Agent Assisted Price Negotiation For Electronic Commerce

Patrick Desharnais

A Thesis
in
The Department
of
Computer Science

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

April 2000

0-612-54330-7

ABSTRACT

Agent Assisted Price Negotiation for Electronic Commerce

Patrick Desharnais

Current electronic retail stores do not offer one-to-one price negotiation capabilities. From a consumer's perspective, price negotiation provides an opportunity to debate the price. From a vendor's perspective, the ability to negotiate allows for flexibility in pricing that a rigidly fixed price policy cannot offer. Hence, we feel that one-to-one price negotiation would be beneficial in online stores and e-commerce in general. Given the high cost of providing human sales agents online, we research if it is possible to design a practicable automated system that is able to autonomously negotiate on behalf of a retail vendor in a commercial one-to-one business environment.

Specifically, this thesis explores the use of "agent assisted price negotiation" applicable between a consumer and a retail vendor. It describes the inherent difficulties involved in automating negotiation, provides a critical analysis of the current approaches to automated negotiation with regards to different business models, and then proposes an "information driven" methodology for the calculation of a "just-in-time personalized price". The thesis also provides the requirements and specifications for a simple and intuitive one-to-one negotiation protocol. As a proof of concept, a Java prototype of a software Sales Agent in a Multi-Agent System architecture is implemented and presented. Overall, by automating negotiation for e-commerce retailing, we hope to increase both the retailer and consumer satisfaction.

AKNOWLEDGMENT

I would like to dedicate this thesis to my wonderful father, who gave me the initial inspiration for my research. Throughout my degree, he never stopped challenging me with constructive questions about my work. Our long discussions gave me good occasions to verify the grounds on which I was building my thesis. Thanks dad. I would also like to thank all my family, my mom, my brother, and my grandfather for believing in me and encouraging me all the way. Special thanks to Isabelle, for thinking the best of me and for having supported me almost everyday.

I would also like to convey my deepest gratitude to my thesis supervisor, Dr. T. Radhakrishnan, for I am very thankful for his guidance and constant support throughout my studies. I am grateful for the confidence he put in me, and for the numerous compliments and encouragements he gave me. His opinion of me is greatly valuated. Mostly, I am very thankful for the opportunity, I had, to meet with Dr. T. Radhakrishnan. Throughout a series of personal experience such as the conferences in China and Toronto, I had the chance to discover a man with good values and integrity, and for whom I have great respect. I also acknowledge the financial support he provided to me through the Canadian Institute for Telecommunications Research grant.

Last but not least, I wish to thank my colleagues at the Multimedia Laboratory of Concordia, especially Ji Lu, Venkat and Rony. By doing our thesis at the same time, we've shared similar experienced. It gave me great comfort in the fact that I was not alone in this process.

Table of Contents

List of Figures

List of Tables

Chapter 1
Introduction

"The future of electronic commerce is an implicit one-to-one negotiation between buyers and sellers." Jerry Kaplan, Onsale Inc.

In physical retail markets, prices are often fixed and not subject to negotiation. Reasons for this include the fact that it is often more convenient and cheaper for a retailer to fix a price only once, than to participate in personalized pricing or engage in one-to-one price negotiation with every potential customer. However, one should note that the notion of fixed list prices is a relatively recent development in our history and, although cheap and convenient, it is a method of pricing that doesn't offer much flexibility. The fact that, even in these so-called *fixed-price* markets, price negotiation still occurs from time to time, such as in buying card, leads us to believe that flexibility in pricing is a desirable thing.

Still, the online[1] counterparts of these markets have yet to provide means for a personalized one-to-one pricing scheme. One reason for this is that supplying online price negotiation comes with greater costs. In the physical world, the cost of bargaining for a retail vendor is to hire and train one or more sales persons to handle negotiations with the customers. However, this cost is partly absorbed by the fact that sales persons are already needed to carry out transactions and manage the store. However in the electronic world, there is no direct human presence behind online stores. Supplying such a presence would mean implementing a real-time communication infrastructure and providing enough sales persons to handle price negotiations with a potentially large number of remote customers, all this on a 24 hours a day basis. This is obviously an expensive solution. An alternative would be to communicate with a sales representative via email, but this approach would not satisfy the real-time needs of consumers and would still require costly human presence.

[1] We define online as a synonym of web based, i.e. whereas the web is the used for the display and exchange of information.

At this point, the following question arises: "In the context of electronic-commerce, can technology be used to replace humans in providing individual pricing and one-to-one price negotiation, thus adding the desired flexibility in pricing without the high cost usually associated with it?" Finding an answer to this question is extremely challenging, as it involves both the fields of negotiation and economics. It is even more challenging considering that over forty years of intense theoretical research in these two areas have failed to produce adequate computational models of the problem [Linhart 92]. As distressing as this fact may sound, the recent and active field of intelligent software agents looks promising to address the problem. Given the cost of providing a human sales agent behind an online store, it be nice if a vendor[2] could delegate the task of price negotiation to an autonomous and personalized software sales agents? A growing body of research results in the field of intelligent agent [Chavez 96] [CMIT-1016] [Wurman 98] and Negotiation Support Systems [Jelassi 89] [Rangaswamy 97] suggests such a thing could be possible in the near future.

1.1 Background

There is general consensus that negotiation plays an important role in today's business processes and that automated negotiation will be a key part of e-commerce in the future. Both researchers and the people in the business industry foresee great economic opportunities in providing the ability to negotiate online. However, the road to profit is filled with obstacles, as automating negotiation is clearly not trivial due to the complex nature of negotiation. Beam, Segev and Shanthikumar [CMIT-1019] summarize the situation:

> *"The inability to negotiate represents a large business need, however, electronic negotiation is a difficult complex process, and success with it is limited at best."*

In this context, this section describes the inherent difficulties of the negotiation problem, provides an overview of the ever-growing area that is e-commerce, clarifies what a software agents is, and finally presents a quick introduction of the type of agent mediated automated price negotiation we have in mind.

---

[2] In this thesis, the terms "vendor" and "merchant" will be used synonymously for the term "retailer".

1.1.1 The Negotiation Problem

The limited success of providing a computational model of the negotiation problem gives us an idea of the complexity of the activity. Problems resolved by negotiation can be described as "*non-algorithmic, their solutions obtained by hazardous process, and their results difficult to evaluate*" [Matwin 91]. Typically in negotiation, there is an issue[3] of conflict that the parties seek to resolve, but the process of resolving the conflict is difficult because, during the negotiation process, each party tries to maximize their own share of the deal. Furthermore, parties are in competition and do not know the other's utility function, as each party usually keeps its valuation private. In these conditions, the party that speaks first is at a disadvantage because it reveals information about its valuation by doing so. Jim Oliver notes that "*in fact, the situation is often even worse, as each side typically have incentive to misrepresent their preferences*" [Oliver 96].

In addition to these difficulties inherent to negotiation, designers of automated systems must also cope with new difficulties, such as representing in bits and bytes the negotiators' preferences and strategy, or addressing the fact that computational power can be used as a tool to infer one's negotiation strategy, something that isn't much of a concern in human negotiations.[4]

1.1.2 Electronic Commerce

According to [Sokol 89], "*Electronic Commerce is the sharing of information using a wide variety of different electronic technologies, between organizations doing business with one another... [it includes] also procedures, policies, and strategies to support incorporation of these electronic messages into the business environment*". No one can deny that e-commerce has experienced a fabulous growth since Sokol came up with this definition in 1989. However, e-commerce is not a new thing and has been around for more than 20 years in the form of the EDI (Electronic Data Interchange). But it is only when the Internet became increasingly popular and available that e-commerce numbers started to grow exponentially.

---

[3] The issue can be as trivial as which child gets to use dad's car on Friday or as complex as bringing two countries to make peace.

[4] Such difficulties will be covered in more details in Chapter 3.

With more than $8 billion generated revenues in retail commerce for 1997 ($600 million in 1996) and predictions that go up to $327 billion by 2002 in the U.S. alone, the World Wide Web has become a new and promising place to sell and buy goods. Customers now have a quick way to remotely access and compare information on products and vendors and, if they wish to, buy online without even leaving their home. For their part, merchants have a new window of opportunity to show themselves and their product to a large population, for the purpose of either selling their products online or at their sales' points.

According to Guttman and Maes in [Guttman 98a], "*online markets are more efficient than their physical-world counterparts thus lowering transaction costs for both merchants and consumers.*" It is the belief of the authors that online marketplaces are an opportunity to retail merchant because "*they offer traditional merchants and additional channel to advertise and sell products to consumers, thus potentially increasing sales.*" However, they are also a threat to these merchants because consumers can easily perform cross-merchant product comparisons with the help of emerging "comparison-shopping agents" and by using third party information systems.

Because of reduced search costs and efforts for the buyer, merchants need to differentiate themselves from their competitors. To cope with this, we believe technology can help these merchants to provide the desired differentiation in pricing. In this context, our research addresses "just-in-time personalized pricing" and "one-to-one negotiation" scheme that can cope with the situation. Moreover, differentiation is facilitated by the fact that individual electronic transactions can be kept secret. As a consequence, occurring and past transactions will not influence future ones. In the real world, there is always the possibility that someone will eavesdrop on a transaction taking place and use such information while haggling on the price.

1.1.3 Software Agents

Although we are in a sense familiar with the concept of an agent, there is no universal definition of the term "software agent". A generally accepted definition is that of an autonomous software component that performs tasks on behalf of a user or another agent.

But since this could be said of almost any software program, a software agent (often simply termed *agent*) should also possess some other desirable properties. Researchers have proposed several properties that could distinguish a software agent from conventional software [Maes 95] [Wooldridge 95] [Foner 97] [Etzioni 94]:

- *Autonomous:* An agent is an independent entity capable of reasoning on it's own. It is able to exercise a non-trivial degree of control over its own actions on behalf of its owner, without requiring explicit permission for every action.
- *Reactive:* An agent can sense changes to its dynamic environment (which may be the sensor inputs of the physical world, explicit user inputs, information gathered from another agent or from sources such as the Internet) and respond to those changes with suitable reactions.
- *Proactive:* An agent is goal-oriented and takes initiatives to fulfill its goals.
- *Persistent:* The notion of agency involves a sense of temporal continuity; an agent is usually continuously running.
- *Trustworthy:* An agent will only do what its owner expects it to do.
- *Personalized:* An agent can either learn or be explicitly taught what to do for each individual or group of users.
- *Social behavior:* An agent can interact with the user or other agents in order to best accomplish its goals.

The general debate in agent studies is in agreeing on "which of these characteristics are essential for a software system to be qualified as an agent". In addition, software agents are often referred to as "intelligent agents". Generally, agents are said to be intelligent if they incorporate some advanced behavior such as reasoning or planning, usually by making use of a knowledge base and an inference engine. Finally, although some software agents (such as Julia [Foner 97]) have been anthropomorphized, anthropomorphism is not considered by researchers as an essential requirement for a software agent.

1.1.4 Automating Price Negotiation through Software Agents

Price negotiation happens because there are situations when vendors feel it is in their own interest to participate in price negotiations. If only to attract more customers or to create more customer satisfaction, the expected payoff of selling at a price lower than the fixed price may be deemed worth it. More specifically, a vendor might be willing to negotiate the price in order to keep a profitable client satisfied, to make room in the inventory, to sell an item that has not been selling well recently, or just to make a definite sale at a lower price instead of a potential sale at the fixed price.

Our goal is thus to research how a knowledge driven software sales agent could recognize such situations, calculate the desired payoff and use this information to negotiate effectively with the consumer under a negotiation protocol. Whether or not such a system can be realistically designed is an open question that this thesis begins to address. But in order to answer this question, much is at stake. First, we need to determine what knowledge is needed. Moreover, the knowledge we use should be general enough to cover negotiation with any given customers and on the full range of products of a given vendor. Jim Oliver notes that "*a completely pre-specified approach is limited because a program that is too specific for a situation would need to be changed if the task changed or the environment changed*" [Oliver 96].

In addition, the calculation of the payoff must be tailored to individual vendors, as valuation is personal and varies from vendor to vendor. Therefore, we need to provide them with a way to input their own valuation functions. We note that extracting such knowledge from the vendor is clearly not trivial. Thence, the process chosen to do so should be as easy and intuitive as possible. Finally, we also need to determine the rules that will govern the negotiation process and find out the negotiation strategy to be used. All of these challenges are part of the motivation for our present research.

In conclusion, by providing a flexible pricing scheme through such negotiations, we hope to: (1) increase the vendor's revenue in the short term by allowing transactions that would not have happened otherwise; (2) increase vendor's revenue in the long term by keeping the customers satisfied and maintaining a long-term relationship with them; (3) increase customer satisfaction by catering to the pleasurable aspect of bargaining and getting deals. However, since we will not build a practicable commercial system, such increase in revenue and customer satisfaction will not be measured analytically. We rather argue that such things are likely to happen.

1.2 CITR

This thesis is part of a CITR (Canadian Institute for Telecommunication Research) research project: "Enabling Technologies in Electronic Commerce". The objectives of this project are: (1) developing fundamental enabling technologies for electronic commerce applications; (2) analysis of networks, computer systems and workflow architectures in view of supporting electronic commerce applications; (3) modeling and understanding the behavior and architecture of electronic commerce applications in order to guide the development of the required enabling technologies. The project consists of the following five major components:

- Development of interoperable multimedia virtual catalogs

  This sub-project addresses the multimedia electronic commerce catalog development issues as well as the interoperability of these repositories to enable users to access multiple, distributed and potentially heterogeneous catalogs in a uniform and transparent manner.

- System and network performance and project management

  This sub-project uses the deployed e-commerce application as a testbed to collect data and conduct measurements on application behavior.

- Quality of service and distributed systems management

  This sub-project addresses the management of quality of service and the adaptation of the application within such system and network environments.

- User interface and intelligent agents

  This sub-project investigates the design and integration of a user interface employing Distributed Virtual Environment and using appropriate intelligent agent technologies.

- Security issues

  This sub-project considers both the general security issues on the Internet and the specific problems generated by this application.

This challenging project involves 9 professors and more than 20 graduate students from 7 universities in Canada. University of Ottawa and Concordia University are both responsible for the user interface and intelligent agents sub-project. For more information, please refer to the following URL: http://www.citr.ece.mcgill.ca/english/enabling_techn_eng_citr.html.

1.3 Organization of Thesis

The contents of this thesis are organized into seven chapters. Chapter 2 provides an overview of the thesis. In this chapter, we present the scope of the thesis and provide specific motivations and objectives. Chapter 3 presents a review of the literature in the field of automated negotiation. We provide a clear definition and description of the negotiation process, describe in more detail the difficulties involved in automating negotiation, overview the current research approaches to the problem and provide a survey of some of the practical work being done in the field.

Chapter 4 provides a critical analysis of the current approaches to the problem with the intent of finding an appropriate solution. With regards to the solution approach proposed in chapter 4 and the difficulties presented in chapter 3, chapter 5 presents the desiderata and describes how such a solution could be implemented. In chapter 6, we describe an example of such implementation by presenting a software Sales Agent prototype that we have designed and implemented. Finally, we conclude in chapter 7 by providing our contributions and suggestions for future work.

Chapter 2
Thesis Overview

The purpose of this chapter is to provide a context for our research, clarify the scope of this thesis and state our motivation and objectives. First, we start by introducing a framework for electronic commerce (section 2.1) and a framework for negotiation within e-commerce (section 2.2). Then, we present the scope of the thesis (section 2.3) and describe our motivation and objectives (section 2.4).

2.1 A Framework for Electronic Commerce

It is beneficial to study the role and place of negotiation in e-commerce in the context of a common procurement framework. The literature is very rich in the field of marketing and consumer behavior in this domain. Based on this rich knowledge, Guttman et. al have proposed a model called CBB (Consumer Buying Behavior) which "*comprises the actions and decisions involved in buying and using goods and services*" [Guttman 98b]. The CBB model is a descriptive model and consists of the following six fundamental stages:

1. Need identification

   In this stage, the consumer who previously had no intention of buying anything in particular, becomes aware of some unmet need. The consumer can be reminded of this unmet need through product stimulation. The result of the consumer's choice at this stage could be a certain type of product to buy, a fuzzy set of possible similar products or even a specific product to buy.

2. Product Brokering

   In this stage, the consumer determines what to buy. At this point, he has a good idea of the type of product he wants to buy, so this stage comprises the retrieval of information to help in deciding which product to buy in the line of product chosen. The result of this stage is a small fuzzy set, or *consideration set*, of possible products to buy in the line of products chosen.

3. Merchant Brokering

   In this stage, the consumer has his mind set on a limited number of products and has to determine who to buy from. This decision is usually based on price considerations, but could also be influenced by the reputation of the merchant, the geography where the products are available, the delivery options or the extras that comes with buying a product (such as warranty and customer service). The end result of this stage is the decision to buy the chosen product at the chosen store, giving that the terms of the transaction can be negotiated and the chosen store supports adequate payment and delivery options.

4. Negotiation

   In this stage, the consumer determines the term of the transaction with the vendor. Some markets leave no room for negotiation and personalized pricing, as the prices are fixed and non-debatable. In other markets such as the automobile or housing market, the negotiation stage is an integral part of the shopping process.

5. Purchase and Delivery

   In this stage, the consumer provides personal information for delivery and payment of the good. The actual payment process can vary depending on the chosen payment option (e.g. cash, credit card, check).

6. Service and Evaluation

   This post-purchase stage comprises the evaluation of the overall satisfaction of the buying experience, including after sales service.

Guttman et.al. note that the use of agent technology is well suited for stages 2, 3 and 4. In stage 2 and 3, successful commercial agents are already being used to help customers locate, compare and buy products and services [Persona Logic] [Jango] [Firefly] [BargainFinder]. In stage 4, agents can be used to negotiate and act on behalf of their owners. However, they are currently used only in fielded research experiments, such as in classified ads marketplace [Kasbah] and in auctions [AuctionBot]. Section 2.2 provides a framework for stage 4.

2.2 A Framework for Negotiation within E-Commerce

Negotiation can be used in a variety of situations and, even within e-commerce, there are different types of negotiation problems. Furthermore depending on the problem at hand, the negotiation process can vary. So to clarify the scope of this thesis, we divided the conceptual road map of negotiation in the following two dimensions: 1) the business model being used 2) the number of issue negotiated.

2.2.1 Business Models

Based on the level of competition and how committed parties are to negotiate with one party at a time, we classify commerce negotiations in one of the four following business models:

1- Many-to-One (many buyers, one seller)
2- One-to-Many (one buyer, many sellers)
3- Many-to-Many (many buyers, many sellers)
4- One-to-One (one buyer, one seller)

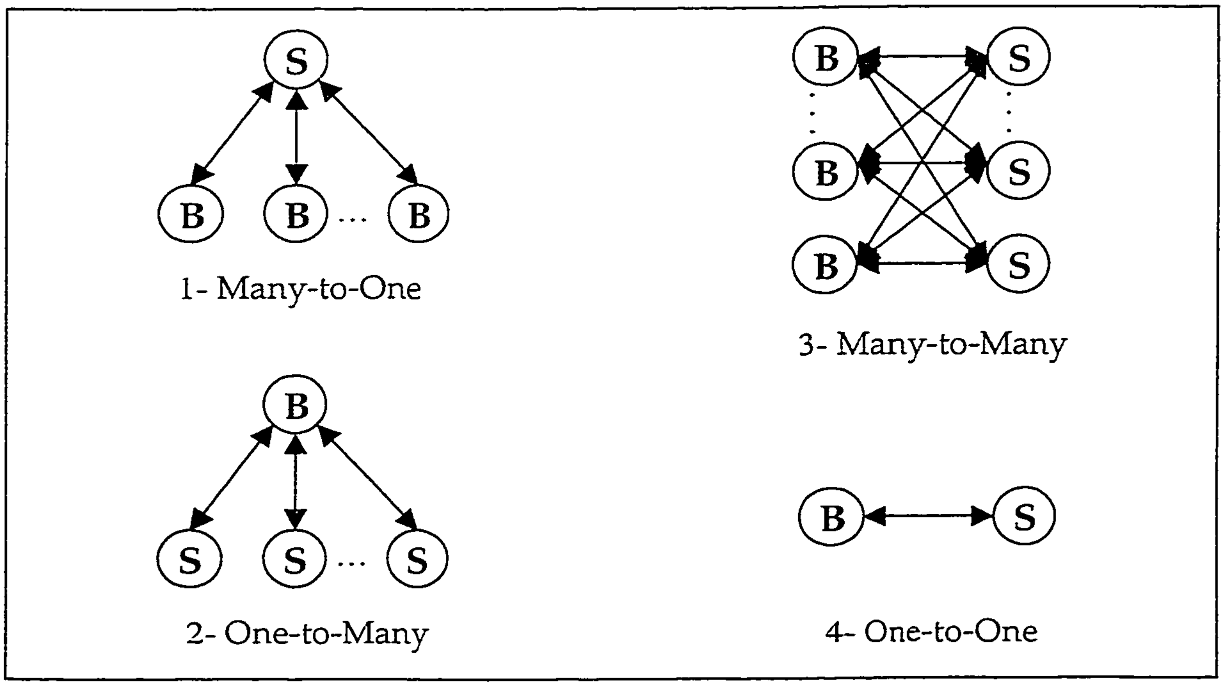

Figure 2.1 Business Models

By looking at figure 2.1, model 1, 2 and 3 could be viewed as higher abstractions of model 4. However, we distinguish model 4 (and similarly model 1 and 2) from model 3 by defining that parties engaged in one-to-one negotiations exchange offers between themselves (and only themselves) till an agreement is reached or negotiation fails, while parties engaged in many-to-many negotiations are not committed to any party in particular and interact with more than on trading partners at a time. By providing such a model, we address the fact that the presence of competing buyers or/and sellers can affect how negotiations are conducted. In model 1, the seller can benefit from the fact that buyers are in competition. Similarly in model 2, it is the sellers that are in competition and the buyer who has the upper hand. In model 3, the situation could be described as a competitive marketplace in which no side has the advantage, as both sides have to cope with competitors. So in model 1, 2 and 3, competition plays a big factor in the layout of how individual transaction will occur. In this thesis, we will refer to negotiations in model 1, 2 and 3 as *market driven* negotiations as opposed to *one-to-one* negotiation for model 4.

2.2.2 Number of Negotiation Issues

Negotiations where a single issue such as price is debated are referred to as single-issue negotiations, or distributive negotiations. The game theory literature describe this situation as a zero-sum game where, as the value along the single dimension shifts in either direction, one side is better off and the other is worse off [Rosenschein 94]. In other words in a game, you typically either win or loose. However, it is not as black and white in price negotiations, whereas it could be possible that both parties benefit from a negotiated agreement, even if the agreement is more beneficial to one party than to the other.

When more than one issue is on the negotiation agenda, the terminology to use is multi-issue negotiation, or integrative negotiation. In business negotiations, price is a major issue, but other issues such as delivery, warranty or extra features might also be negotiated. Multi-issue negotiations allow the possibility to tradeoff among issues, i.e. to compromise on an issue while asking more on another. These negotiations usually focus on finding the tradeoffs that make both parties better off. Jim Oliver notes that "*finding these tradeoffs in a competitive environment is fundamentally challenging*" [Oliver 96].

2.3 Scope of The Thesis

This thesis focuses on single-issue, one-to-one price negotiation in retail business contexts, whereas price negotiation is viewed as an extension to personalized pricing.[1] We consider the single-issue case of price because it is easier than the more general and complex multi-issue negotiations. We view the single-issue negotiation case as a starting point to provide a solution to the extended problem of multi-issue negotiations and leave the issue for future research.

We address the case of one-to-one price negotiation because it is characteristic of individual negotiations and it allows a personalized pricing approach not fitted for competitive market driven negotiations. Furthermore, research on price negotiation has been mostly done in competitive market driven environments. Hence, there is a lack of research in a one-to-one negotiation that we plan to address. Finally, we are lead to believe that addressing the negotiation problem in a one-to-one model is not only useful, but that the one-to-one model is also the most appropriate model for retailing (see section 4.1 for a complete analysis).

While we recognize the fact that two parties bargaining with each other are influenced by the opportunity to negotiate with other trading partners, we won't use this opportunity as a leverage for what we call "power bargaining", i.e. to use the competition offering as bargaining power to make a party lower its price. We assume the competition has been looked upon and each party is fairly committed to reach an agreement. However, if negotiations reach an impasse and a deal cannot be made at a satisfactory price, the competition's offering becomes relevant again.

Furthermore, current research in automated negotiation do not properly address the practical issues in designing viable automated system. Rather, the focus is put on theoretical results, giving solutions to relatively simple experimental problems in fielded laboratory tests and under usually very restrictive assumptions. Moreover, agent research in the field of automated negotiation has been nearly exclusively focused on agent-to-agent negotiation in

[1] The idea is that if one is able to provide a tailored price to each of its customer, one has grounds to use such information as a lowest acceptable price threshold in price negotiation.

closed environment. In particular, the term applies to agent technology, whereas it is possible for the designer of an electronic market to control the actions of the software agents in his market by providing the users with a set of predefined or customizable agents. In such a closed system, a third party agent entering the system would be able to take advantage of the other agents because it would not be subject to the same pre-designed limitations and restrictions. On the other hand, our motivation is to design a viable real world application where the agents would be built by separate self-interested designers. Hence, the rules, or *protocol*, that governs the system must be coded separately from the software agents. We feel such an *open* approach is more desirable for the implementation of practicable systems in the real world because: 1) it is an approach that allows for some of the negotiating parties to be humans, in our case the customers 2) it is an approach that allows for the users to provide their own custom made agents 3) it is an approach more robust to malicious attacks to the system. For such reasons, our research work aims at designing systems in *open* environments.

To conclude, our thesis takes a computational rather than an economical approach to price negotiation. In other words, we are more concerned with the technological feasibility of the idea than with the commercial aspect of it.

2.4 Motivation and Objectives

Currently existing electronic retail online stores do not offer one-to-one price negotiation capabilities. From a consumer's perspective, price negotiation provides an opportunity to debate the price. From a vendor's perspective, the ability to negotiate allows for flexibility in pricing that a rigidly fixed price policy cannot offer. Hence, we feel that one-to-one price negotiation would be beneficial in online stores and e-commerce in general. Moreover, the ability to negotiate the price online could be described as a win-win situation where both the vendor and consumer experience increased satisfaction. Given the high cost of providing a human sales agent online, we are motivated in answering the following question:

> *Is it possible to design a practicable and viable automated system that is able to autonomously negotiate on behalf of a retail vendor in a commercial one-to-one business environment?*

Answering this question is undoubtedly complex given how difficult humans find it to rationalize the negotiation process and given the limited results of practical work in the field. Therefore we do not expect a comprehensive answer to this question. However, we do hope to find out what are the basic requirements and issues involved. In more details, this thesis aims to:

- Study the issues and difficulties involved in automating negotiation.
- Analyze the current state of automated negotiation to gain insight of available technology.
- Provide a critical analysis of why a market driven model of negotiation is not an appropriate model to be used for retailing.
- Show that the competitive automated solutions available are no good when applied in a cooperative setting.
- Propose a negotiation protocol and a methodology for eliciting the negotiation strategy from the vendor.

A software sales agent and multi-agent architecture will be prototyped to:

- Begin to show proof of concept in the feasibility and utility of intelligent agents in negotiating on behalf of retail vendors in a cooperative setting.
- Discover practical implications or limitations while designing such a system.
- Show that our proposed avenue of solution is feasible and useful.
- Satisfy the CITR e-commerce project requirement for intelligent agent support
- Provide a test bed for other multi-agent projects and studies

In designing such a software sales agent, we are motivated by the following objectives:

1. Agreement should be reached without the use of a mediator or central decision system.
2. The agent should be *fully* autonomous, i.e. require no human presence to make a transaction.
3. Contrary to current automated price negotiation solutions, the agent should be able to negotiate over more than one product at a time.
4. The agent should be able to negotiate with different customers under different conditions.
5. The agent should be trustworthy and predictable. We believe a key factor in order for the merchant to trust his agent is the capacity of the agent to motivate the decisions it makes.

Chapter 3
Literature Review

While chapter 1 and 2 provided an introduction and overview of the thesis, this chapter explores the issue related to this thesis in more detail. First, we provide an introduction to the research field of negotiation by providing a definition of the term negotiation, explaining its key characteristics and discussing some of the research models proposed for it (section 3.1). We then address the issue of automating negotiation and discuss the difficulties involved (section 3.2). In section 3.3, we present an overview of the various approaches taken by researchers to solve the problem. Finally in section 3.4, we provide a brief survey of the practical work being done in the field of agent-mediated automated negotiation for e-commerce.

3.1 An Introduction to Negotiation

3.1.1 A Definition of Negotiation

Like the notion of agency, there is no agreed upon definition of the term negotiation. Jim Oliver describes it as a search process by which negotiators jointly explore a multi-dimensional space in order to agree to a single point in space [Oliver 96]. On the other hand, Lewicki defines it as a "*basic social process used to resolve conflicts*" [Lewicki 85]. In this thesis, the term negotiation will refer to the following definition.

Definition: *Negotiation is the process by which self-interested parties jointly participate in order to try to reach a unanimous solution to an issue for which it is in their own interest to try to come to an agreement rather than to break contact.*

Moreover, the nature of the issue is such that no established solutions, traditions, "rational methods," or higher authority available can be used to resolve the conflict [Lewicki 85]. The process may involve the exchange of information, the relaxation of initial goals, mutual concessions, lies or threats [Rosenschein 94]. The solution found as a result of the negotiation process is considered as the core of a binding agreement between the parties.

The negotiation process usually involves defining beforehand, although sometimes implicitly, a set of rules and conventions, or *protocol*, which will govern how the negotiations will be conducted.[1] For example, a negotiation protocol can define what kind of solutions, or *offers*, will not be considered acceptable solutions. The protocol usually defines what will be the *negotiation mechanism*, i.e. the process by which an end-offer will be reached or determined. A fair negotiation mechanism *necessarily* allows for all interested parties to make counter-offers or have their say in resolving the issue, although it is possible for parties to reach an agreement on the first offer.

If parties reach an irreconcilable conflict during the course of negotiation, a *decision mechanism*, or *conflict resolution* mechanism, may be agreed upon by all parties and used to resolve the matter.[2] Decision mechanisms usually make use of a global utility function to determine which offer is the best or winning offer under the circumstance. If it is not possible for all parties to determine or agree upon what makes one offer better than an another, the utility function chosen can be a simple random function such as flipping a coin or rolling a dice.

Joint participation, unanimous solution and binding agreement are key concepts of our definition. Joint participation entails that all parties concerned by the issue and who have interest in participating in the negotiation process have the ability of doing so, either by making offers and counter-offers or by having their words in the decision mechanism used. Unanimous solution implies protocol consensus, which means that all parties must explicitly agree to play by the rules of the negotiation protocol and thence, abide by the negotiation mechanism and decision technique used. Further, a unanimous solution is found only if each party gives explicit consent to the current offer on the table, either directly by accepting the offer, or indirectly by agreeing to abide by a decision mechanism. In addition to agreeing to play by the rules, parties must also agree to oblige by the unanimous solution or offer (if one is found), i.e. the agreement reached is a binding agreement.

---

[1] The protocol itself can be an issue of negotiation.

[2] A *decision* mechanism can sometimes be referred to as a *negotiation* mechanism when the decision mechanism regulates the whole negotiation process.

3.1.2 Modeling the Negotiation Problem

Researchers have proposed different models, classifications and axis of comparison for negotiation. In this section, we present some of them as a research introduction to negotiation. Our goal is to provide the reader with a general understanding of the different dimensions of the problem and to position our work within these models. Unless explicitly stated, the term *agent* will refer to either a human or software entity in the following sub-sections.

3.1.2.1 Negotiation Domains

According to Rosenschein and Zlotkin [Rosenschein 94], negotiation can be categorized in three domains: 1- Task Oriented Domains 2- State Oriented Domains 3- Worth Oriented Domains. They describe Task Oriented Domains as being a subset of State Oriented Domains, which in turn form a subset of Worth Oriented Domains. In each of these domains, agents have goals which they want to attain. Depending on the type of domain, conflict or match with other agents' goals may arise and will affect negotiations.

- Task Oriented Domains (TOD)

  These domains are characterized by the fact that the agents have all the resources to accomplish their goals, in this case a set of tasks. However, each party could be better off if the task could be redistributed among them. Negotiation is viewed as a cooperative coordination process to find mutually beneficial task redistribution. Key issues here are that the tasks are indivisible and that each agent can accomplish its tasks alone.

- State Oriented Domains (SOD)

  These domains are characterized by the fact that the agents may not have all the resources necessary to accomplish their goals, as they might need the resources of other agents. Worse, they could be in competition with other agents' goals or need for resources. Moreover, the world is viewed as a state domain, whereas the goal of an agent is to move the world from an initial state to a goal state with minimum cost. In these conditions, it is possible there may be no state of the world that satisfies the goal of all agents.

- Worth Oriented Domains (WOD)

The Worth Oriented Domains are a generalization of the State Oriented Domain, wherein the world is not viewed as black and white as in SODs. In WODs, the agents associate worth to every state of the world in terms of valuation and costs. In the words of Rosenschein and Zlotkin, "*those states with the highest values of worth might be thought as those that satisfy the goal completely, while others, with lower values, only partially satisfy the goal*" [Rosenschein 94]. In this context, each agent attempts to maximize their gain by reaching the states of the world with maximum worth according to each one of them. The notion of partially satisfying a goal allows for reaching compromises and hence possibly increasing the overall efficiency of coming to an agreement. Because of the inherent concept of price and valuation, we evaluate negotiation in commerce to be better represented as a Worth Oriented Domain.

3.1.2.2 Degree of Cooperation

Researchers have acknowledged the fact that the degree to which agents are willing to cooperate with one and another is an important issue [Rosenschein 94] [CMIT-1016]. Cooperation can be defined in terms of sharing personal information, compromising individual goals or accomplishing extra tasks in the name of global benefit. Basically, cooperation entails that all the agents have the same goal. In the words of Beam and Segev, "*there are two types of problems, the cooperative and the non-cooperative, which represents two extremes on a continuum of possibilities*" [CMIT-1016].

At one extreme of the continuum when cooperation tends to be very high, we see negotiation as being more of a *coordination* process than a *conflict resolution* process. As an example, a process by which parents debate with their children the about of the next family vacation would be described as a cooperative problem type of interactions. No individual has incentive to lie about his preferred destination and it is most likely that one will compromise in face of majority or if another family member desperately wants to go at a specific location. At the other extreme when cooperation tends to be nil, the nature of the conflict determines the negotiation process. If parties are not even willing to cooperate enough to negotiate with one and another, if the agents' goals are totally irreconcilable or if no one is willing to compromise from its position, no agreement will be reached and other scenarios such as going against the other's will by force may be considered. In the case where the agents' goals are partly reconcilable (such as in WOD), the distance between the least acceptable agreement and the agreed upon deal is considered as a surplus [CMIT-1016].

The level of cooperation is particularly relevant when building software agents. A software agent that assumes falsely that the other agent will cooperate puts itself at the mercy of the other. For example, consider a process where a manager agent has to delegate a certain task to an assumed cooperative contractor agent. If the payment is based on cost figures provided by the contractor, the manager agent exposes itself to pay whatever the contractor agent says the cost are, whether they reflect the real cost of carrying out the task or not. Because each party wants to maximize its own share of the deal and has no incentive to reveal personal valuation to the other party, we view price negotiation as essentially a non-cooperative problem.

3.1.2.3 Interaction Types

Rosenschein and Zlotkin [Rosenschein 94] have studied the various kind of interactions that two agents can encounter when trying to achieve their goals. The authors define four possible interactions from the point of view of an individual agent:

1- Symmetric cooperative situation

In this situation, the presence of the other agent is desirable or even necessary to both agents to accomplish their goal. Here, each agent welcomes the presence of the other agent as there exists a deal in the set of negotiated solutions which both agents prefer over achieving their goal alone. An example of such situation is the child car-pooling example [Rosenschein 94]. In this scenario, two neighbors with respectively three and four children, some of them attending the same school, try to find a joint agreement to take all their children to school. Obviously, each neighbor gains by reaching an agreement, as each of them can do no worse than taking his own children to school all the time.

2- Symmetric compromise situation

In this situation, both agents would prefer to be alone in the world but are forced to cope with the presence of the other. Here, the presence of the other agent is not welcomed, as each agent would be better off achieving his or her goal alone. A simple example of such situation could be the scenario where a lottery winner founds out he/she has to split the lot with another unknown winner.

3- Non-symmetric cooperative/compromise situation

In this situation, one of the two agents would prefer to be alone in the world while the second welcomes the presence of the other. An example of such situation is the scenario wherein a child is forced by his parents to share his new computer with his younger brother or sister.

4- Conflict situation

In this situation, no state of the world can satisfy all the parties. Either no deal will be made or one of the agents will not achieve his goal. Just consider the scenario wherein one of two roommates wants to paint the living room in dark blue while the other wants it in light beige.

The authors note that in SOD, all four types of interaction can arise, while only the symmetric cooperative situation can ever exist for TOD. We further add that for WOD, the conflict situation is less likely to arise because the notion of partially achieving goals is more flexible than the binary measure of success in SODs. To conclude, we evaluate negotiation between a seller and a buyer to be essentially a symmetric cooperative situation because the presence of the other is necessary and welcomed.

3.2 Automating Negotiation: Difficulties

We note that automating negotiation can be done at two different levels: 1) at the participant level, i.e. taking as input the negotiators' goals and strategies and providing automated means to reach theses goals using the prescribed strategies; 2) at the process level, i.e. defining how negotiations will be conducted and providing the infrastructure. At the participant level, the use of intelligent software agents, combined with knowledge elicitation programs and means to create personal negotiation strategies, is viewed as a promising avenue. At the process level, protocols, negotiation and decision mechanisms, communication interfaces and monitoring systems can be design and implemented. As the software agents could use such process infrastructure to negotiate, the reader should note that issues at the two levels are not disjoint.

Apart from the inherent difficulties involved in the negotiation process itself, automating negotiation involves the following additional three problems as noted by Beam and Segev in [CMIT-1016] and [CMIT-1022]:

1- The need for an ontology
2- The formulation of the negotiation
3- The exploitation of the negotiation strategy

To these problems, we add a fourth issue of our own, the *termination of negotiation* problem. In brief, it addresses the issue of when and how to terminate the iterative process of negotiation. We discuss of these four problems in more details in the following sections.

3.2.1 The Need for an Ontology

An ontology is a formal semantic representation and specification of the objects in a domain. In other words, it's a standardized way of naming and classifying things in order to remove ambiguity when referring to something. In order to make sure that each side "talks" about the same thing when negotiating, we need such a representation for the goods and services which are to be traded. Because computers are by default semantically unaware, a software agent looking to buy a "car" would not even engage in negotiation with an agent selling an "automobile" and an agent searching for a "gray" car would not consider a car which is "light gray". This kind of naming and synonyms problem have been called the "you say tomato, I say tomahto problem" [Bhargava 91]. For a more concrete example of the naming problem, here is a search we've done for a Canon digital camera using Excite's product finder agent [Jango]. It returned the following six different model names for the *same* product:

(1) Powershot A5
(2) Powershot A5 Digital Camera
(3) Powershot A5 Digital Color Camera
(4) Powershot A5 Digital Camera 1024x768 pixels Color LCD.
(5) Canon Powershot A5
(6) Canon Powershot A5 Digital Camera

In addition, the ontology must provide a sufficient level of details to describe the goods or services in their entirety. For some products such as music CD's, only a small number of attributes (such as the artist's and album's name) might be needed to describe the product completely. Other products such as computers or cars do not easily lend themselves to such simple specifications. As noted by Beam and Segev [CMIT-1016], it is crucial that the ontology captures all important attributes and features of an object, e.g. color, size, options etc. We see two reasons for this requirement: (1) to distinguish one product from another so as to compare apples with apples, not apples with oranges; (2) one might rely on these attributes and features to determine the worth of the product.

Much like the Universal Product Code (UPC) system of bar codes and the fruit codes[3] that certain merchants use, a comprehensive ontology for e-commerce is needed. Although it is not the focus of our work to provide such an ontology, we do plan to address the ontology problem. See the work on KIF [Genesereth 95] and Ontolingua [Gruber 93] at the Stanford University for examples of research being done in this domain.

[3] The fruit code is the small sticker with a number that you may have found on the fruit upon purchase.

3.2.2 The Formulation of the Negotiation

In order to be able to bargain electronically, an organization needs to *explicitly state* what it whishes to achieve from the negotiation in terms of *goals* and *strategies*. As discussed in [CMIT-1016], this is especially difficult since human negotiators often don't have a clear and well-defined idea of their own goals or preferences when negotiating and cannot articulate in advance what the desired strategy or response to a given situation would be. Instead when negotiating, humans often have the implicit strategy of extracting as much as possible out of the other parties, agreeing to a reasonable offer only when they feel that no further gain can be made. This intuitive feeling of knowing when the limit was reached is often based on hints or signals that the other negotiators are giving out. Beam and Segev [CMIT-1016] note,

> "*Moving negotiation to electronic media deprives the negotiation process of many small hints and signals human negotiators give out; rather than hints, these signals must be explicitly codified. What was acceptable when only hinted at may be completely unacceptable when brashly, explicitly stated.*"

In addition, it is often unclear to oneself of the conditions under which one would consider an offer reasonable, i.e. what is one's goal in terms of preferences. Sometimes, a reasonable offer is simply the best offer the other party can give, whereas an agreement is better than no agreement at all. However, when having to define a reasonable offer in terms of personal preferences, hard facts and valuation functions, human negotiators often find the task difficult. For example, how would one rate the desirability of a car in terms of its color? What is the list of attributes that make a car more (or less) desirable? In work contract negotiation, is an offer of 2 weeks vacation and 5% increase in salary better than an offer of 3 weeks vacation and 3% increase in salary? As one can see from these simple questions, it is not trivial to elicit preferences and the level of difficulty rapidly increases with the state space of the negotiation problem. In our opinion, the problem of formulation is the biggest barrier to automated one-to-one price negotiation. The following two sub-sections address the problems involved in computerizing the negotiators' goals and strategies in more details.

3.2.2.1 Goals

When mathematically representing ones preferences, one can state individual preferences with an ordinal or cardinal representation measure. Consider the following example where one wants to buy a car, whereas the preference set for the color of the car is blue, red, green and black. The following is an ordinal representation of the preferences: I prefer blue over red and black; don't care between red and black; prefer red over green. A cardinal representation of the same example would measure the preferences in terms of worth, e.g. green is worth 100$, red and black is worth 200S, blue is worth 500$ and perhaps yellow is worth -2000S.[4] In negotiation where a price is involved, a cardinal representation is often more appropriate than an ordinal representation because there is already a notion of worth involved and because ones preferences directly affects ones willingness to pay. Generally, the cardinal representation is a more useful and complete representation, but it is more difficult to elicit the knowledge needed for this representation.

When there is more than one relevant attribute, not only is there the problem of explicitly eliciting and representing each attribute, but there is the additional problem of combining the different attributes as a whole. With an ordinal representation, the problem gets complex rapidly because it is proportional to the number of attribute and attributes values, as each point in the space of deals must be rated against each other. For example, if we add the attribute radio with value "CD" or "cassette" in our car example, we have to explicitly state which one we prefer more, a red car with cassette or a green car with CD. With the cardinal representation, we can simply add the worth of the individual attributes, given that the attributes are independent. For example, if one values the CD at 200$ and the cassette at 100$, the value of "red and CD" would be 400S. If the attributes are not independent as in the case where two attributes combined is worth more (or less) than the individual values, extra steps must be taken to elicit such information from the user.

[4] Although convenient, the worth doesn't have to be measured in dollars. Furthermore, worth can also be measured relatively to a fixed point, e.g. the maximum estimated worth of the car. In this case, a buyer with the above preference would consider buying a given car at, say 15 000$ if it's blue, 14 700$ if it's black or red and 13 000$ if it's yellow.

In addition to the above difficulties, there is the risk that when the user simply states preferences, there might end up with logical contradictions or missing cases and exceptions in the preferences set. An example of contradiction would be one preferring blue over red, red over green, yet green over blue. A missing exception, such as forgetting to include gray in the preferred color set, can be a cause of problem given the exception situation arises, as the consequences are unknown and potentially undesirable.

3.2.2.2 Strategies

In the words of Robinson and Volkov, "*a negotiation strategy refers to the plan by which an agent intends to interact with other agents, while using a particular negotiation protocol, in an effort to achieve desired outcome*" [Robinson 98]. Note that the negotiation protocol often influences the choice of a negotiation strategy. Further, the actual interactions between participants can vary depending on the negotiation protocol being used. Interactions may consist in making one or more offers, or even no offer at all. In the latter case, the participants' interactions are replaced by a decision mechanism, such as flipping a coin. However in the more general case, negotiation is based on an exchange of offers and counter-offers and we need to explicitly encode the strategy.

To encode a strategy, we need several decision functions: 1) to decide what should be the first offer 2) to determine the making of counter-offers 3) to decide if we accept or refuse an offer being made to us. In making these choices, people generally consider their goals and preferences, the negotiation mechanism at hand, general domain knowledge, but also beliefs about the other negotiator estimated lowest acceptable deal. Typical questions one might answer include: Should we make our first offer independently from the other negotiator, say at 20%, 25% or 35% over our reservation price? [5] Or should our first offer be based on the estimation of the other agent's reservation price, if so at 10%, 17% or 30% bellow the estimated price? How do we calculate this estimated price? If we receive an offer of X$ in round Y, what should we counter-offer with? Should our counter-offer be based on the previous history of counter-offers or follow an arbitrary functions? Should we accept the first satisfactory offer or refuse in hope for a better offer to come?

[5] A reservation price is the maximum price a buyer is willing to pay or, in the case of a seller, the minimum price he or she will sell for.

In short, these questions reveal the complexity of the activity. And as mentioned before, people are often guided by intuition, pride and human signals when faced to answer these questions. Unfortunately, when having to computerize a negotiation strategy, we are deprived of such inherently humans characteristics.

3.2.3 The Exploitation of Strategy

This section addresses the consideration that, when replacing a negotiation strategy with a computer algorithm, there is a risk that the algorithm can be exploited and even inferred by other parties. The risk of being exploited is even higher in open systems because parties are of unknown intention and can't be assumed to be trustworthy. Indeed as we discussed in section 3.1.2.2, one can be at a great disadvantage if he or she doesn't assume that the other side will try to get the better of him. For example, an agent that says to someone "Your price, is my price" greatly exposes himself at the mercy of a vicious party if he doesn't specify at least a reservation price. Anyone could offer 1$ for the good or service in question and walk away with it because nothing prevents such an offer. In the real world, this consideration is less of an issue because negotiation is an interactive process and transactions don't occur immediately and automatically after an offer is made, contrary to the electronic transactions. The sales person could always say "1$ is not serious" and refuse to sell or ask the client to make a more reasonable offer.

Another potential harmful situation is when a party tries to infer your strategy in order to gain significant advantage over you. For example, if someone knows or guesses that the strategy of a software sales agent is to accept any offer above a certain threshold, he might be tempted to start an offer at 1$ and progressively increase it by 1$ until he reaches the sales agent's threshold. An extension of this example occurs in multi rounds negotiations with a software agent, where one assumes the agent will make counter-offers till it reaches its bottom price. In this case, one could make ridiculous counter-offers and let the software agent spiral down to its bottom price. In both these cases, the worst possible deal for the software sales agent is made.

Again in the real world, these considerations are less of an issue because in these cases, the vendor will probably stop the negotiations and refuse to engage in further negotiations with this customer. Automated negotiation systems do not have a priori the same human capacities to adapt. In these settings, the design of the negotiation protocol is the critical aspect of the overall system, as all these situations must be accounted for [Rosenschein 94]. McMillan [McMillan 94] describes true anecdotes of poorly designed auctions and the real life consequences of such designs. Finally, we note that even complex algorithms are not totally safe from inference because the other party might be a software agent with all the necessary time and computer power to crack the algorithm.

3.2.4 The Termination of Negotiation

Another fundamental problem that arises in multi-round negotiation between two software agents is how to determine the end of negotiations. Consider figure 3.1 that models the price negotiation problem:

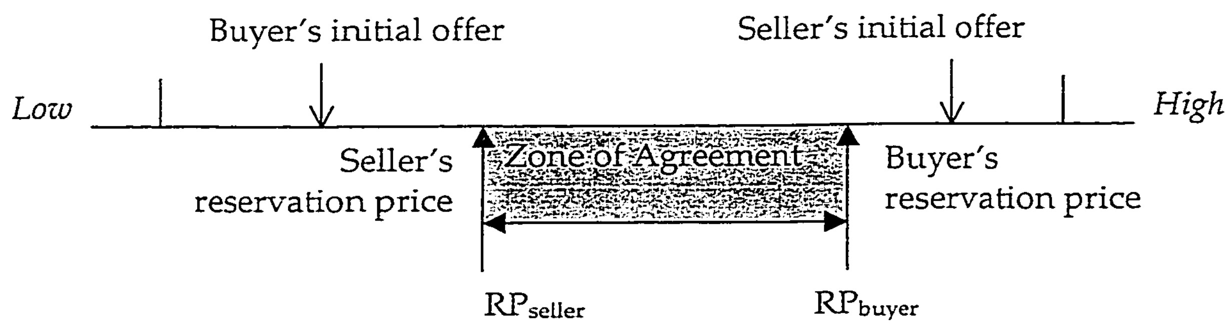

Figure 3.1 Zone of Agreement: $RP_{buyer} > RP_{seller}$

Intuitively, one could say that the end of negotiation occurs when a) one agent makes an offer in the zone of agreement or b) both agents have reached their reservation price and found that $RP_{buyer} < RP_{seller}$, i.e. found that there is no zone of agreement. Although this affirmation is correct, it poses some practicable computational problems under the different strategies that the agents may adopt.

First of all, if both agents stand to their offer and wait for the other agent to make the next counter-offer, we enter a deadlock and the negotiation will not end. Then, there is the problem of determining if there exists a zone of agreement at all. Remember, neither one of them knows the other's reservation price. In these conditions, how does one know that the other has reached his bottom price? Common sense suggests that if the other agent rejected our previous offer and is not making any more counter-offers, it has reached is bottom price. However as we mentioned above, the other agent might just be waiting for an offer as part of his negotiation strategy.

Another approach would be to bring the notion of *final* offers, whereas the agent would declare that it has reached his reservation price by saying: "this is my final offer". In this case, the acceptance or refusal of the other agent would end the negotiation. The problem with this approach is as follows. If one agent knows the other agent will eventually make a final offer, he has everything to gain by standing firm in his position (or compromising very little) and wait for the other to make that final offer. In this case, the worse possible deal for the other agent is made. If both agents adopt this approach, then each one of them has incentive to wait and chances are they will enter into a deadlock situation.

3.3 Research Approaches to Automated Negotiation

In this section, we review three areas of research that have approached the problem of automating negotiation, namely the fields of *Negotiation Support Systems* (NSS), *intelligent agents* and *economic mechanism design*. Our intention is to provide research context to our work by presenting a basic coverage of how researchers have tackled the problem.

3.3.1 Negotiation Support Systems

Researchers in the field of Decision Support Systems (DSS) attempt to build computer systems to help and support humans in making better decisions. Within the field of Decision Support Systems, a special class of DSS emerged with the emphasis to support group decision, namely Group Decision Support Systems (GDSS). Furthermore, Negotiation Support Systems (NSS) are a class of GDSS specially designed to provide assistance in reaching negotiated agreements. Rangaswamy and Shell [Rangaswamy 97] classify NSS into two categories: 1) Preparation and evaluation systems; 2) Process support systems. According to the authors, preparation and evaluation systems help individuals to organize information, develop preference representations, refine pre-negotiation strategies and evaluate offers. The process support systems operate in lieu of a bargaining table and are designed to help negotiators move towards more integrative settlements.

In other words, a NSS aims at providing computer assistance to human negotiators in all the different aspects of negotiation, such as putting in place the initial set up of the problem, facilitating individual preparation of each party, use of algorithmic power to explore multi-state space, generating options for mutual gain, structuring decision making and communication etc. The computer tools for supporting such activities are varied and include multi-attribute functions, distance measures (for offers), decision trees, risk analysis and forecasting methods. Jelassi and Foroughi [Jelassi 89] have acknowledged the need to design NSS that address behavioral characteristics and cognitive perspective of negotiators. In particular, they believe it is important to use objective criteria to separate people from the problem and avoid culture, language and pride barriers, as well as to make objective decisions.

In particular, we are interested in models and theories used by NSS that deal with acquisition and modeling of individual preferences, as they are pertinent to solve the formulation of negotiation problem. Such theories include Multi Criteria Decision Making (MCDM) and Multi Attribute Utility Theory (MAUT). Essentially, these two theories are based on the presence of individual decision makers with their own goals and criteria separate from the opposing participant. In the words of Jelassi and Foroughi, "*MCDM methods have been used for preference elicitation and aggregation, alternative generation and solution ranking*" [Jelassi 89]. Such methods include weighting methods, sequential elimination methods, mathematical programming methods and spatial proximity methods [MacCrimmon 73]. As for MAUT, it relies on the notion of *utility functions*, where utility can be defined as the difference between the worth of achieving a goal and the price paid in achieving it. More information on these theories can be found in [Zeleny 82], [Hwang 87] and [Keeney 76].

Work in the field of NSS includes: visualization of the negotiators history of moves in the negotiation space by FACILITATOR [Chaudhury 91]; supporting knowledge elicitation of preferences by PREFCALC [Lauer 87] and NA [Rangaswamy 97]; providing mediator and arbitrator facilities by MEDIATOR [Jarke 87] and CAP [Fraser 81]. As opposed to the non-cooperative problem we are addressing, it is to be noted that work in GDSS and NSS has almost exclusively focused on cooperative problems solving, a point recognized by [Bui 89] and [Rangaswamy 97]. See [Jelassi 89] for a review of existing NSS and design issues.

3.3.2 DAI and Intelligent Agents

The field of Distributed Artificial Intelligence (DAI) has approached the negotiation problem from a Multi-Agent Systems perspective. In brief, Multi-Agent Systems (MAS) are distributed software systems in which individual modules possess characteristics of agency, such as autonomy, mental state and individual agendas. Typically, MAS have been used to manage inherently distributed problems with interdependent activities, such task scheduling and resource allocation. Based on the degree of cooperation exhibited by the individual agents, researchers [Bond 88] [Rosenschein 94] usually distinguish between two types of Multi-Agent Systems:

- Cooperative Multi-Agent System (CMAS)
  Historically, DAI was concerned with ways of getting a society of multiple automated cooperative (or *benevolent*) agents to interact appropriately for the greater good of the system. This field of research became known as Distributed Problem Solving (DPS). Motivations for DPS included providing solutions to large and distributed cooperative problems, such as air traffic control and network management.

- Self-Interested Multi-Agent System (SMAS)
  A separate field of DAI began to emerge when researchers started to consider the implications of designing the agents as self-motivated individual entities. In contrast to previous DSP work, utility became an individual issue in SMAS, i.e. was no longer necessarily defined in terms of "greater good". In SMAS, agents are assumed to cooperate only when it is in their best interest to do so [Genesereth 86].

A large body of DAI researchers has and is still doing significant work under the heading "negotiation". The reasons for this include the fact that there are probably as many definitions of negotiation as there are researchers in this field. Generally from a DAI perspective, negotiation is closely linked to the term coordination, as negotiation is viewed as a communication process used to reach coordination. Thence, a large body of work that aims at achieving coordination in societies of agents is also known under the heading of negotiation. Still, researchers such as Nwana and Jennings [Nwana 96] argue that the distinction between coordination and negotiation is quite fuzzy. For our part and in accordance to section 3.1.2.2, we view this fuzziness as the degree of cooperation of the problem addressed by the Multi-Agent System (MAS). Moreover, we view the interaction process in CMAS as essentially *coordination*, as opposed to *negotiation* in SMAS.

In the following sub-sections, we present a variety of DAI research foci in the area of negotiation/coordination. Like NSS, it is to be noted that DAI work has mostly focused on problem solving of *cooperative* problems. In addition since it is relevant to our work, we also present a sub-section on *communication*, another important area of research in MAS.

3.3.2.1 Planning

Several DAI researchers have viewed the problem of achieving coherent behavior in a society of software agents as a *planning* problem, wherein planning means determining beforehand a multi-agent plan that details all the future actions of the agents. The purpose of such a plan is to avoid inconsistent, conflicting or inefficient actions and interactions between multiple agents. There are two types of multi-agent planning architecture, namely centralized and distributed. In a centralized architecture, agents form their individual plans and forward them to a central coordinator, who in turn analyses them, finds potential inconsistencies and conflicts, removes them and synchronizes the agents activities [Georgeff 83]. In a distributed architecture, the idea is to provide each agent with a model of a multi-agent plan, wherein the individual plans contain personal actions of the agents and the believed actions of other agents [Corkill 79]. The agents proceed to exchange individual plans and update their beliefs accordingly until they all converge to the same global plan.

As a critique, multi-agent planning requires that agents share and process substantial amounts of information. Furthermore, the centralized architecture does no profit from the distributed nature of the system and the distributed architecture assumes that each agent will eventually have a global view of the system, which may not always be possible.

3.3.2.2 Contracting

A now renowned and widely used technique for task and resource allocation is the Contract Net protocol [Smith 80]. In brief, the Contract Net protocol is based on a decentralized market architecture in which organizational structure, task decomposition and contracting are used for dynamic task and resource allocation. In the proposed MAS architecture, agents can either take the role of a *manager* or of a *contractor*. In summary, an agent playing the role of a manager will send a task announcement message to other agents, which in turn will bid on the task according to their capacities to fulfil the task. The manager will award the task to the agent with the winning bid, which will become a contractor for the task. Furthermore, agents don't have a priori specific roles and can change their role during execution.

As noted by Smith [Smith 80], the Contract Net protocol is best suited when tasks lend themselves easily to decomposition into a set of relatively independent tasks. Originally, the architecture was designed for benevolent agents with non-conflicting goals, but several researchers have extended the Contract Net protocol to competitive agents and agents with conflicting goals [Sandholm 93] [Conry 88].

3.3.2.3 Mechanism Design based on Game Theory

This line of research can be traced to Rosenschein's doctoral thesis (which is synthesized in his book co-authored with Zlotkin [Rosenschein 94]) and attempts to design negotiation mechanism between two fully rational and self-motivated agents in an open system. As opposed to benevolent agents in which cooperation has been built into, self-motivated agents must be brought to behave appropriately in a society. Key concepts in this game theoretical approach to negotiation include the following: utility functions; a space of deals; negotiation strategies and protocols.[6] More specifically, a game theory approach to negotiation aims at designing mechanisms that ideally possess the following characteristics [Rosenschein 94]:

1. Efficiency: An efficient mechanism guaranties that the agents will reached an agreement given that an agreement can be reached, i.e. given that a zone of agreement exists. Moreover, efficiency can be measured in terms of Pareto Optimality (no agent could derive more from a different agreement without the other agent deriving less from that alternate agreement) and joint utility (no better deal exists for both agents).
2. Stability: Agents should have no incentive to lie nor should they benefit from knowing the other agent strategy or the decision mechanism used. This attribute is highly desirable and aims at addressing both the exploitation of strategy and termination problem because the strategy can be made public.
3. Distribution: For trust and performance reasons, the system should not require a central decision maker. Moreover, such a system would not be stable, as the agents would have incentive to lie to the central decision maker in order to bias the system and receive decisions in their favor.
4. Symmetry: The mechanism should not treat an agent differently from others because of inappropriate criteria. In other words, the mechanism should be fair to all agents.

In the book, several protocols, strategies and mathematical proofs are provided. However, Nwana, Lee and Jennings argue that they apply for very specific problems and might not suffice for real-life applications [Nwana 96].

3.3.2.4 Learning

Learning implies some sort of skill adaptation to a new environment or to new knowledge. In a context where self-interested agents engage in multiple rounds of negotiations, this field of research addresses the opportunity to make use of the new information that each round of offers reveals. For example, each offer in a one-to-one negotiation partly reveals some private valuation. So instead of explicitly and statically coding the strategies, the field of machine learning allows for dynamic strategies through

---

[6] We have already introduced some of these relevant concepts in section 3.2.

learning algorithms such as Bayesian Probability and Genetic Algorithms. Generally, the intention is to reach Pareto Optimal deals, limit the number of rounds, increase the number of agreement reached and achieve better individual and joint utility.

In [Zeng 96], Zeng and Sycara propose a sequential decision making model, called Bazaar, which is able to learn through a Bayesian probability belief update process. In a price negotiation example, they show how the buyer's belief about the seller's Reservation Price (RP) (and vice versa) can be updated through a set of probability vector during negotiation. In [Zeng 97], the authors present experimental results where learning agents in Bazaar do better in terms of joint utility and reach agreements in fewer number of "offer-exchange" than non learning agents. It should be noted that these results are not broad in scope because they were derived from an experiment conducted under very restrictive conditions. First of all, they limited the scope of the problem by ensuring that a zone of agreement always exists, and by limiting the range of possible value for RP to 100. Secondly, the learning agents have similar initial belief about each other and so naturally converge to this belief, ensuring higher joint utility. Finally, the non-learning agents take longer to reach agreements simply because the values proposed as counter-offers follow a relatively low increase of 1.5% over the previous offer.

Oliver [Oliver 96] presented a thorough study of diverse experimentations in competitive electronic negotiation using a Genetic Algorithm as the underlined learning algorithm. In short, Genetic Algorithm (GA) is a technique inspired by Darwin's theory of evolution and the concepts of variation and natural selection. In the context of negotiation, each agent begins with a pool, or *population*, of randomly generated negotiation strategies, in this case simple threshold strategies. The strategies are then tested in rounds of bargaining under a predetermined multi issue negotiation game with specific rules and payoffs. The best strategies based on individual utility are then preferentially chosen to be *parents* and crossed over to create new candidate solutions (strategies) that comprise the next *generation*. Mutation may be randomly introduced in the cross over process of creating a *child*. The major disadvantage of GA is that it requires many trials (400 trials in this case) to achieve fairly good results.

Oliver's conclusion is that artificial adaptive agents can learn to negotiate and achieve performance similar to humans under the direction of a basic GA. However, he also concludes that adaptive agents are exploitable in terms of strategies, whereas a "tough" agent would do better than a "soft" agent. This conclusion is not surprising, as the field of machine learning specifically addresses the opportunity to exploit the revelation of private information, hence benefiting from the exploitation of the other agent's strategy. Other work in machine learning for e-commerce includes learning in auctions [Preist 98] and competition based pricing [Tesauro 98].

3.3.2.5 Communication

In a MAS, interaction between a society of agents is desired and inevitable. Thence, *communication* between such agents is necessary. Agents may need to exchange all sorts of information to accomplish their goals. Such information can be general knowledge about the system, payoff matrix, partial results, requests, commands, goals, plans etc. Information may be addressed to one agent in particular, to a group of agents or to all agents. Within the large body of work in DAI, two different information exchange architectures stand out: the *blackboard architecture* and the *message passing architecture*. In a blackboard architecture, agents use a shared space, represented metaphorically by a blackboard, to read, write and possibly erase pertinent information. If the message is not intended to all, such information needs to contain specifics of to whom the message is addressed. By contrast in a message passing architecture, messages are physically sent to the appropriate recipient(s). Depending on the implementation of the architecture, network facilities such as registries and routers may be needed to send or broadcast messages.

As for the content of the information being exchanged, researchers have acknowledged the importance to have an Agent Communication Language (ACL). In order for the information to carry some meaning, most ACLs use a set of "performatives" derived from the Speech Act Theory [Searle 69]. Briefly, performatives are the speech-act components of the language used to convey what one can do with the content of the message. Examples of performatives include "tell", "ask", "assert", "perform" and "deny". One of the two emerging standards for an ACL is KQML [KQML], the other being ARCOL [Sad 96].

KQML stands for Knowledge Query and Manipulation Language and is contributed by the DARPA Knowledge Sharing Effort (KSE). KQML messages are formed of performatives and performatives parameters. Most performatives include parameters such as "sender", "receiver", "content" and "language". KQML supports various assertive and directive performatives, and even includes network performatives such as "register", "broadcast" and "forward". However, it is to be noted that because of its lack of precise semantics, KQML has not raised to its expectations. This is one of the reasons why FIPA (Foundation for Intelligent Physical Agents) is now tending more towards ARCOL as a standard. ARCOL (a France Telecom product) has less performatives than KQML and includes semantics at the message level. Still, both languages present a lack of semantics at the communication level, which is a problem for real world open applications.

3.3.3 Economic Mechanism Design

In the field of automated price negotiation for e-commerce, a strong focus has been put on the economic mechanism known as the *auction*. McAfee and McMillan define the auction as "*a market institution with an explicit set of rules determining resource allocation and prices on the basis of bids from the market participants*" [McAfee 87]. The auction is not a new mechanism and has been around for thousands of years. Generally, designers of auction mechanisms are motivated to reach efficiency and stability criteria[7], more specifically to ensure trade efficiency and to allocate resources to parties that value them the most. Since auction theory is a broad and complex economic subject, only a short review will be presented here. See [Milgrom 89] and [McAfee 87] for a more complete treatment.

Depending on the rules of how to bid and the definition of how trades occur, there are different types of auctions. To keep up with standard terminology, auctions come in *one-sided* and *double-sided* format, depending on whether "ask" bids from sellers are permitted or not. *One-sided* auctions are used when many parties want to buy (sell) the same product or service and wherein the context allows for only one party to do so. *Double-sided* auctions, also commonly known as *double auctions* (DA), allow multiple buyers and sellers to interact at the same time and are used to match buying offers with selling offers. Generally, *one-sided*

[7] Efficiency and stability as defined in [Rosenschein 94] and presented in 3.3.2.3

auctions are useful to ensure that a party who more greatly values the good or service will be favored, while *double-sided* auctions are useful to ensure trade efficiency (all possible trades will take place). Depending on the public or private nature of the bids, the auctions are known as *public bid* or *sealed bid* auctions. As opposed to *public bid* auctions, *sealed bid* auctions don't involve iterations. For example in a *one-sided sealed bid* auction, all interested buyers secretly bid only once on the good. When the bids are later simultaneously revealed, the highest bidder is declared the winner. Table 3.1 presents the most common type of auctions.

Auction name	Brief description
English	The stereotyped traditional auction where bids are announced publicly and where bidders continuously have to bid higher than the last bid to have the current winning bid
First Price Sealed Bid	A sealed bid auction where the winner pays the price of his bid.
Second Price Sealed Bid	A sealed bid auction where the winner pays the second highest bid. Also known as the *Vickrey* auction.
Dutch	The auction is the opposite of the *English* auction: the seller starts at a high price and progressively lowers its price. The first bidder to accept the seller's descending price is declared the winner.
Continuous Double Auction	A public bid auction where sellers' offers and buyers' bids are made in real-time and wherein a trade is made when a bid and an offer match. This type of auction is used in stock markets such as NASDAQ (with small variations).
Sealed Double Auction	A private bid auction that relies on the presence of a central auctioneer to calculate the market price. When the market is cleared, all trades take place at the market price. Also known as the *Call Auction*.

Table 3.1 Common Auction Types

There are thousands of online auctions on the Internet today and they have become increasingly used and popular among the customers. Guttman and Maes observe that "*reasons for their popularity include their novelty and entertainment value in negotiating the price of every day goods, as well as the potential of getting a great deal on a wanted product*" [Guttman 98a]. In fact, the auction is probably the most used negotiation protocol in e-commerce today. Auctions are well suited for electronic negotiation because they have a number of characteristics that address the electronic negotiation problems discussed previously. These characteristics are:

- The ontology problem is somewhat resolved because the problem is left in the hands of the human buyers. The item for sale is usually displayed and the buyers may inspect and gather its specifications.
- The formulation of the negotiation problem is made simple. Auctions restrict the negotiation space to the single dimension of price and the rules of the protocol are simple and well understood. In addition, the seller's goal is clear: sell to the highest bidder. The buyer's strategy is also clear: bid higher than the last bid given that the buyer's limit is not reached.
- The exploitation of strategy is resolved from the seller's standpoint because his strategy (sell to the highest bidder) is made public to the buyer with no disadvantage. This is because the auction is a stable mechanism.[8]
- The termination problem is addressed by time constraints.

The auction also has "*the additional advantage of being an institution where the conduct can be delegated to an unsupervised agent.*" [Milgrom 89] Indeed, the protocol is clear and well understood and allows buyers and sellers to come with a strategy beforehand and tell their agents how to behave in the auction. For example, the bidder can tell its agent the absolute maximum he is willing to pay for in an English auction. Auction contests have even been held where participants would provide a computer program to buy or sell in the auction [Rust 93].

[8] Given that certain conditions are met, such as the presence of more than one bidder.

As noted in [CMIT-1019], it is easier to participate in an online auction than in a physical one due to geography and time issues. In the real world, participants must gather simultaneously in the same room and items are auctioned infrequently to assure a critical mass of bidders. The Internet eliminates the geography issue and allows a wider range of bidders to participate, thus reaching a critical mass is relatively faster. As a result, delays between auctions can be reduced.

3.4 Survey of Work in Agent-mediated Negotiation for E-Commerce

In this section, we present three agent-mediated automated negotiation systems for electronic commerce: Kasbah, Tete-a-Tete and AuctionBot. In contrast to the research work presented in section 3.3, we provide the reader with a survey of practical applications that are being used.

3.4.1 Kasbah

Chavez and Maes [Chavez 96] [Kasbah] from MIT Media Lab have created Kasbah, a multi-agent online marketplace for the selling and buying of goods. The idea behind the system was to reinvent the classified ads. In the Kasbah marketplace, buyers and sellers create their personal software agents that proactively seek out for each other and make offers. The concept follows a continuous double auction, but the implementation has some element of continuous English auctions and continuous Dutch auctions. Kasbah has initially been used for the selling and buying of used books, but has latter been extended to other domains such as CDs.

Upon creating an agent, the user is prompted to enter a description and the specifications of the product to sell/buy. This description will be used by other agents to find potential trading partners. The market system uses string comparison to find these matches. Since this method does not resolve the ontology problem, facilities to create a buying (selling) agent based on a selling (buying) agent already in the market is provided to limit the impact of the problem. However, it requires the users to search and browse the market to see if an item they want to buy or sell is already there.

The user is also required to enter the desired date to sell (buy) the item by, the desired price and the lowest (highest) acceptable price. These parameters define the agent's goal: to sell (buy) the product at the highest (lowest) possible price, starting from the desired price and reaching the lowest (highest) acceptable price at the expiry date. In terms of strategy, the user can specify how he wants his agent to proceed in lowering (increasing) the price as the expiry date approaches. More specifically, the user has a choice of three decay (raise) functions: linear, quadratic and cubic. Each function is respectively represented metaphorically by the terms anxious, cool-headed and greedy (frugal) agent. Overall, these parameters address the formulation of the negotiation problem.

Since it is a closed marketplace, i.e. the strategies of the buying and selling agents are created using the same unbiased system provided by the marketplace, exploitation of strategy is by design not a concern. However, a third party agent entering the system would be able to exploit this design limitation and take advantage of the other agents. As for the termination problem, the desired date to sell (buy) is used to end negotiation. It is also to be noted that the agents communicate using a home designed language and set of performatives.

Furthermore, the authors provide in the paper results and insights from a live-user experiment. From this experiment, several qualitative observations were made. In general, the feedback was positive as the participants thought using Kasbah was quite fun. However, the users were disappointed when their agent did "clearly stupid things", such as accepting the first feasible offer when a better one was available. Although this kind of behavior is unfortunate, it emulates the real world where timing is critical, as a better offer may always come not long after one has already committed itself in accepting a less interesting offer. Perhaps the main critic given by the users is that they feel it is a non-trivial burden to give the agent a precise set of instructions. Rather, they would have wanted the agents to act more pro-actively in terms of making decisions. For example, many users found that even specifying a desired price was a burden and would have preferred that their agents derive the information from the current market situation.

3.4.2 Tete-a-Tete

Tete-a-Tete is also a MIT Media Lab creation and is the product of Robert Guttman's master's thesis [Guttman 98c] [Tete-a-Tete]. Currently, Frictionless Commerce[9] is actively commercializing the shopping technologies behind Tete-a-Tete. In brief, Tete-a-Tete proposes to fix the merchant brokering stage of online shopping by guiding it away from price comparisons[10] and toward value comparisons by considering other qualities such as brand, customer service, delivery time, warranty, and other value-added services. In that sense, Tete-a-Tete is somewhat the extension of price comparison agents (such as [BargainFinder]) in terms of merchant differentiation, but also encompasses features of product comparison systems (such as [Jango] and [Compare.net]). Tete-a-Tete's goal is to seamlessly integrate the product brokering, merchant brokering, and negotiation stages of the online shopping process (see section 2.1).

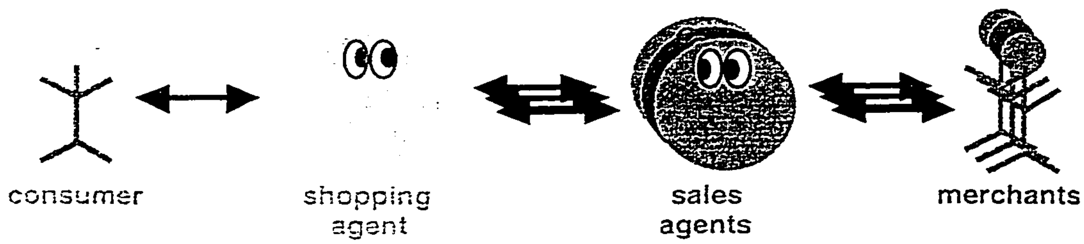

Figure 3.2 Tete-a-Tete Architecture[11]

The shopping process, as depicted in Figure 3.2, is initiated by the consumer who requests, through his shopping agent, some quotes for a list of products from the merchants' sales agents. The sales agents consult their merchants' catalogs and return the appropriate list of matching product. A decision support module (based on multi-attribute theory) is then used to rank the merchants offering based on the user's preferences, which consist of a weighted list of selected product features and merchant's value-added-service attributes. Furthermore, Tete-a-Tete uses an interaction protocol described by the author as a negotiation protocol based on bilateral argumentation.

[9] http://www.frictionless.com
[10] See [Guttman 98b] for a discussion on price comparison agents
[11] Figure from [Guttman 98a]

However in accordance to section 3.1.2.2, we believe the interaction protocol to be more of a *coordination* protocol because of the cooperative nature of the individual one-to-one interactions. We describe an individual interaction in Tete-a-Tete as a cooperative situation where the consumer approaches the merchant with a list of needs in terms of product, and the merchant willingly provides a list of products that meet the customer's requirement. There is no real conflict between what the consumer wants and what the merchant provides. Nonetheless, some could argue that there is conflict between the merchants. While this is true, the consumer is not bound to buy from the merchant that best fulfills his need (or any of them for the matter), and so the situation viewed from this angle does not qualify as negotiation under our definition (see section 3.1.1). The protocol would qualify as negotiation if the consumer was bound to buy from the vendor that best fits its need as part of a joint agreement and process between all the merchants involved.

As an ontology, Tete-a-Tete uses a shared database maintained centrally through a human editorial board similar to Yahoo. However and as pointed by the author, merchants seldom share the same ontologies[12] of products and the task of reconciling them is time consuming. For its part, the formulation of the negotiation is quite simple because there is almost no need for strategies given the cooperative situation.[13] The phrasing of the goals is also quite clear on both sides: the consumer identifies his needs; the vendor consults his catalog to see what products satisfy the consumer's needs. Because all the information is openly shared and there are no apparent strategies, the exploitation of strategy problem is not an issue. The termination problem is also not an issue since it is in the merchant's interest to always respond to a quote for products and because the making of a deal rests on the shoulder of the consumer, not his agent. Finally, we note that XML is used as the communication language and performatives are exchanged via TCP/IP.

3.4.3 AuctionBot

Developed at the University of Michigan, AuctionBot [Wurman 98] [AuctionBot] is a highly versatile and configurable Internet auction server that supports both human and

[12] Not to mention the same database.

[13] We say almost no need for strategies because the merchants could have some minor decision to make in choosing which product to return given only 5 matching products can be returned.

software agents. To the best of the authors' knowledge, AuctionBot is the only online auction site with explicit support for user-written software agents. Originally, it was designed to provide a comprehensive research testbed in market based resource allocation, but its usage has been extended to the general Internet population. The authors report that AuctionBot has been used extensively in the classroom, but that the volume of public activity was still small at the publication date, presumably because of the presence of large commercial sites such as EBay.

Through a web interface, sellers can become auctioneers and virtually create any type of auction they desire by specifying as set of parameters, such as the type of participation for the buyers and sellers (1-to-many, many-to-1, many-to-many), the bid rules, the clearing schedule, closing conditions, the allocation policy etc. AuctionBot can also support simultaneous auctions and bidders, sellers and auctioneer alike can monitor the running auctions directly from the web or be sent event notification by email. Perhaps its most interesting feature is that the users can write their own software agents and use them to interact and bid on their behalf in the auctions. The framework uses TCP/IP and the AuctionBot API message protocol. However when providing agents of their own, the users must explicitly formulate the strategy to be used.

Chapter 4
Critical Analysis

This chapter provides a critical analysis of the pertinence of each business model presented in Chapter 2, with regards to potential automated pricing solutions for retailing. Our goal is to determine which model is most suited for retailing and to delineate a promising solution to follow.

4.1 Many Buyers – One Seller

In this business model, buyers compete with one another for the seller's offering. Clearly, this is a departure from what happens in traditional retail markets where the competition (if any) is among the merchants, not the consumers. The absence of consumer competition in such markets can be partly explained by the fact that retailers typically sell production goods, i.e. goods that are available in fairly unlimited number and for which it is relatively easy to determine the marginal cost prices. All things considered, the selling of production goods follows a very different economical model than the selling of limited goods: when supply is sufficient, there is no competition due to demand.

Moreover, it has been shown that the relationship that most online retail merchants whish to have with their customers is not competitive [Forrester 97]. On the contrary, what these retailers really want is to keep their customer satisfied through a highly cooperative long-term relationship with them. Guttman and Maes note that "*unlike most consumer-to-consumer (e.g. classified ad) and commodity markets (e.g. stock markets), merchants often care less about profit on any given transaction and care more about long-term profitability*" [Guttman 98a]. By maximizing customer satisfaction, merchants hope to capitalize on repeat customer purchases. Additionally, they are staking that customer satisfaction will lead to additional purchases, either directly through word-of-mouth referrals or indirectly through positive reputation.

4.1.1 Many-to-One Auctions

Because of their growing popularity on the Internet, many-to-one online auctions could seem appealing to some retail merchants. Despite their competitive nature, they are well suited for electronic negotiation and have a certain advantages. First of all, the entertainment value of the online auctions is a non-negligible component, as it has been observed that the customers like the bidding frenzy created in an English auction [CMIT-1032]. But perhaps the main advantage is that merchants would no longer need to determine the price of their goods because this responsibility would end up in the hands of the consumers and the market. But as Guttman and Maes note, "*although auctions can relieve merchants of the burden of establishing prices for limited resources (e.g. fine art and stocks), this benefit is less realizable for production goods as in retail markets*" [Guttman 98a]. Based on Beam et al.'s work [CMIT-1019] on optimization problems for auctioning several identical items, Guttman and Maes support their claim by noting that it is non trivial to determine the optimal size of the auctioned lots and the frequency of the auctions for the selling of production goods. Hence, they say that the retailers are still burdened with determining a priori the value of their goods.

Still, a closer look at applying many-to-one online auction for retailing reveals several disadvantages. They are presented in the following sub-sections.

4.1.1.1 Winner's curse

The English auction is by far the most prevalent type of online auctions, with proportionally 85% of the cases as per a recent survey [CMIT-1032]. A reasons for this is perhaps that the English auction is well known and simple to understand. However, it has been shown that in an English auction, the winning bid is always greater than the product's market valuation. This downside for the buyer is commonly called the "winner's curse", and is due to the fact that the consumers valuations are private and can vary a lot from one bidder to the other. As Guttman and Maes say, "*although winner's curse is a short-term financial benefit to retailers, it can be a long-term detriment due to eventual customer dissatisfaction of paying more than the value of the product*" [Guttman 98a]. For limited resources such as collectibles, rare and used items, the winner's curse is acceptable because no one can tell exactly what the good is worth. In these conditions, the market value is considered a good reference for the product value and as such, buyers are generally satisfied with the price they paid. However in retail market, the product's value is usually much easier to obtain, especially considering the ease of access to information offered by the Internet and shopping agents. Hence, the discovery of such information would lead directly to consumer dissatisfaction. To make matters worse, the products are non-returnable, which means that customers could get stuck with products that they're unhappy with and paid too much for [Guttman 98a].

4.1.1.2 Delays

Another problem with most online auctions that could add to the customer dissatisfaction is the long delays between the start and the end of the auction. According to [CMIT-1032], 58% of the auctions surveyed ran over a period of 3 days or more, the majority (25%) closing once a week. Note that these numbers are partly clouded by the fact that the auction duration was unavailable in a surprising 28% of the auctions surveyed.[1] Guttman and Maes advance that these delays are "*due to communication latency issues and wanting a critical mass of bidders*" [Guttman 98a]. First of all, the presence of such delays is a clear departure from the conventional retail store way of selling products. In such stores, everything is for sale at all time and there are no delays to complete a transaction. Secondly, unless facilities exist to place phantom bids[2] or use a software agent, the consumer must follow up on the auction and continuously bid until the auction closes several days later. Additionally, since bids are non-retractable, consumers cannot consider other product offerings during these delays. Perhaps the biggest drawback is that only the winner of the auction can buy the good, meaning that the other bidders find themselves back at square one, i.e. they must wait until the good is auctioned again and undergo the whole process all over again. In summary, all these do not cater to impatient or time constrained consumers, let alone the impulsive buyers [Guttman 98a].

---

[1] This is by the way a major flow because it is strategically important for the bidders to know how much time remains to make bids.

[2] A phantom bid is one in which the bidder can privately tell the auctioneer the absolute maximum the bidder is willing to pay for a given auction. The auctioneer then proceeds on behalf of the bidder. [CMIT-1032]

4.1.1.4 Risks

According to the National Consumers League, 68% of frauds related to selling online in 1998 came from auction sites. Unfortunately, this number is growing compared to the 27% it was in 1997. There always have been risks inherent to conducting auctions. Still, it is much harder to detect fraud while conducting auctions online. There are typically two kind of undesirable, and most often considered illegal behaviors in auctions: *shills* and *collusion ring*.

As defined in [Guttman 98a], "*shills are bidders who are planted by sellers to unfairly manipulate the market valuation of the auctioned good by raising the bid to stimulate the market*". One thing with shills is that there is no negative consequence if the shill win the auction, the seller just has to re-auction the item. In the virtual world, it is very hard to detect shills because one has often no way to verify the identities of the participants, especially if the participant is a software agent.

For its part, a collusion ring is composed of a group of buyers who agree not to outbid one another, thus acquiring auctioned goods at a lower price. In a context of auctioning limited resources, the risk of seeing a collusion ring being formed is low because the goods cannot be redistributed. However in our case where the retailers would sell production goods, the same good would have to be re-auctioned again and again, thus encouraging the formation of coalition rings. In a physical auction, the risks of collusion is limited by the fact that, usually, people don't know one another before the auction and have no easy means of talking to each other during the auction. Interestingly, Beam and Siegev [CMIT-1032] report that, despite collusion risk, 16% of the English auction surveyed provided some sort of contact information about the other bidders on the site during the auction.

An additional concern comes from the fact that most of the auctions conducted on the web are self-hosted auctions, i.e. the retailer plays the role of the auctioneer. In these conditions, the marketplace is biased, as the seller can unfairly manipulate the outcome of the auction by withholding information, propagating misinformation etc. In order for the consumer to have trust in marketplace negotiations, it must be conducted by an unbiased third party.

4.2 One Buyer – Many Sellers

In this business model, it is the sellers who compete with one another for the buyer's patronage. Although this is a good model of what happens in a non-monopolistic retailing market buying situation, we note that the model is limited only to the buying of goods in such situations. Additionally, the infrastructure for this model is lacking, mostly because it is not in the interest of the sellers to see such a selling model arise. So even if such an infrastructure could be put in place (by either the consumer or a third party), it is uncertain that sellers will use such competitive channels to sell their goods.

4.2.1 Many-to-One Auctions

Also known as a *reverse* auction, a many-to-one auction is basically the mirror image of a one-to-many auction. For example, in a reverse English auction, the bids go down instead of going up, and the winning bid is the lowest bid as opposed to the highest. Because of their similarity to one-to-many auctions, they suffer at a lower degree from substantially the same problems. We note that while the winner's curse was at the advantage of the seller in a one-to-many auction, it is at the buyer's advantage in a reverse auction. Delays are still a concern, but not as bad a problem. Reason is the buyer doesn't have to monitor the auction and is at least assured to buy the product (from the winning seller) at the end of the auction. Even in auctioning a single unit of a good, the risk of seeing the sellers form a collusion rings is still present under the assumption that other customers exist and thus many of these auctions will be conducted. As for shills, it is possible but perhaps unlikely that the buyer could impersonate a seller, especially under the assumption that the infrastructure would be provided by an unbiased third party.

From a seller's point of view and considering the potential large amount of auctions they would be required to participate in, an automated solution for the seller would require the auction infrastructure to provide means of automation. This could translate in software agents or phantom bids facilities, with most probably a complex scheme of automated invitation and creation of agents upon the creation of auctions. Nonetheless, it is non-trivial to determine the optimal bidding strategy and to provide a level of automation for a large range of different goods.

4.2.2 Market Based Software Agent Pricing

This solution is very similar to a reverse auction, but it is a non-negotiated solution. The sellers still competes with one another over price and the seller with the lowest price still presumably wins, but the buyer is not bound to buy from the seller with the lowest price or from any seller for the matter.[3] Moreover, there is generally more than one possible buyer at a time in the picture. In other words, the situation can be described as pure market competition, with the underlined assumption that most, if not all buyers, will go for the merchant who has the lowest price for the same product. Instead of bidding like in a reverse auction, the sellers simply adjust their catalog fixed prices according to the competition price changes. Due to the necessity to quickly update the prices in response to competition price changes, this automated solution is most appropriate for software agents. "*Software agents are capable of making decisions orders of magnitude faster than humans, and can potentially base those decision on greater volume of much fresher information*" [Sairamesh 98].

By competing only on price, vendors are most likely to find themselves in price wars where they engage in price undercutting with one another to gain short-term advantage over the competition. In [Kephart 98], Kephart et al. have shown that in a large scale-economies of software agents, the potential exists for unending cycles of such disastrous competitive price wars. Furthermore, Kephart and others report in [Tesauro 98] that this situation is due to a number of differences between software agents and human players:

(1) greater ability of humans to predict long-term consequences of their price setting actions;
(2) reduced frictional effects such as consumer inertia in agent economies;
(3) reduced localization effects due to much greater connectivity offered by the Internet

Additionally, the authors note that such price and niche wars are damaging not only for the sellers, but also for the consumer in the long term.

[3] Although it might be in the interest of the consumer to buy from the seller with the lowest price.

4.3 Many Buyers – Many Sellers

This business model is characterized by the fact that both the sellers and the buyers have competitors in the market. Examples of such market include classified ads with both "for sale" and "wanted" sections, as well as financial markets such as NASDAQ. We note that retailing currently does not use this business model for the selling of goods, typically because channels to let consumers post binding requests for products are almost absent in the physical world. Nonetheless, third party companies such as Exchange.com[4] and Priceline.com[5] have started to put such facilities on the Internet with general success. Still, the current market for such companies is not retailing, most probably because the traditional ways of retailing are still well in place. As such, the appropriateness and success of this model for retailing is unclear, but we suspect it will be a while before resistance to change is overcome.

As for automated solutions, we note that the many-to-many auction, or *double* auction, is a likely choice. Other forms of online marketplace that dynamically match buyers and sellers could also be used. Two good examples of automated solutions for this model (although not for retailing) are the NASDAQ electronic stock marketplace and MIT's Kasbah (see section 3.4.1).

4.4 One Buyer – One Seller

This business model is best defined by the absence of market competitors in the Negotiation stage of the CBB. This can be explained by the fact that while market driven models tie the Negotiation stage to the Merchant Brokering stage, both stages are viewed as independent in the one-to-one model: first one determines who to buy from[6]; only then one determines the terms of the transaction. Still, we note that the market often, if not always, influences the terms of a transaction between any two parties. Nonetheless, the process of determining such terms engages only one buyer and one seller in the one-to-one model.

---

[4] http://www.exchange.com
[5] http://www.priceline.com
[6] Or *sell to* from a merchant point of view.

Because of this absence of market competitors, the one-to-one model allows for the desired cooperative relationship that retailers want with their consumers. For that reason and in comparison to the other models, we feel that this model is the most suited model for retailing. Still, care must be taken in choosing an automated pricing approach that does favor a cooperative long-term relationship between the retailer and the consumer.[7]

4.4.1 One-to-One Auctions

Because goods are allocated to those who value them the most, auctions ensure that participants reveal their true valuation of the good being auctioned. In a one-to-one auction, such a truth revelation characteristic would be useful in determining a fair price between the seller and the buyer. Moreover, a one-to-one auction would not suffer from the same problems and limitations of most other types of auctions. Because such an auction would involve only two parties, there would be no delays in reaching a critical mass. For the same reason, illegal behavior such as shills or collusion ring wouldn't even be a concern.

Unfortunately, auctions rely on market forces to accomplish their goals. Without competing bidders or bids, an auction looses most if not all of its powers and benefits. For example in a one-to-one context, a one-sided auction such as the English auction would be the equivalent of telling the customer "Your price is my price"... [8] As for double-sided auctions, it has been shown that under modest conditions, no mechanism exists that would ensure that both parties reveal their true valuation of the good, i.e. how much they are willing to buy/sell [Myerson 83]. As a consequence, it is impossible to design a one-to-one auction that would determine a fair price between the buyer and the seller. Moreover, trade efficiency cannot be guaranteed. This deserves more explanation.

Let RP_{buyer} be the reservation price of the buyer and RP_{seller}, the reservation price of the seller. Initially, both reservation prices are private. Suppose without loss of generality that $RP_{buyer} > RP_{seller}$, i.e. that a zone of agreement exists. In these conditions, a fair price for both

[7] Here, we are referring to the fact that negotiation is sometimes perceived as a win-loose painful and adversarial process. If an automated negotiation approach is used, it must be pain free and perceived as a win-win situation.

[8] A reservation price below which an offer would be considered unacceptable could be specified, but it would be no better than having fixed a price in the first place.

parties would be $(RP_{buyer} - RP_{seller}) / 2$.[9] Hence, what we are seeking a mechanism that would have both parties reveal their reservation prices in order to make such a calculation. Such a revelation can be done in two ways: turn taking revelation or simultaneous revelation of RPs. Suppose in turn taking revelation that the first party reveals his true RP. In this case, the second party has incentive to lie about his RP by revealing a false reservation price that is closer to the one revealed. Knowing this, the first party has also incentive to lie about his RP.

Now suppose that by some means, both parties reveal their RP simultaneously, wherein the trade price would be the middle point between the two. In these settings, both parties can influence the trade price and thus have incentive to lie when revealing their RPs. All in all, because parties are bound to lie, there will be situation where a zone of agreement will not exist in the reservation prices that were revealed. In these situations, no trade will be made, even if $RP_{buyer} > RP_{seller}$. Consequently, the mechanism is not trade efficient.

4.4.2 Negotiation Support Systems

While Negotiation Support Systems (NSS) aim at providing computer assistance and automation in decision making, they are not designed to support fully automated negotiations. They are meant to work exclusively with human parties on all sides of the negotiation process and require near constant human input. Since our goal is to design an automated system that is able to autonomously negotiate on behalf of the retail vendor, Negotiation Support Systems do not fit our needs.

However, the preparation module of such tools can be used as a starting block for designing a fully automated system. In particular, the work in preference representations, knowledge elicitation and pre-negotiation strategies (discussed in section 3.3.1) can be combined with the work in agent technology to create rational decision making software agents.

[9] In economic, profitability is more important then fairness and as so, the goal of an auction is to maximize the gains from trade. In this case, the buyer would have to pay RP_{buyer} instead of $(RP_{buyer} - RP_{seller}) / 2$.

4.4.3 Information Driven Software Agent Pricing

Our approach to price negotiation in retail markets proceeds as follows: provide the retailer with a software Sales Agent (SA) that could dynamically give or negotiate just-in-time personalized prices to consumers. The agents would base its decisions on information such as consumer history of purchase, recent product sales, total purchase of transaction, number of items in the vendor's inventory, retail cost prices etc. Note that the competition could be factored in the pricing decision, but would not be the sole factor in the equation, contrary to the market driven software agent pricing approach. While this approach might be too demanding for human agents, it is well suited for software agents because "they *are capable of making decisions orders of magnitude faster than humans, and can potentially base those decision on greater volume of much fresher information*" [Sairamesh 98].

The reason we foresee this approach as promising is that it has several advantages. First of all, unlike other automated approaches presented, it is not limited to the selling of one product at a time nor is it limited to the single negotiation issue of price. Additionally, the decision-making is rational, as the software agent can justify with clear facts why it gave this price to this consumer. We claim that the capacity of the agent to motivate his decision leads directly to the retailer's trust in his agent. This is a key factor in bringing agent technology to use in e-commerce. Also, this type of approach allows for "full-blown" negotiations, i.e. negotiations where no conflict resolution mechanisms are used to resolve conflicts and wherein all parties have to agree explicitly and directly to the last offer made in order to reach an agreement. In our one-to-one context, we argue that "full blown" negotiation is the logical form of negotiation to use, given that there exists no appropriate resolution mechanism for one-to-one negotiation.

Then, there is the issue of whether providing personalized pricing is sufficient or should the approach also involve the consumer in setting the price by negotiating it. To pursue our goal of flexibility in pricing, a negotiated approach would undoubtedly provide more flexibility then a fixed price approach, whether they be personalized or not. Following the line of thought of Beam and Siegev in [CMIT-1032], we think that the entertainment value of the interactive nature of negotiating the price is a non-negligible component for it to be widely used by customers. Furthermore, we also think that giving the consumer the power to participate in setting the price of the products he/she is buying is a strong marketable asset that could create the same kind of acceptability and popularity that online auctions have created in recent days.

Finally, there is the issue of whether the consumer should also be represented by a personal software agent. The literature uses the term "fully automated" negotiations when all parties are represented by a software program, and "semi-automated" negotiations when humans are negotiating with software programs [CMIT-1019]. However, this terminology suggests that fully automated negotiation is more desirable than semi-automated negotiation, which might not always be the case because automation comes with a cost. In our case, we feel that a semi-automated approach is more appropriate because it relieves the consumer from the unnecessary burden of creating and providing strategies to his software agent every time he wants to buy something. Unlike the automation of the selling of goods, there is not much to gain in automating the buying process unless in a business-to-business market. Moreover, it is unlikely that the consumer could come up with a general buying strategy that could apply in all situations. Nevertheless, the presence of a consumer shopping agent is not incompatible with our approach.

4.5 Summary

In summary, the appropriateness of a particular business model for the selling of goods is not only driven by the type of goods being exchanged, but also influenced by the type of relationship merchants want to maintain with their customers. Because retailers sell production goods and desire to have a cooperative long-term relationship with their customers, retailing lures itself easily to the one-to-one business model.

Additionally, current approaches to fully automated pricing solutions for electronic commerce not only suffer from drawbacks such as winner's curse, potential risks and likelihood of price wars, but are also limited to the negotiation of one product at a time over the single issue of price. This limitation can be explained by the fact that under a market driven context, price is usually the mean to resolve a conflict, not the issue of conflict itself. Consequently, current automated solutions exploit the fact that there is a resource conflict between parties over one instance of a good. This is due to the nature of trading, wherein individual transactions can occur only between two parties.

Moreover current automated negotiation solutions such as the various auctions provide automation at the process, or *protocol*, level, but not at the participant level: bidders are still burdened to determine goals and optimum strategy for them. Furthermore, trading partners no longer negotiate in the true sense of the word. Instead, the negotiation phase is replaced by the statement of whether the public conditions under which contracts will be concluded are given or not [Reimers 96]. In a way, negotiation skills are replaced with market forces.

Finally, our proposed information driven software Sales Agent (SA) is an automated pricing solution that we claim well suited to retailing. Reasons for this include the fact that in retail markets, it is relatively easy for the vendors to determine the marginal cost price of their offering, including products and value added service. Therefore, it is also easy for them to determine the kind of profit margin they want to make on any given transactions. Equipped with vendor cost prices and profit margins knowledge, a software agent could compute a dynamic price for a specific offering of products and value added services. Moreover, it can use fresh volume of relevant information as input to these calculations. In Chapter 5, we propose to use such a dynamically computed price as a threshold for the lowest offer acceptable in the negotiations. Additionally, the approach of personalized pricing and negotiation has the advantage that individual transactions can be kept secret, hence not influencing the market and future transactions.

Chapter 5
Research Methodology

In this chapter, we provide the requirements and specifications for implementing the information driven software Sales Agent we proposed in Chapter 4. At the process automation level, we tackle the issue of designing a negotiation protocol and propose one that meets some basic desiderata (section 5.1). At the participant level, we discuss the knowledge driven process of specifying the vendor's goals and strategy to an automated software Sales Agent (section 5.2). More specifically, we propose a methodology to calculate a "just-in-time personalized price" based on "worth" associated to different information factors. Additionally, we present a strategy that uses such a dynamically computed price in the negotiation with the consumer. In section 5.3, we discuss potential solutions to cope with profit losses due to negotiation.

5.1 Process Automation: The Negotiation Protocol

As we've mentioned before, the specification of a negotiation protocol can have substantial, rippling effects on the nature of the overall system [Rosenschein 94]. Consequently, one must take special care in designing such a mechanism. In that sense, we feel that a one-to-one negotiation protocol for E-commerce should have the following design goals:

1- Should be intuitive and easy to understand for the consumers.
2- Allow both the consumer and the SA the possibility to make offers and counteroffers.
3- Allow both parties the option of refusing an offer made by the other party.
4- Allow the no deal option
5- Favor a fair process, i.e. not totally to the consumer's or vendor's advantage.
6- Address the "exploitation of strategy" problem.
7- Address the "termination" problem.
8- Should not introduce significant delays in the buying process.

In the following section, we propose a protocol that meets such requirements.

5.1.1 A Protocol Proposal

We have designed our negotiation protocol by modeling upon what typically happens in price negotiations in a physical retail store. In such negotiations, a consumer and a salesperson engage in a process of making offers and counter-offers in order to try to come to an agreement. Our approach to the problem is to 'mimic' such a negotiation scenario. By doing so, we are catering to design goals 1 to 4. A key characteristic of such a scenario is that an initial offer is already on the table, namely the original fixed price of the retailer. Moreover, this offer is typically always available even if negotiation fails. So even if no agreement was reached at one point in time, the negotiation cycle never really ends because an offer is always available for the consumer to accept at a latter time. Figure 5.1 provides a state transition model that takes into consideration these characteristics in depicting the possible interactions in such a negotiation scenario between a retail vendor and a consumer under our protocol. Note that as the model is state driven, the interaction takes the form of a turn-taking exchange of offers.

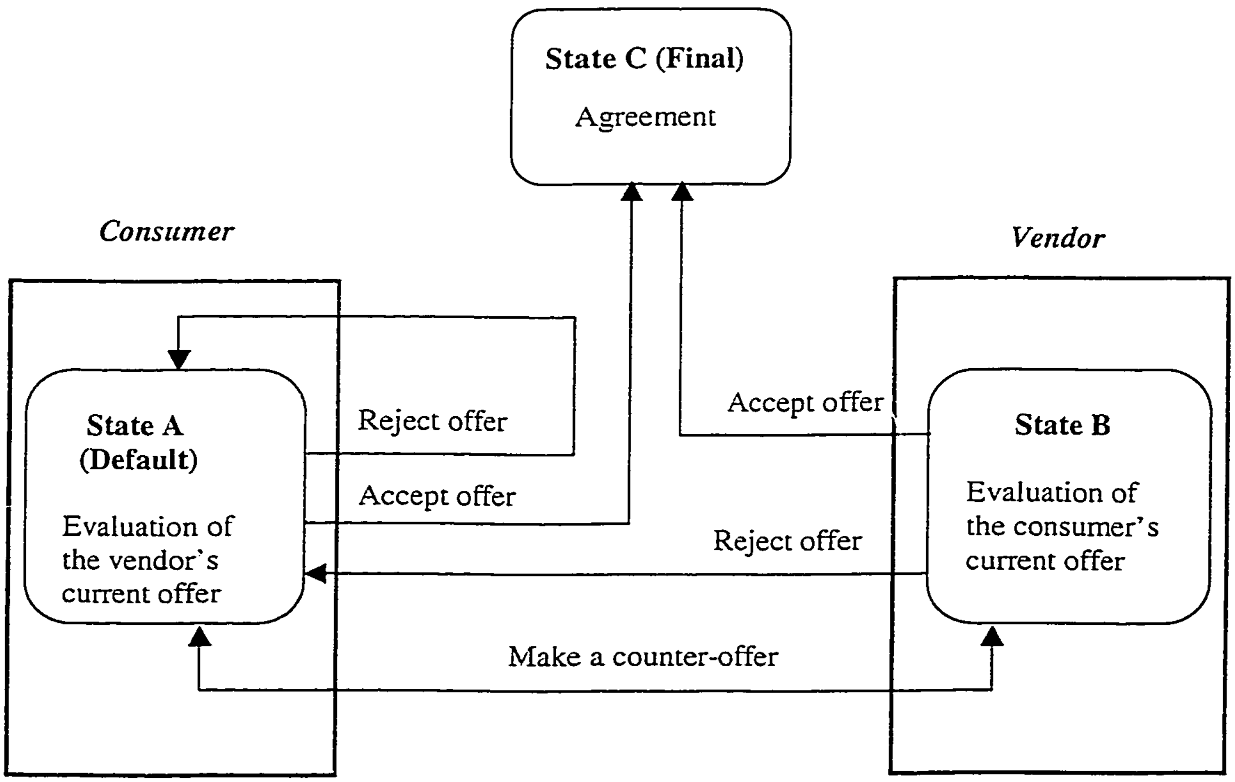

Figure 5.1. Negotiation Between the Vendor and the Consumer

The figure shows that the initial and default state is *State-A*, i.e. the state where the consumer evaluates the vendor's initial offer or current counter-offer. The vendor also has a corresponding evaluation state, which is represented on the figure by *State-B*. Since the vendor already made a first offer, we explicitly chose *State-A* as the initial state to address our design goal of fairness. Furthermore, we feel that turn-taking will ensure fairness in the overall negotiation process. So initially, the "ball" is in the consumer's court as he/she can decide to do one of the following three things:

1- Accept the current offer (an agreement is reached);

2- Reject the current offer (no agreement reached yet);

3- Make a counter-offer (start a round of negotiation).[1]

If the consumer decides to make a counter-offer and start a round of negotiation, the "ball" then moves to the vendor's court. To transit out of *State-B*, the vendor has the same three options that the consumer has in *State-A*, namely to accept or reject the offer or to counter-offer. Note that in the way we modeled the interaction, a round of negotiation is always initiated by the consumer. Thus, the vendor finds himself in *State-B* if and only if the consumer makes a counter-offer. Furthermore, it's not up to the vendor to decide that no agreement will be reached, as the consumer always has the choice of accepting the last offer of the vendor or the initial fixed price offer. Finally, the process can loop back and forth from *State-A* to *State-B* until either side decides to move to the end *State-C*.

To address the exploitation of strategy problem, we need to prevent situations where the consumer attempts to get the best price possible by starting bidding with a very low offer and slowly increasing his offer till it is accepted. We also need to prevent another similar scenario where the consumer makes very low increment counter-offers and waits for the sale agent to make successive counter-offers down to its lowest price. To prevent these situations from happening, we propose to: (1) notify the consumer that each offer he/she makes is a commitment to buy at the offered price[2] (2) impose to the consumer a maximum of one offer (and thus guess on the vendor's reservation price) per negotiation session.

---

[1] We define a round of negotiation (initiated by the consumer) as the transition from *State-A* to *State-B*, followed by a transition from *State-B* to either *State-A* or *State-C*.

[2] Mechanisms to enforce such commitment need to be put in place.

To implement such an idea, we propose to impose *significant* delays between rounds of negotiation, i.e. between the time the consumer first makes an offer and the time he is allowed to make the next one. The idea is that if a consumer is fairly committed to buy something now, hopefully he/she might not be willing to wait X units of time to pursue negotiations if X is sufficiently high. Furthermore, we note that this solution is not incompatible with our goal of not introducing delays, as there is always an offer on the table that the consumer can accept immediately, namely the fixed price offer or the current counter-offer from the vendor. Finally, we address the termination problem for the retailer by discharging the responsibility to end the negotiation to the consumer.

5.1.2 Summary

By modeling on the real world, our protocol is intuitive and offers the basic desiderata that one would expects from a negotiation protocol. Additionally because the vendor is at a disadvantage by making the initial offer[3], our proposed protocol reestablishes the balance by letting the consumer make the next offer. For the vendor, this means that the consumer might offer more than what the vendor was willing to go down to. The protocol is also restrictive enough to limit progressive negotiation to discover the vendor's bottom price. Ideally, the protocol would also ensure that the consumer reveals his true valuation of the goods. i.e. the maximum price he/she is willing to pay for them. Unfortunately under the impossibility results from the literature (section 4.4.1), there exists no trade efficient mechanism that can measure the consumer's willingness to pay in a one-to-one negotiation. Still, the protocol does incites the consumer to give his best offer, since not doing so would increase the chance of being rejected and hence, having either to pay the full price or wait before negotiating again.

Additionally, our protocol has the advantage of keeping delays to a maximum of one round of negotiation. While multiple rounds of negotiations serve a purpose in multi-dimensional issues negotiations to increase efficiency in reaching agreements, we feel it is superfluous in single-issue negotiations where we want to prevent the exploitation of private valuation through progressive revelation.

---

[3] Because the consumer might have been willing to pay more.

5.2 Participant Automation: The Software Sales Agent

How can we implement the notion of goals and strategy on the vendor's side? To address this question, we will use the notion of *point of diminished return*. Intuitively, the point of diminished return is the threshold after which the cost of doing something becomes greater than the payoff it brings. In the case of price negotiation for a vendor, it is the bottom price after which the expected gain of reducing the price further is no longer deemed worth it in the situation at hand. Our intention is to use such a threshold price as the vendor's reservation price in the negotiation with the consumer. Moreover, we propose a systematic information driven methodology to design a software Sales Agent (SA) that would calculate this point of diminished return for the vendor.

Our goal is to provide a methodology that is easy for the vendor to understand and put into practice, yet powerful enough so that the software Sales Agent created could handle different situations. For example, the SA should be able to negotiate on the full "shopping cart" of the consumer, not just over a single product. Still, reaching this goal is not easy given the formulation of the negotiation problem (section 3.2.2). The biggest challenge yet is to extract relevant negotiation information from the retailer. On this matter, Carrie Beam and Arie Segev raise the need for companies to formulate a bargaining strategy with buyers in terms of overall corporate negotiation policy [CMIT-1016].

Additionally, while Artificial Intelligence techniques can support complex reasoning systems, Chavez and Maes found out in their studies on Kasbah that a key factor to the success of eliciting a negotiation strategy from the user was to use a simple and intuitive negotiation strategy over a complex one. This ties directly to the formulation of the negotiation problem, but also involves trust issues in the agent's decision making capabilities. In order for the user to have trust in his agent, he/she must understand what it is doing. To cope with this, our approach is to use a knowledge-based system, and incrementally add more complexity to the system if need be. Since knowledge-based systems allow for backtracking the supporting facts for a decision, logging could be done. This way, the vendor would not have to rely blindly on the initial design, but would have records of the negotiations and could refine his agent if necessary.

5.2.1 Methodology: Phrasing Goals

In our studies to determine how to phrase the retailer's negotiation goals, we tried to find the factors that could be of valuation to retailing, i.e. factors that could justify selling below the fixed price for a given transaction with a specific customer. As a result, we came up with the following non-exhaustive list of such factors:

- A good customer
- A substantial total bill for the transaction
- A recent history of low sale volume for a product
- A very high inventory for a product

Assuming appropriate data is available in the retailer's information system, our intention is to use such factors in calculating the point of diminished return for a given transaction. Additionally, the retail cost price and retail fixed price of the products are of importance for our calculations. We define the retail cost price as the price at which the retailer makes no profit for a given product. In the same vein, the retail fixed price is the public listed catalog price for the product. We assume that both the retail cost price and the retail fixed price are available and a priori set by the retailer, and that the retailer will not sell below the retail cost price.[4]

Let:

TCP = the transaction retail cost price

TFP = the transaction fixed price

PDR = the point of diminished return

TPM = the transaction profit margin as TPM

Then:

$$PDR \in [TCP , TFP] \qquad (5.1)$$

$$TPM = TFP - PDR \qquad (5.2)$$

$$TPM \in [0 , TFP-TCP] \qquad (5.3)$$

[4] Note that in [Tesauro 98], it was shown that humans are better to set prices than software agents are. As a consequence, we did not give our SA the responsibility to set the retail fixed price.

Our idea is to have the retailer measure the worth of the various information factors in terms of a percentage of the transaction total fixed price, which we feel is an intuitive parameter. However when giving rebates in term of a percentage of the fixed price, there is a risk that the overall rebated price ends up under the cost price. Hence in calculating the PDR, we take the maximum value between TCP and the overall rebate percentage.

$$\mathrm{PDR} = \mathrm{Max}(\mathrm{TCP}, \mathrm{TFP} - X\%) \qquad (5.4)$$

At this point, it becomes necessary to develop mathematical functions to represent the information factors. Our idea is to use fuzzy logic membership functions to map the possible values of each factor into the interval from 0 to 1, where 0 has no membership value and 1 has full membership value. Consider the following mathematical representation of such a membership function $m(x; \alpha, \beta)$ for a given factor F measured in terms of variable x:

$$\left[\begin{array}{lll} m(x;\alpha,\beta) = 0 & \text{for } x < \alpha & \text{not a member} \\ m(x;\alpha,\beta) \in\ ]0,1[& \text{for } \alpha <= x < \beta & \text{a member} \\ m(x;\alpha,\beta) = 1 & \text{for } x >= \beta & \text{a full member} \end{array}\right. \qquad (5.5)$$

In (5.5), α and β are parameters to the function, wherein α is the lowest value for which x is considered a member of F and β is the highest value after which an increase of x does not increase the membership value anymore. Figure 5.2 shows an example of a membership function.

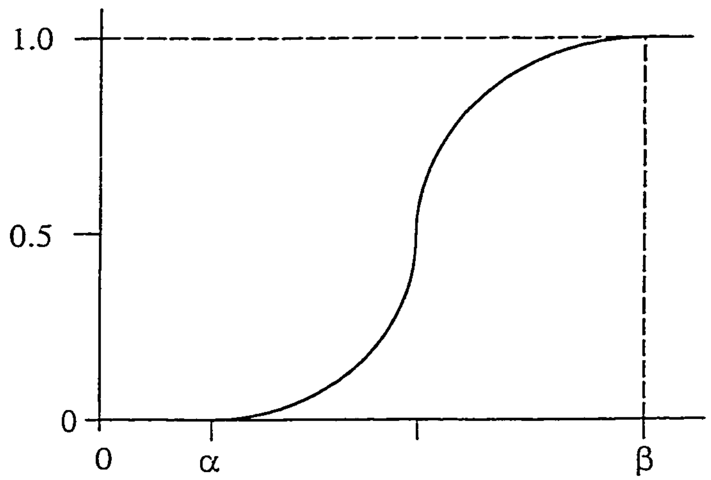

Figure 5.2 Membership Function

Depending on the choice of the membership function, the manner in which the mapping from x to the interval from 0 to 1 is done can vary. More specifically, m could be a continuous function such as a linear, logarithmic or exponential function, or it could be a function by part, such as a simple step function or the S-function shown in figure 5.2.

As the interval from 0 to 1 has no intrinsic value for retailing, a logical approach would be to map the 0 to 1 interval back to an interval with more significant values. Consequently for a given factor, we propose to use the lowest and highest worth accorded by the retailer for this factor. Respectively, a $0 + \in$ grade of membership would map to the lowest worth accorded to the factor while a 1 grade of membership would map to the highest worth. Figure 5.3 shows a concrete example of what the function could look like for the factor "Total Bill".

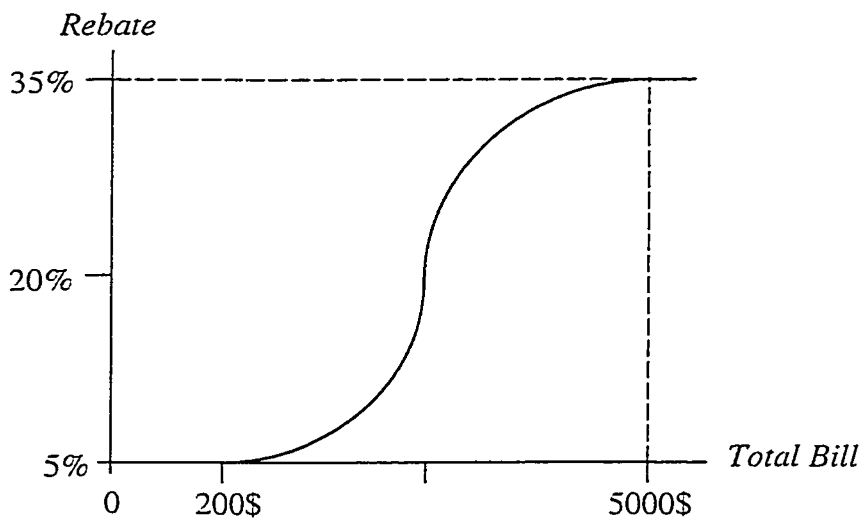

Figure 5.3 Membership Function for Total Bill

Note that Figure 5.3 differs from Figure 5.2 only from a unit change at the Y-axis: the function and the scale of both axes did not change. Furthermore, note that the unit on the X-axis depends on the factor at hand. We can deduce from the figure that to be worthy of a rebate, the total bill must be over 199.99$. Additionally, a total bill of 200$ would yield a 5% rebate, while a total bill over 5000$ would produce a saving of 35% to the consumer. The maximum rebate for the "Total Bill" factor is 35%, and it is reached when the total bill is equal to 5000$.

To recapitulate, our methodology requires that the retailer identify a list of information factors that could justify selling below the fixed price. Additionally, the retailer is required to provide the following data for each factor F:

- a membership function m;
- the nature of x;
- the membership boundaries α and β;
- the corresponding worth in terms of TFP at α and β.

An example of end result data derived from our methodology is shown in Table 5.1

Factor	$m(x)$	x	α	β	worth at α	worth at β
Profitable customer	S-function	purchase history ($)	1000	10000	15%	25%
Substantial total bill	linear	total bill ($)	200	1000	5%	15%
High inventory	Quadratic	items in inventory (n)	500	1500	10%	35%

Table 5.1 Example of End Result Data

The question that next arises is how to combine the worth of the different factors into an overall percentage rebate to be used to calculate the point of diminished return. The simplest method, and the one we propose, is to treat each factor as being independent of each other and sum up their individual worth.[5] Still, when summing percentage rebates, there is a risk that the overall rebate be greater than expected and thus introduce undesirable profit cuts. To cope with this, the retailer could specify a maximum percentage rebate (MPR) threshold. Even if no such threshold is used, the worst case scenario will be that the transaction will occur at the total retail cost price.

$$\text{PDR} = \text{Max}(\text{TCP}, \text{TFP} - X\%, \text{TFP} - \text{MPR}\%) \qquad (5.6)$$

[5] Other methods could also be used to combine dependent factors without loosing the essence of what we are doing.

5.2.1.1 Discussion

Because retailers are not used to articulating the kind of data our methodology requires from them, the elicitation process to gather the data is bound to be difficult. As such, it is the most critical part of our overall proposed solution, as it is essential for its practicability. But, however important the issue is, it is outside our scope in this thesis to address it. Still, knowledge engineers and simple elicitation programs could be used to assist the retailer in this task.

5.2.2 Methodology: Phrasing Strategies

Once the software Sales Agent knows how to calculate the PDR, it needs to determine an appropriate negotiation strategy to use under our negotiation protocol. More specifically, we need to provide a set of rules for the Sales Agent so that it can regulate the making of counter-offers and the acceptance of consumer offers. Since by definition anything above the PDR is deemed profitable, we propose to use the PDR as grounds for the Sales Agent reservation price and use the simple strategy of accepting any offer equal to or over the PDR (hence rejecting any offer below the PDR). In the case where SA rejects the consumer offer, we suggest that the SA makes its counter-offer at the PDR value. See Figure 5.3 for pseudo code of this strategy.

```
#Decision rule for SA
IF Consumer Offer >= PDR THEN
        ACCEPT the offer
ELSE
        COUNTER-OFFER at price  = PDR
```

Figure 5.4 Decision Rule for SA

The choice of the above strategy is motivated by the following reasons: (i) it is an intuitive strategy that the retailer can easily grasp and understand; (ii) it allows for easy a posteriori observation of the Sales Agent decision making through a log of the negotiation session; (iii) the strategy is trade efficient under basic desiderata. Recall that a mechanism (in this case a strategy) is trade efficient if a trade always occurs when there exists a zone of agreement between the two parties, i.e. a trade always occurs if the consumer reservation price is greater than the vendor reservation price. Assuming that a zone of agreement does exist, the revelation of the vendor reservation price in a negotiation session would ensure trade efficiency because the consumer would know that no more concessions could be extracted. Consequently in the case where the Sales Agent rejects the consumer offer and makes a counter-offer at the PDR, SA should inform the consumer that this offer is the best it can do, which is true by definition of the term "point of diminished return".

Still, while intuitive and trade efficient, our strategy might not be the most profitable strategy for the retailer. Consider the situation where a consumer is willing to pay the TFP and can engage freely with relatively low cost in price negotiation with our Sales Agent. The likely scenario to occur is that the consumer will make an offer below the TFP, which will probably lead to a direct profit loss of (TFP –PDR). However, the problem is not so much with our strategy than it is with the nature of both retailing and the one-to-one negotiation problem. Because retailers usually do not have nor the power or the interest to explicitly refuse to sell to at the list price, the consumer has nothing to loose to attempt bargaining knowing that the fixed price is available to fall back to.

Additionally under the impossibility results from the literature (section 4.4.1), it is not clear where is the balance between: (1) loosing profit because high reservation prices lead to loss of trades; (2) making more profit on trades that do occur at these high prices. Under these circumstances, the best the retailer can do is to view the PDR as a personalized fixed price, whereas anything he gets over it through negotiation is a surplus. But the vendor shouldn't expect it. It is up to the retailer to set the PDR accordingly to the profit margins it wants to get. Finally while coping with profits losses due to negotiation is outside the scope of this thesis, we do provide potential solution avenues in the next section.

5.3 Discussion: Coping with Profits Losses

5.3.1 Selective Negotiation

This approach to the problem attempts to model upon the real world. In a physical retail store, it is often unclear to the consumer if prices are negotiable, as a fixed price policy is assumed in most cases in North America. Such a situation suggests selectively offering negotiation only to the consumers that are hesitant to buy at the fixed price. However, the way such a thing can be done in the online world is not trivial. It suggests inferring the consumer's interest and defining the conditions under which we can consider that the consumer has decided not to buy anything from the store. In the real world, interest in products can be inferred by seeing the consumer look at the products, try them on etc. In the same vein, decision not to buy can be derived by the fact that the consumer is physically leaving the store without buying the items he was interested in buying. In the online world, it is uncertain how such conclusion could be derived from monitoring the consumer's browsing from one web page to another.

However, if such a thing could be done, the Sales Agent might go to the hesitant consumer and offer him the possibility to enter price negotiations. Note that by keeping the ability to bargain online a private and selective matter, the vendor cannot make use of the marketable value of offering price negotiation to his site.

5.3.2 Restricting Strategy

In this approach, we ask the consumer to give us the maximum amount he is willing to pay for the products. The actual implementation is similar to the one proposed for our strategy, whereas the difference is that the agent doesn't make any counter-offers and, given the consumer's offer was rejected, denies the consumer the option to buy at the listed price. In other words since the agent asked the consumer to give his "best offer", it takes actions to enforce and ensure that it is. However, the viability of such an approach would need to be studied, as it is counter-intuitive to conventional retailing practices.

5.3.3 Putting Cost to Negotiation

Even if a fixed price policy is assumed, it wouldn't hurt people in physical retail stores to attempt negotiating with the salesperson anyway, because it is in their own interest to do so. Why is that so? The worst that can happen is that the salesperson replies that he can't cut prices. The thing is that it is often considered an embarrassing or painful process to ask for a better price. People often think they are breaking a rule, or will feel cheep if they ask for price negotiation or reduction. People also fear that the process might take time, that the sales person will be bothered, make fun of them or simply that the clerk contacted won't have the proper authority to give a price reduction.

Because of this, people are often shy and fall short of asking for price negotiation or reduction. Consequently because they feel negotiation comes with a cost, only a small number of consumers go out of their league to actually ask for price negotiations. These customers have perhaps just more guts than others, but most of them are most likely assumed not willing to pay the fixed prices. By purposely adding cost to the negotiation process, perhaps only the consumers that are not willing to pay the fixed prices will engage in price negotiation. To dissuade the non-serious consumers, cost could be added by charging negotiation fees or by adding complexity and delays to the negotiation process. However, it is unclear of what should be the fees or if fees should be charged if negotiation fails.

5.3.4 Risk Evaluation Strategy

Another approach is based on the fact that, even thought it would be in the vendor's interest to accept any consumer offer over the PDR, the vendor could want to gamble by rejecting the offer anyway, in the hope that the consumer will still buy at the fixed price. Such a situation could be implemented by having the vendor specify a probability of acceptance threshold. In other words, a retailer could tell his Sales Agent to accept 80% of the offers that are equal or above the PDR. The disadvantage of this strategy is that it filters all types of consumers, the ones not willing to pay the TFP as well as the others. Hence, it is not trade efficient.

The strategy could perhaps be improved by using the following heuristics: 1) the closer the consumer offer is to the TFP, the higher the chance that the consumer will pay the TFP anyway; 2) the farther the consumer offer is from the PDR, the less serious the offer is assumed to be and higher the chance that the consumer will pay the TFP. Still, the validity of such heuristics remains to be tested.

Chapter 6

A Software Agent and Multi-Agent System Prototype

In this chapter, we present the software Sales Agent prototype and the Multi-Agent System architecture we designed under the CITR E-commerce project. To provide context to our work, section 6.1 describes the overall CITR project's setting, with focus on our part of the project at Concordia University, i.e. the User Interface and Intelligent Agent subproject. Section 6.2 presents the Multi-Agent System (MAS) architecture with the inter-agent communication language we developed for the project, while section 6.3 provides implementation details of both the SA and MAS prototype.

6.1 CITR Project – Enabling Technologies in Electronic Commerce

The setting for the overall CITR project is that of a virtual shopping mall with multiple independent vendors. In such a mall, the human consumers (or users) are represented on the screen by 3D animations called "avatars". The users can navigate their avatar through the virtual environment, interact with other avatars, visit some stores and look at 3D representations of products. Through a personal User Interface Agent (UIA) [Lu 99], users can also make sophisticated searches for items of interest in the mall's catalog, add items to a shopping cart and send out purchase orders to the appropriate software Sales Agent (SA). In this context, we designed and implemented the software Sales Agent prototype. We also developed a Multi-Agent System with inter-agent communication language as part of the requirements for the User Interface and Intelligent Agent CITR subproject.

Note that in an endeavor like electronic commerce, it is quite natural to employ multi-agent technology to communicate with other agents. For example, a User Interface Agent may want to consult another UIA about the quality of a product, the reliability of a vendor, or the level of satisfaction of the services provided by a vendor. In the end, there might even be special type of agents who have accumulated more experience and have become some sort of "Better Business Bureau" source of information.

6.1.1 User Interface Agent

As mentioned above, instead of navigating through the virtual shopping mall, the user may activate an intelligent user interface agent for retrieving items of interest [Lu 99]. The UIA converts the user's request to a SQL query, sends the query to a remote multimedia database server and returns the matching information. The UIA is designed to be capable of dealing with user's incomplete or ambiguous queries by making use of context-based substitution according to the user's profile. More specifically, the user profile takes the form of the UIA's internal knowledge base, derived from a user model that includes tasks, preferences, constraints, and the user's shopping characteristics and choices [Lu 99]. Overall, the UIA can assist the user either reactively by responding to user actions, or proactively by monitoring certain events and drawing the user's attention if necessary.

By applying machine learning techniques while monitoring, a UIA can observe the user's behavior and incrementally add attribute values to the profile. For example, a UIA can use the past behavior of a user to make reasonable guesses about the user's preferences and interests, e.g., his preferred store to purchase certain products, his usual price range, his favorite manufacturer (brand), etc. For this purpose of anticipation, it becomes necessary to characterize the "situations" under which observed attributes can contribute to learning and to decide which learning methods are better suited for the selected domain of application.

6.1.2 Software Sales Agent

In the CITR project, the software Sales Agent is the logical entity to represent the vendor in the mall. Moreover, each retailer in the mall has one instance of a Sales Agent to handle negotiation and manage incoming customers' offers.[1] Additionally, the SA responds to a consumer's offer by making use of "decision rules", some of which are based on private information (such as cost prices and inventory numbers) not available in the public catalogue of the mall.

[1] Although for performance reasons there could be physically more than one instance of the SA software, it is useful to think of it as a single entity.

A response consists of either the acceptance or rejection of the consumer's offer. In case of acceptance, the SA notifies the user that the transaction occurred at the consumer's offered price. In case of refusal, no transaction takes place and the user is notified of the refusal. In a refusal notification, the SA has also the liberty of making a counter-offer, which the consumer can either accept or refuse. Acceptance of the counter-offer when communicated to the SA leads to the completion of the transaction. Both the consumer offer and the SA counter-offer are valid for a period of pre-determined duration. Moreover in this period, no other offers can be made by the consumer for the same items under our negotiation protocol.

6.2 Prototype Design

6.2.1 General Architecture

Our architecture is that of a distributed Multi-Agent System (MAS), wherein each vendor in the mall is represented by one instance of a SA and similarly wherein each consumer is represented by a User Interface Agent. All agents are separate entities running independently in their own process, possibly on different machines. UIAs are created as users log into the virtual mall, while the vendors' SAs are assumed to be continuously running. Figure 6.1 provides a graphical representation of the architecture.

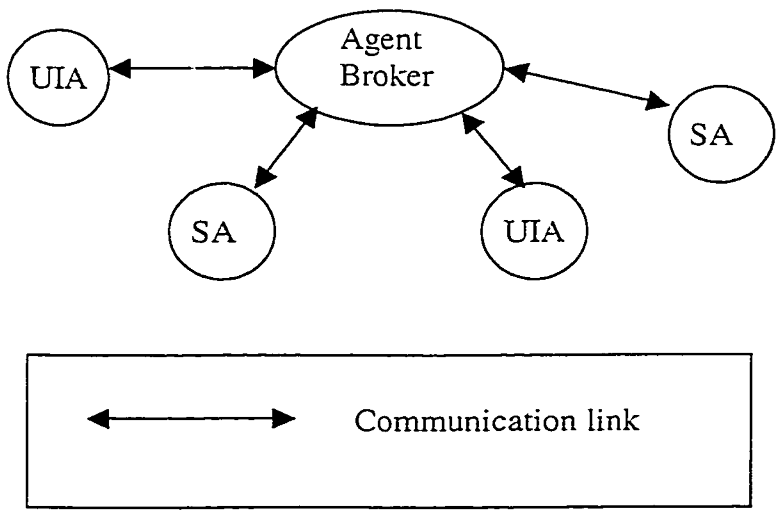

Figure 6.1 Multi-Agent System architecture

As one can see from the figure, our architecture makes use of a special entity that we call the "Agent Broker". The Agent Broker's main function is to register agents in the system and to relay incoming messages to the appropriate agent recipient. Think of the Broker as a general purpose post office that manages addresses and delivers messages. While dynamic address assignment is managed at the Broker level, the problem of naming, for the purpose of identification, is handled at the virtual mall level. More specifically, each SA is identified by the unique name of the vendor it represents (e.g. Sears), and each consumer is identified by the unique *userid* provided upon registration in the virtual mall. The Broker itself has a fixed address, which is known a-priori to all agents. Another feature of the Broker is that it can hold messages intended for agents that are currently offline or temporarily unavailable. Overall, the architecture is that of a message passing system.

6.2.2 Agent Communication Language

A multi-agent system implies agent communication and thus an agent communication language (ACL). Although KQML is perhaps the most prevalent ACL standard used today, we chose for the sake of simplicity not to use KQML. The reason is that our agents are locally built and thus can be made to communicate via our own pre-defined set of performatives.[2] Still, nothing in our system prevents the use of KQML as the agent language.

The language we developed for agent communication is somewhat inspired from KQML. It is based on the exchange of messages, wherein messages are composed of a header and a set of parameters which form the body. The header contains a performative that describes both the format of the message and what to do with the message; the body contains the raw information that is being communicated. To suit our needs, we have defined different performatives to:

1- handle the delivery of messages;
2- handle registration;
3- handle negotiation and ordering.

Table 6.1 presents the various performatives that each type of agent can interpret.

---

[2] Speech acts theory [Searle 69]

Performatives Interpreted by the Broker	
Send :to <agentname> :from <agentid> :content <Message>	Indicates that the sender agent in :from wants the Broker to send the message in the :content parameter to the agent in the :to parameter.
Broadcast: :from < AgentIdentity > :content <Message>	Indicates that the sender agent in :from wants the Broker to send the message in the :content parameter to all registered agents.
BroadcastUIA: :from < AgentIdentity > :content <Message>	Indicates that the sender agent in :from wants the Broker to send the message in the :content parameter to all registered UIAs.
BroadcastSA: :from < AgentIdentity > :content <Message>	Indicates that the sender agent in :from wants the Broker to send the message in the :content parameter to all registered SAs.
Register: :from < AgentIdentity >	Indicates that the sender agent in :from wants the Broker to register him
Unregister: :from < AgentIdentity >	Indicates that the sender agent in :from wants the Broker to unregister him.
Performatives Interpreted by the User Interface Agent	
Confirmation :from < AgentIdentity > :content <OfferReply>	Indicates that the sender sales agent in :from has accepted the user's offer and that the confirmation details can be found in the :content parameters.
Refusal :from < AgentIdentity > :content <OfferReply>	Indicates that the sender sales agent in :from has refused the user's offer and the refusal details can be found in the :content parameters.
Counteroffer :from < AgentIdentity > :content <OfferReply>	Indicates that the sender sales agent in :from has refused the user's offer, but that a counteroffer can be found in the :content parameters.
Performatives Interpreted by the Sales Agent	
Order :from <AgentIdentity> :content <Offer>	Indicates that the sender user interface agent in :from is making a request to order products and that the details of the offer can be found in the :content parameters.

Table 6.1 Agent Performatives

6.3 Implementation

6.3.1 Implementation Environment

The whole system was implemented using the Java programming language. As the overall application is intended for the web, the use of Java was a natural choice. In addition, Java is a language that is highly portable, a desirable feature for a distributed system. Communication links were implemented using Java sockets and TCP/IP. Furthermore, the software Sales Agent uses an expert system shell called Jess as the decision engine to process the consumers' offers (Jess is roughly the Java version of Clips). The Java Expert System Shell (Jess 50a5) is not part of the standard Java Development Kit (JDK 1.1.6) and must be installed separately [Jess].

6.3.2 Software Architecture

The software architecture of our system consists of several classes, among which the two principal ones are the Agent class and the Message class since most of the classes inherit from either one of these two base classes. Figure 6.2 depicts the relationship among the main classes.

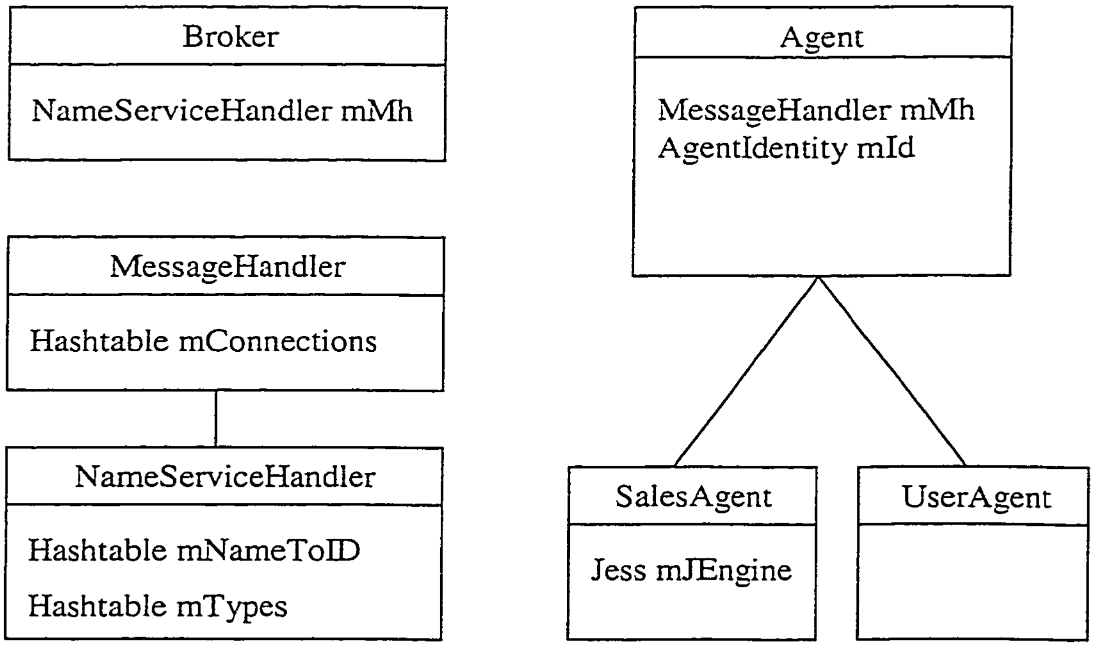

Figure 6.2 Class Relationship

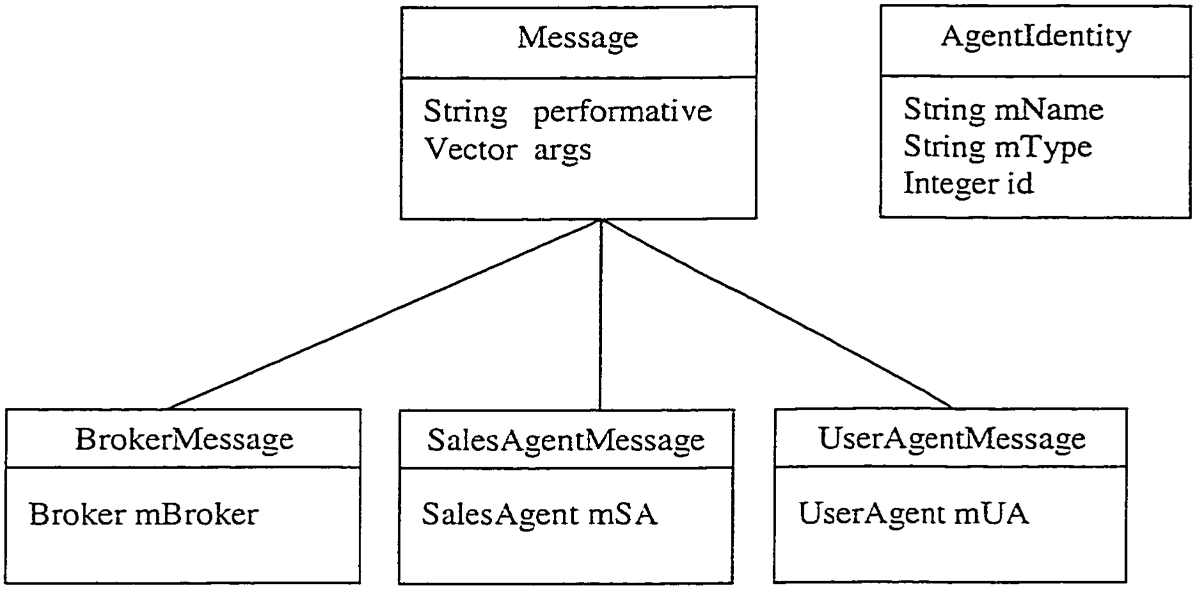

Figure 6.2 Class Relationship (continued)

The Agent class provides basic facilities to connect and communicate with the Broker. More specifically, all connections and communications are handled by the MessageHandler member object. This object has a table of active point-to-point socket connections and provides methods to add and remove connections. Additionally, the MessageHandler object is responsible to send and receive Messages through these connections. To do so, it runs a separate thread for each connection, which makes the process of sending and receiving messages an asynchronous process. In other words, an agent doesn't have to stop what it is doing just to listen for messages. Moreover, execution does not depend on the reception and waiting of incoming messages. The Agent class also offers methods to create Messages based on performatives that the Broker understands. Finally, the AgentIdentity member object holds basic information about the agent, such as its name and type (SA or UA).

The Message class is small, but very useful. It holds a String for the performative and a Vector of Object for the message's argument. It also provides functions to physically read and write arguments to the socket connection. Moreover, the arguments can be of any type, string or Object, as long as they implement the Serializable interface. An important function of this class is the Do() method, which is called by the MessageHandler upon reception of a

Message. The Do() method is actually a "virtual" method in the Message class, i.e. that it is rather implemented by the BrokerMessage, UserAgentMessage and SalesAgentMessage subclasses. This method is used to extract the performative, get the corresponding arguments and call the agent method for that performative. Note that for the mechanism to work, each of these subclasses holds a reference to the agent that created them.

Like an Agent, the Broker also has a MessageHandler member object, except it is of type NameServiceHandler. In addition to the MessageHandler functionality, this class separates connections based on the type of the agent connected and provides a name resolution service to retrieve the ID of the agent when given its name. This allows the Broker to send a message by name or by ID, and to broadcast messages to agent of a certain type. Furthermore, the Broker has methods associated to each performative it understands, such as methods to handle registration, the delivery of messages etc. Similarly, the UserAgent and Sales Agent have methods associated with the performatives they understand. The overall functionality provided by these methods will be discussed in the following sub-section.

6.3.3 Basic Process Description

6.3.3.1 Registration

When an instance of an agent is created, it tries to get a socket connection to the broker. If it gets it, it waits to receive an AgentIdentity response. Upon getting this new connection, the broker sends the new agent an AgentIdentity object containing a unique id number that it will be identified by. When it receives this object, the agent registers with the broker by sending a Register message, which contains an AgentIdentity object containing its name, type and newly received id. This information will be used by the Broker's naming service to send messages to agent named "x" or to broadcast a message to agents of type "y". When a connection goes down, the party connected at the other end unregisters the connection and kills the thread in which it was running. This avoids having unused threads running for nothing.

6.3.3.2 Message Delivery

As mentioned previously, the Broker handles the delivery of messages. To do so, the Broker breaks up the Message and determines the destination, origin and data of the message. With that information, it builds a new Message and delivers it to the destination. If there is no connection active for the intended recipient, the Broker keeps the pending message and will send it to the appropriate agent the next time that agent registers.

6.3.3.3 Negotiation

In our prototype, negotiation is a communication process that involves the user, the User Agent (UA) and the software Sales Agent, whereas the User Agent has been integrated as a component of the User Interface Agent (UIA) [Lu 99]. The User Agent's role consists of presenting information to the user, and relaying the user's offers (or acceptance of offers) to the Sales Agent. But as our implementation of the User Agent relates more to the design of the user interface than to the implementation of the negotiation process, it will be presented in the User Interface sub-section (6.3.4).

For its part, the Sales Agent's role is more complex, as the agent has to handle both the negotiation protocol and a decision module to process the users' offers. The rules that make up our negotiation protocol are shown as pseudo code in Figure 6.3.

```
FOR ALL products IN useroffer
     IF product id NOT IN database THEN
          REPLYWITH "We do not sell this product"
     ELSE
          GET product info FROM database
          SET totalrp TO totalrp + product retail price
          SET totalcp TO totalcp + product cost price
ENDFOR
```

Figure 6.3 Negotiation Protocol

```
IF useroffer's price >= totalrp THEN
      REPLYWITH "Transaction confirmed"
      RECORD transaction
ELSE
      IF EXISTS counteroffer FOR user AND
         NOTEXPIRED counteroffer FOR user THEN
            IF useroffer's price >= counteroffer's price THEN
                  REPLYWITH "Counter-offer acceptance confirmed."
            ELSE
                  REPLYWITH "Counter-offer is at" +
                  counteroffer's price
      ELSE
            IF EXISTS previous_useroffer FOR user AND
               NOTEXPIRED previous_useroffer FOR user THEN
                  REPLYWITH "Your previous offer is still valid"
            ELSE
                  MAKE decision USING Jess
```

Figure 6.3 Negotiation Protocol (Continued)

In addition to the protocol rules, the figure shows that the Jess engine is called if a decision needs to be made about the user's offer. As mentioned before, Jess is an expert system shell written entirely in Java. Furthermore, it uses the CLIPS syntax to define the declarative rules that make up the knowledge base. Rules in expert systems are somewhat similar to IF...THEN statement of procedural languages, the main difference being that the rules are tested over and over as part of a loop. The idea is to react to events that lead to changes in the beliefs. Overall, the whole process is data driven and relies on the inferencing of new data.

One of the advantages of Jess being written in Java. is that it can be embedded and called directly from our Java program. Furthermore, CLIPS rules can be stored separately from the Java code (in a .clp file). This allows for two things: 1) the rules can be easily ported to a CLIPS engine, 2) the rules can be changed without any need to recompile or even stop the Java program.

Figure 6.4 shows sample CLIPS rules derived from our methodology and the example data presented in Table 5.1.

```
(defrule calculate-customer-worth "Calculate the worth for the customer factor"
 (declare (salience 100))
 (purchasehistory ?x)
 (totalworth ?w)
=>
(assert (totalworth (+ ?w  (* (S-function ?x 1000 10000) (/ (+ 0.15 0.25) 2))))))

(defrule calculate-totalbill-worth "Calculate the worth for the total bill "
 (declare (salience 100))
 (totalbill ?x)
 (totalworth ?w)
=>
(assert (totalworth (+ ?w  (* (Linear-function ?x 200 1000) (/ (+ 0.5 0.15) 2))))))

(defrule calculate-inventory-worth-1234
 "Calculate the worth for the inventory of product 1234 factor"
 (declare (salience 100))
 (products $?p&:(member$ 1234 $?p))
 (inventory 1234 ?x)
 (totalworth ?w)
=>
(assert (totalworth (+ ?w  (* (Quadratic-function ?x 500 1500) (/ (+ 0.1 0.35) 2))))))
```

Figure 6.4 Decision Rules

```
(defrule calculate-pdr "Calculates the PDR from the totalworth"
 (totalworth ?w)
 (totalcostprice ?tcp)
 (totalrfixedprice ?tfp)
=>
 (assert (pdr (max ?tcp (- ?tfp (* ?tfp ?w)) (- ?tfp (* ?tfp 0.6))))))

(defrule reject-offer "We reject the offer if below the cost price"
 (pdr ?pdr)
 (totalcostprice ?tcp)
 (consumeroffer ?co&:(<= ?co ?tcp))
=>
(assert (answer no))
(assert (reason "Your offer was refused")))

(defrule counter-offer
"We counter-offer if offer above cost price but below PDR"
 (pdr ?pdr)
 (totalcostprice ?tcp)
 (consumeroffer ?co&:(> ?co ?tcp))
 (consumeroffer ?co&:(< ?co ?pdr))
=>
(assert (answer counteroffer))
(assert (counteroffer ?pdr))
(assert (reason "Your offer was too low.")))

(defrule accept-offer "We accept the offer if above the PDR"
 (pdr ?pdr)
 (consumeroffer ?co&:(>= ?co ?pdr))
=>
(assert (answer yes))
(assert (reason "Your offer was accepted")))
```

Figure 6.4 Decision Rules (Continued)

6.3.4 User Interface Description

Since our interactions with the user are limited, our prototype doesn't require much in terms of a Graphical User Interface (GUI). In fact, most of the functionality the user needs from the overall shopping system is implemented in the User Interface Agent (UIA) [Lu 99]. The user interfaces with our system either through the shopping cart panel of the UIA, or through the simple notification dialog box of the User Agent (UA) component we've incorporated into the UIA. Figures 6.5 shows the shopping cart panel from the UIA.

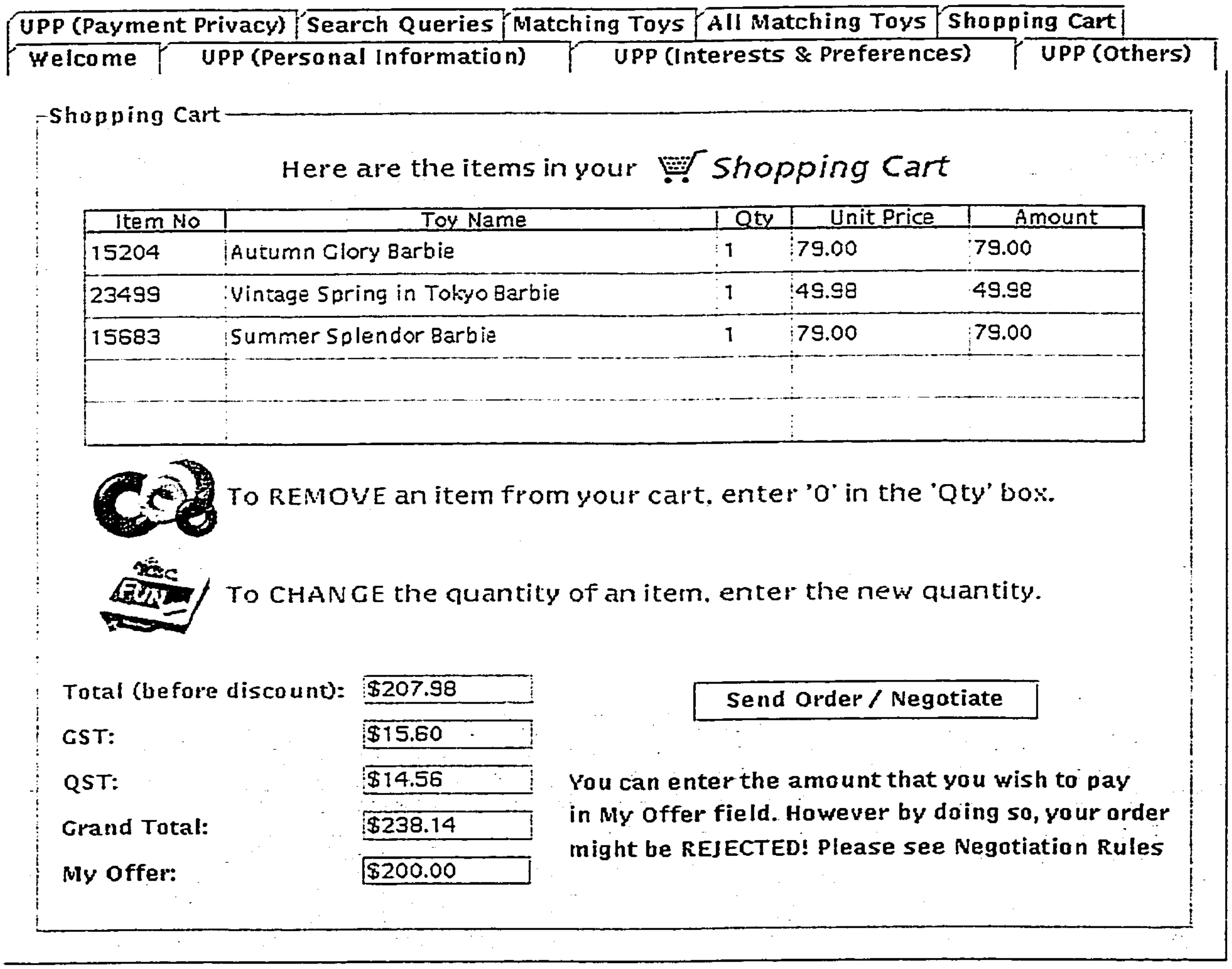

Figure 6.5 Shopping Cart Window [Lu 99]

Figure 6.6, 6.7 and 6.8 show examples of notification message boxes from the UA.

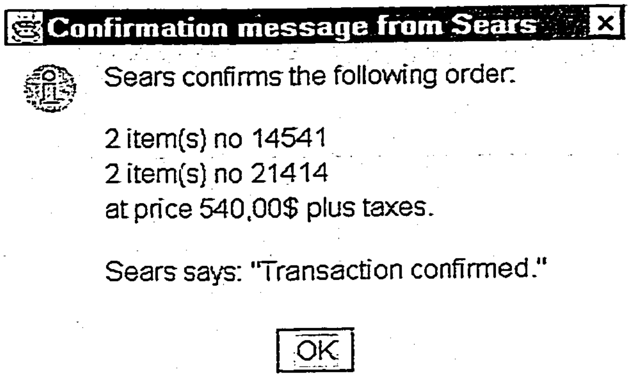

Figure 6.6 Confirmation Message Box

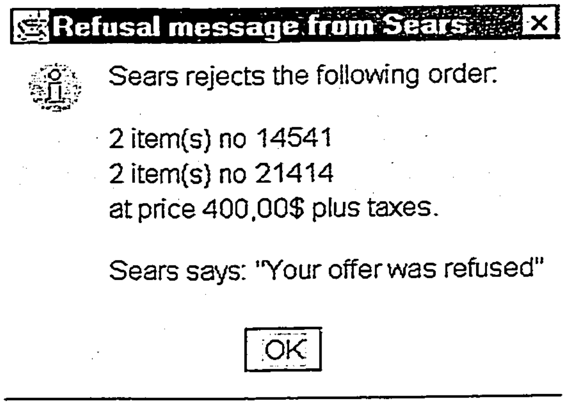

Figure 6.7 Refusal Message Box

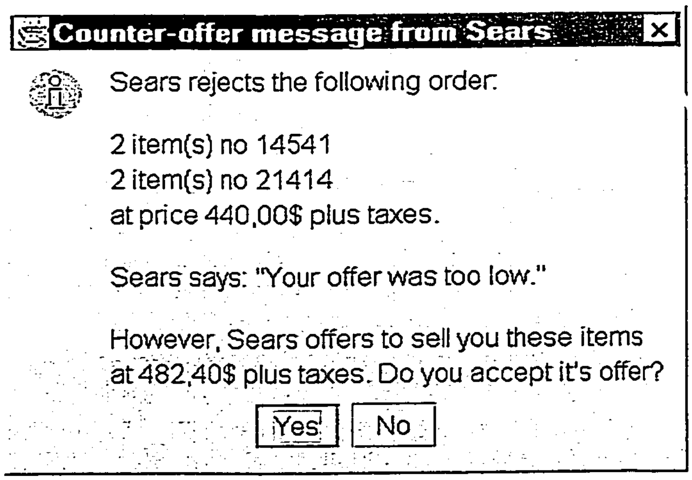

Figure 6.8 Counter-offer Message Box

Chapter 7
Conclusion

Electronic commerce will undoubtedly change the way business is done. Already, we see that the processes that lead to the selling and buying of goods are taking new forms and new directions. Although new business models are emerging, online retail stores still lack an important aspect of today's businesses: negotiation. In order to support conventional business practices as well as new ones on the Internet, the electronic commerce systems need the ability to negotiate. With the help of intelligent software agents, we believe that retail vendors can provide a negotiation service that would give them, at a low cost, a desired flexibility in pricing. We argue that such flexibility will likely lead to increased customer purchases and satisfaction.

7.1 Summary

In this thesis, we have studied the use of software agents in providing an individual one-to-one price negotiation solution to retail markets. Under the scope of the Consumer Buying Behavior model (CBB), we have underlined the fact that current agent technology is still at a research level with regards to the negotiation stage. Similarly as per the business model framework presented in Chapter 2, we have identified a further lack of research in cooperative one-to-one negotiations. Analysis of the different market-driven business models in this thesis has resulted in the conclusion that the cooperative one-to-one approach to negotiation is the most suited approach for retailing.

This thesis has also discussed the non-trivial difficulties involved in automating negotiation, revealing the complexity of the task at hand. In our search for an automated one-to-one negotiated pricing solution, we have shown that the market driven automated solutions to negotiation are no good when applied in a one-to-one setting. Further, we have provided the requirements and specifications for a negotiation protocol, and proposed an "information driven" methodology for the calculation of a "just-in-time personalized price". As a proof of concept, we have also presented a prototype for a software Sale Agent in a Multi-Agent System.

7.2 Results and Contributions

Overall, we have developed and proposed a one-to-one solution to automated negotiation for e-commerce retailing. To the best of our knowledge, the automated system we've outlined in this thesis is the first practicable solution to automated one-to-one negotiation that has been proposed for retailing in e-commerce so far. More specifically, the results and contributions of this thesis are as follows:

- A negotiation protocol that meets some basic desiderata for e-commerce has been developed. The protocol is intuitive, allows for both the consumer and the retailer to make offers, and addresses both the problems of termination and exploitation of strategy. It does not add delays to the buying process, and does not require the use of a third party.

- A systematic information driven personalized pricing methodology has been proposed. The methodology addresses the problem of formulating the goals and strategies by combining the notion of point of diminished return, with valuation associated to information factors provided by the retailer. We have proposed a measure for the retailer valuation in terms of fuzzy logic membership functions. As a strategy, the point of diminished return has been suggested as the retailer's reservation price. Overall the methodology is flexible enough to handle negotiations with different consumers and over any number and type of products.

- A software Sales Agent (SA) operational prototype has been implemented using Java, wherein the decision component required to deal with consumer offers was built using Jess, a Java expert system shell. The prototype serves as a proof of concept that automated agents can be used to autonomously negotiate on behalf of a retail vendor in a one-to-one e-commerce environment. Further, it deals with the ontology issue by using the retailer's online catalog as the common semantic representation and specification between the consumer and the SA.

As part of the CITR e-commerce project requirement for intelligent agent support, we have implemented a Multi-Agent System (MAS) to provide a test bed for our Sales Agent prototype. The MAS allows for a completely distributed system running multiple agents, wherein communication is done asynchronously using an Agent Communication Language (ACL) proposed in this thesis. Additionally, the MAS is robust enough to queue messages for latter delivery when the intended recipients are not currently online.

7.3 Future Work

In order to determine its commercial viability for retailing, the computational methodology addressed in this thesis needs to be tested by real merchants. In particular, the knowledge elicitation aspect of the solution needs to be examined. In cooperation with retail merchants, extensive usability testing and measuring (on a pilot commercial prototype of our solution) will determine if the solution is simple and useful enough to be used commercially. In such testing, we propose to use the increase in customer purchases and satisfaction as analytical measurements of the overall usefulness of the solution. Additionally, our proposed methodology for single-issue price negotiation can be extended to a more flexible multi-issue solution. In such negotiations, the concept of merchant valuation could be applied to factors such as warranty, delivery, after sale service etc.

Prototype wise, the logging of the Sales Agent's decisions has not been implemented in the current version of SA. This is something that needs to be done in a commercial application, because, without a log of the underlying facts that lead to a decision, the agent will simply not be able to gain the trust of the retailers. Furthermore, not enough information about the negotiation protocol is provided to the consumer at the moment. This needs to be addressed because, as we found during the design of our prototype, the user interface has practical implications on the overall consumer understanding and behavior in the negotiations. In terms of added functionality, features such as "Buy at list price if offer is refused" could be interesting for the consumer. As for our Multi-Agent System (MAS), work can be done to improve the scalability of the system. Further, one could replace our custom Agent Communication Language (ACL) by a more standard language like KQML [KQML] or ARCOL [Sad 96].

REFERENCES

[AuctionBot] http://auction.eecs.umich.edu

[BargainFinder] http://bf.cstar.ac.com/bf

[Bhargava 91] Bhargava, Hemant, S. O. Kimbrough, and R. Krishnan, "Unique Names Violations, A Problem for Model Integration - or - You Say Tomato, I Say Tomahto", *ORSA Journal on Computing*, vol. 3, no 2, p. 107-120, 1991.

[Bond 88] A.H. Bond and L. Gasser, *Readings in Distributed Artificial Intelligence*, Morgan Kaufmann, San Mateo, CA, 1988.

[Bui 89] T.M.J. Bui and M. Shakun, "Non-Cooperation in Group Decision Support Systems: Many Problems and Some Solutions", *Journal of Management Science and Cybernetics*, vol. 18, nos. 1-2, p. 51-62, 1989.

[Chaudhury 91] A. Chaudhury, H.R. Rao, S. Rathnam, "What Can Computer Programs Do To Facilitate Negotiation Processes", *Conference Proceedings on Organizational Computing Systems*, p. 269 – 284, 1991.

[Chavez 96] A. Chavez and P. Maes. "Kasbah: An Agent Marketplace for Buying and Selling Goods". *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96)*, London, UK, April 1996.

[Clearwater 96] S. Clearwater, *Market Based Control*, World Scientific, 1996.

[CMIT-1016] C. Beam and A. Segev, "Electronic Catalogs and Negotiations", *Working Paper 96-WP-1016*, Center for Management and Information Technology, 1996. Available at http://haas.berkeley.edu/~citm/wp-1016-summary.html

[CMIT-1019] C. Beam, A. Segev, and J.G. Shanthikumar, "Electronic Negotiation Through Internet-based Auctions", *Working Paper 96-WP-1019*, Center for Management and Information Technology, 1996.
Available at http://haas.berkeley.edu/~citm/wp-1019-summary.html

[CMIT-1022] C. Beam and A. Segev, "Automated Negotiations: A Survey of the State of the Art", *Working Paper 97-WP-1022*, Center for Management and Information Technology, 1997. Available at http://haas.berkeley.edu/~citm/wp-1022-summary.html

[CMIT-1032] C. Beam and A. Segev, "Auctions on the Internet: A Field Study", *Working Paper 98-WP-1032*, Center for Management and Information Technology, 1998. Available at http://haas.berkeley.edu/~citm/wp-1032.pdf

[Compare.net] http://www.compare.net

[Conry 88] S.E. Conry, R.A Mayer and V.R. Lesser, *Multistage Negotiation in Distributed Planning", Readings in Distributed Artificial Intelligence*, Morgan Kaufmann, San Mateo, p367-384, 1988.

[Corkill 79] D.D Corkill, "Hierarchical Planning in Distributed Environment", *Proceedings of the 6th Joint Conference on Artificial Intelligence*, p. 168-179, August 1979.

[Etzioni 94] O. Etzioni and D. Weld, "A Softbot-Based Interface to the Internet", *Communication of the ACM*, vol. 37, no. 7, p 72-79, 1994.

[Firefly] http://www.firefly.com
[Fisher 91] R. Fisher, W. Ury and B. Patton, *Getting to Yes: Negotiating Agreement Without Giving In*, 2nd ed., Houghton Mifflin, Boston, 1991.

[Foner 97] L. Foner, "What's an Agent, Anyway? A Sociological Case Study", MIT Media Lab, 1997. Available at http://agents.www.media.mit.edu/groups/agents/papers.html
[Forrester 97] Forrester Research Report, "Affordable Intimacy Strengthens online Stores", September, 1997.

[Frank 96] R.H. Frank, *Microeconomics and Behavior*, 3rd ed., McGraw-Hill, Inc., 1996

[Fraser 81] N.M. Fraser and K.W. Hipel, "Computer Assistance in Labor-Management Negotiations", *Interfaces*, vol. 11, no 2, p. 22-29, April 1981.

[Genesereth 86] M. Genesereth M.L. Ginsberg and J. Rosenschein, "Cooperation without Communication", *Proceedings of the National Conference on Artificial Intelligence*, p 51-57, Philadelphia, Pennsylvania, August 1986.

[Genesereth 95] M. Genesereth, "KIF Interchange Format Specification", Knowledge Systems Laboratories, Stanford University, 1995.
Available at http://logic.stanford.edu/kif/kif.html

[Georgeff 83] M. Georgeff, "Communication and Interaction in Multi-Agent Planning". *Proceedings of the 1983 National Conference on Artificial Intelligence*, p 125-129, August 1983.

[Gruber 93] T.R. Gruber, "A Translation Approach to Portable Ontologies", *Knowledge Acquisition*, vol. 5, no 2, p. 199-220, 1993. Available at
http://ksl-web.stanford.edu/knowledge-sharing/papers/README.html#ontolingua-intro

[Guttman 98a] R. Guttman and P. Maes, "Agent-mediated Integrative Negotiation for Retail Electronic Commerce", *Proceedings of the Workshop on Agent-Mediated Electronic Trading AMET'98*, Minneapolis, May 1998.

[Guttman 98b] R. Guttman, A. Moukas and P. Maes, "Agent-mediated Electronic Commerce: A survey", *Knowledge Engineering Review*, June 1998.

[Guttman 98c] R. Guttman, "Merchant Differentiation through Integrative Negotiation in Agent-mediated Electronic Commerce", *Master's Thesis*, at MIT Media Lab, 1998.
Available at http://amec.media.mit.edu/Internal/tete-a-tete/thesis/index.html

[Hwang 87] Hwang and Lin, *Group Decision Making Under Multiple Criteria*, Berlin: Springer-Verlag, 1987.

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

[Jarke 87] M. Jarke, M.T. Jelassi and M.F. Shakun, "MEDIATOR: Towards a Negotiation Support System", *European Journal of Operation Research*, vol. 31, no. 3, p. 314-334, September 1987.

[Jelassi 89] M.T. Jelassi, A. Foroughi, "Negotiation Support Systems: An Overview of Design Issues and Existing Software", *Decision Support Systems*, vol. 5, no. 2, p. 167-181, 1989.

[Jess] http://herzberg.ca.sandia.gov/jess

[Kasbah] http://kasbah.media.mit.edu

[Keeney 76] R. Keeney and H. Raiffa, *Decision with Multiple Objectives: Preferences and Value Tradeoffs*, John Wiley & Sons, 1976.

[Kephart 98], J.O. Kephart, J.E. Hanson and J. Sairamesh, "Price-War Dynamics in a Free-Market Economy of Software Agents", *Proceedings of Artificial Life VI*, The MIT Press, 1998.

[KQML] http://www.cs.umbc.edu/kqml/

[Lauer 87] T.W. Lauer and M.T. Jelassi, "PREFCALC – A Multi-Criteria Decision Support System: User Tutorial", *Working Paper #W714*, Institute for Research on the Management of Information Systems, Indian University, November 1987.

[Lewicki 83] R. Lewicki, "Lying and Deception: A Behavioral Model", *Negotiation in Organizations*, Sage Publishing, 1983.

[Lewicki 85] R. Lewicki, and J. Litterer, *Negotiation*, Irwin Co, 1985.

[Lewicki 97] R. Lewicki, D.Saunders, and J.Minton, *Essentials of Negotiation*, Irwin, 1997.

[Linhart 92] P.B Linhart, R. Radner and M.A. Satterthwaite, *Bargaining with Incomplete Information*, Academic Press, San Diego, 1992.

[Lu 99] J. Lu, "User Interface Agents in Electronic Commerce Applications", M*aster's Thesis*, Concordia University, 1999

[MacCrimmon 73] K.R. MacCrimmon, "An Overview of Multiple Objective Decision Making", *Multiple Criteria Decision Making*, University of South Carolina Press, Columbia, S.C, 1973.

[Maes 95] P. Maes, "Artificial Life Meets Entertainment: Life-like Autonomous Agents", *Communication of the ACM* , vol. 38, no. 11, page 108, November 1995.

[Matwin 91] S. Matwin, T. Szapiro and K. Haigh, "A Genetic Algorithm Approach to Negotiation Support Systems", *IEEE Transactions on Systems Man and Cybernetics*, vol. 21, no. 1, p. 102-114, 1991.

[McAfee 87] R.P. McAfee and J. McMillan, "Auctions and Bidding", *Journal of Economic Literature*, p. 699-738, June 1987

[McMillan 94] J. McMillan, "Selling spectrum rights", *Journal of Economic Perspectives*, vol. 8, no. 3, p. 145-162, 1994.

[Milgrom 89] P. Milgrom, "Auctions and Bidding: A Primer", *Journal of Economic Perspectives*, p. 3-22, summer 1989.

[Myerson 83] R.B. Myerson and M. Satterthwaite, "Efficient Mechanism for Bilateral Trading", *Journal of Economic Theory*, vol. 29, p. 265-281, 1983.

[Nwana 96] H. Nwana, L. Lee and N. Jennings, "Coordination in Software Agent Systems", *The British Telecom Technical Journal*, vol. 14, no. 4, p. 79-88, 1996.

[Oliver 96] J.R. Oliver, " On Artificial Agents for Negotiation in Electronic Commerce", *Ph.D. dissertation*, Wharton School, 1996. Available at http://opim.wharton.upenn.edu/~oliver27/dissertation/diss.zip

[Persona Logic] http://www.personalogic.com

[Preist 98] C. Preist and M.V. Tol, "Adaptive Agents in a Persistent Shout Double Auction", *Proceedings of the first International Conference on Information and Computation Economics*, Charleston, South Carolina, October, 1998.

[Pruitt 81] D. Pruitt, *Negotiation Behavior*, Academic Press, New York, 1981.

[Raiffa 82] H. Raiffa, *The Art and Science of Negotiation*, Harvard University Press, Cambridge, 1982.

[Rangaswamy 97] A. Rangaswamy and G.R. Shell, "Using Computers to Realize Joint Gains in Negotiations: Toward an Electronic Bargaining Table", *Management Science*, vol. 43, no. 8, August 1997.

[Reimers 96] K. Reimers, "The Non-market Preconditions of Electronic Markets Implications for their Evolution and Applicability", *European Journal of Information Systems*, vol. 5, no. 2, p. 75-83, Summer 1996.

[Robinson 98] W.N. Robinson and V. Volkov, "Supporting the Negotiation Life Cycle", *Communication of the ACM*, vol. 41, no. 5, p95-102, May 1998.

[Rosenschein 94] J. Rosenschein and G. Zlotkin. *Rules of Encounter: Designing Conventions for Automated Negotiations among Computers*, MIT Press, 1994.

[Rust 93] J. Rust and D. Friedman, *The Double Auction Market: Institutions, Theories and Evidence*, Workshop on Double Auction Markets, Addison-Wesley Reading, MA, 1993.

[Sad 96] Sadek, D., Bretier P. and Panaget, F., "France Télécom's ARTIMIS technology, ARCOL agent communication language and MCP, CAP and SAP agent's cooperativeness protocols", December 1996.
Available at http://drogo.cselt.stet.it/fipa/torino/cfp1/propos97_015.htm.

[Sairamesh 98] J. Sairamesh and J.O. Kephart, "Price Dynamic of Vertically Differentiated Information Markets", *Proceedings of the First International Conference on Information and Computation Economy*, p 28-36, 1998.

[Sandholm 93], T. Sandholm, "An implementation of the Contract Net Protocol Based on Marginal Cost Calculations", *Proceeding of the Eleventh National Conference on Artificial Intelligence*, p 256-262, Washington DC, 1993.

[Sandholm 95] T. W. Sandholm and V. R. Lesser, "Coalition formation among bounded rational agents", *Proceedings of Fourteenth International Conference on Artificial Intelligence*, 1995.

[Schwartz 97] E. Schwartz, *Webonomics: Nine Essential Principles for Growing Your Business on the World Wide Web*, Broadway Books, 1997.

[Searle 69] J. Searle, *Speech Acts*, Cambridge University Press, Cambridge, MA, 1969.

[Smith 80] R.G. Smith, "The Contract Net Protocol: High-Level Communication and Control in a Distributed Problem Solver", *IEEE Transaction on Computers*, vol. 29, no. 12, December 1980.

[Sokol 89] P. K. Sokol, *EDI: the Competitive Edge*, McGraw-Hill, New York, 1989.

[Tesauro 98] G.J. Tesauro and J.O. Kephart. "Foresight-based Pricing Algorithms in an Economy of Software Agents", *Proceedings of the First International Conference on Information and Computation Economics*, Charleston, South Carolina, October, 1998. Available at http://www.research.ibm.com/infoecon/paps/html/ice98_fs/fs_public.html.

[Tete-a-Tete] http://ecommerce.media.mit.edu/Tete-a-Tete

[Wiederhold 94] G. Wiederhold, "Interoperation, Mediation, and Ontologies," *Proceedings of the International Symposium on Fifth Generation Computer Systems*, Workshop on Heterogeneous Cooperative Knowledge-Bases, vol. W3, p. 33-48, ICOT, Tokyo, December 1994.

[Wooldridge 95] M. Wooldridge and N.R. Jennings, "Agent Theories, Architectures, and Languages: a Survey", *Intelligent Agents,* Berlin: Springer-Verlag, 1995.

[Wurman 98] P.Wurman, M. Wellman, and W. Walsh, "The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents", *Proceedings of the second International Conference on Autonomous Agents (Agents '98)*, May 1998.

[Zeleny 82] M. Zeleny, *Multiple Criteria Decision Making*, McGraw-Hill, New York 1982.

[Zeng 96] D. Zeng, and K. Sycara, "Bayesian Learning in Negotiation", *Working Notes of the AAAI 1996 Stanford Spring Symposium Series on Adaptation, Co-evolution, and Learning in Multiagent Systems,* also available at http://www.cs.cmu.edu/~zeng

[Zeng 97] D. Zeng, and K. Sycara, "Benefit of Learning in Negotiation", *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, Rhode Island, 1997.