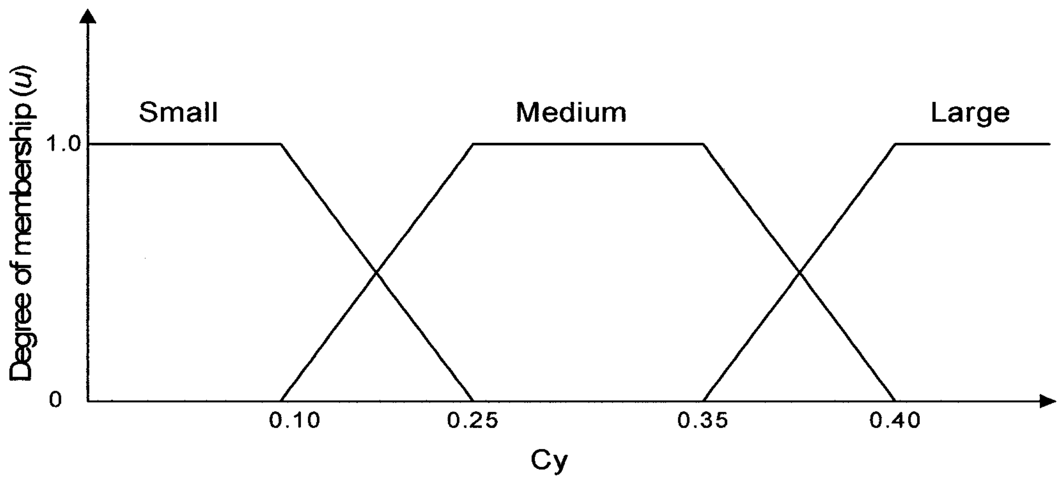


Figure 69. Fuzzy logic graph for Cy

If 0 < Cy < 0.10, Then:

- μ_small = 1
- μ_medium = 0
- μ_large = 0

If 0.25 < Cy < 0.35, Then:

- μ_small = 0
- μ_medium = 1
- μ_large = 0

If Cy >0.40, Then:

- μ_small = 0
- μ_medium = 0
- μ_large = 1

If $0.10 < Cy < 0.25$, Then:

- $\mu\_small = (0.25 - Cy)/(0.25 - 0.10)$
- $\mu\_medium = (Cy - 0.10)/(0.25 - 0.10)$
- $\mu\_large = 0$

(if $\mu\_small < \mu\_medium$, then $\mu\_medium = 0$)
(if $\mu\_small > \mu\_medium$, then $\mu\_small = 0$)

If $0.35 < Cy < 0.40$, Then:

- $\mu\_small = 0$
- $\mu\_medium = (0.40 - Cy)/(0.40 - 0.35)$
- $\mu\_large = (Cy - 0.35)/(0.40 - 0.35)$

(if $\mu\_medium < \mu\_large$, then $\mu\_large = 0$)
(if $\mu\_medium > \mu\_large$, then $\mu\_medium = 0$)

C_now for the different strategies:

μ_small = μ_small from C_last + μ_small from C_opponent

μ_medium = μ_medium from C_last + μ_medium from C_opponent

μ_large = μ_large from C_last + μ_large from C_opponent

$$\text{Accommodating} = \frac{0.1(\mu\_small) + 0.25(\mu\_medium) + 0.5(\mu\_large)}{\mu\_small + \mu\_medium + \mu\_large}$$

$$\text{Compromising} = \frac{0.05(\mu\_small) + 0.2(\mu\_medium) + 0.4(\mu\_large)}{\mu\_small + \mu\_medium + \mu\_large}$$

$$\text{Competitive} = \frac{0.03(\mu\_small) + 0.15(\mu\_medium) + 0.3(\mu\_large)}{\mu\_small + \mu\_medium + \mu\_large}$$

Calculate the proposed rate for suggesting offer packages

rate_proposed = U_last– C_now

Find five best offer packages at rate_proposed

Suggest these packages on if rate_proposed is greater than rate_received

Appendix E: Questionnaire for Usability Test

Part I: Pre-Test

1. How many hours a week do you spend on the Internet? ____________________
2. Do you search for products on the Internet? ____________________
3. Have you bought any thing from the Internet? ____________________
4. Have you used an auction or negotiation Website? ____________________
5. Did you complete a successful transaction on an auction or negotiation site? ____

Part II: Post-Test **(Without agent)**

1. Would you use eAgora to sell or buy products? ___________________
2. Do you think that eAgora is easy to use? ___________________
3. Did you reach an agreement without the agent? __________ If not, why?

____

____

4. How would you rate your eAgora experience overall (0 poor, 1 acceptable, 2 good, 3 very good, 4 great and 5 excellent)? _____

5. What did you like about eAgora? ____

____

6. What did you dislike about eAgora? ____

____

Part III: Post-Test (**With agent**)

7. Did you think the agent helped you in negotiations? _________________
8. Did you receive any warnings/alerts from the agent? _________________
9. Did you think that the offer suggestions provided by the agent were helpful (yes/no)? ____ . Did it affect your decision? ____ Please explain.

____

____

10. How would you rate the agent's features overall (0 poor, 1 acceptable, 2 good, 3 very good, 4 great and 5 excellent)? _____

11. Did you reach an agreement with the agent? __________ If not, why?

____

____

12. What did you like about the agent? ______________________________________

____

13. What did you dislike about the agent? ___________________________________

____

14. Did you think you were always in control of the negotiation? ______If not, why?

____

Appendix F: Table of Results for Usability Test

Pre-Test Questions	
1. Average hour per week that each participant spends on the Internet.	17 hours/week
2. Percentage of participants that have searched for products over the Internet.	100 %
3. Percentage of participants that have bought products over the Internet.	83 %
4. Percentage of participants that have engaged in e-negotiation prior to eAgora.	17 %
5. Percentage of participants that have completed a successful e-negotiation transaction prior to eAgora.	17 %
Post-Test Questions (Without agent)	
1. Percentage of participants that would use eAgora to sell or buy products.	92 %
2. Percentage of participants that claimed eAgora is easy to use.	100 %
3. Percentage of participants that have reached an agreement using eAgora. (Participants that did not reaching an agreement cited the reason to be: could not come to a compromise between the price and issues)	67 %
4. Average rating for participants' satisfaction with eAgora.	3.3 rating
5. eAgora's most[5] cited advantages:	
▪ Easy to use. ▪ Can add issues during negotiation.	75 % 50 %
6. eAgora's most cited disadvantages:	
▪ Does not have any help functions. ▪ Does not allow for different protocols. ▪ Cannot upload pictures to show product.	67 % 50 % 75 %

[5] Derived from a percentage of respondents greater than or equal to 50 %.

(With agent)	
7. Percentage of participants that claimed the agent is helpful during negotiations.	83 %
8. Percentage of participants that received warnings/alerts from the agent.	100 %
9. Percentage of participants that claimed the agent's offer suggestions are helpful.[1] (All of which responded that their decisions were affected by the suggestions)	83 %
10. Average rating for participants' satisfaction with the agent's features.	2.8 rating
11. Percentage of participants that have reached an agreement using the agent. (Participants that did not reached an agreement cited the reason to be: could not come to a compromise between the price and issues)	83 %
12. Agent's most cited advantages:	
▪ Provides helpful offer suggestions. ▪ Can analyze offers received.	67 % 67 %
13. Agent's most cited disadvantages:	
▪ Cannot adjust pre-negotiation settings during negotiation. ▪ Does not help users select a strategy. ▪ Does not have an entity.	50 % 50 % 75 %
14. Percentage of participants that felt in control of negotiations.	100 %

Table 6. Results from usability test

[1] The comments are stated in Section 8.2.